

FACHHOCHSCHULE WEDEL

SEMINARARBEIT

in der Fachrichtung

Wirtschaftsinformatik

Thema:

Conflict-free Real-time AGV Routing

Beschreibung eines Algorithmus mit Zeitfenstern

Eingereicht von: Max Schildt
Schmilinskystraße 24
20099 Hamburg
Tel. (040) 28 00 77 77
Mobil. 0170 58 69 424

Erarbeitet im: 8. Semester

Abgegeben am: 19. Juni 2006

Leitung (FH Wedel): Prof. Dr. Sebastian Iwanowski
Fachhochschule Wedel
Feldstraße 143
22880 Wedel
Tel. (041 03) 80 48-63

Inhaltsverzeichnis

Inhaltsverzeichnis	II
Abbildungsverzeichnis	III
Abkürzungsverzeichnis	III
1. Einführung	1
2. Autonomes Fahren	2
2.1. Einführung	2
2.2. Techniken zur Steuerung automatischer Fahrzeuge	3
3. Der Container Terminal Altenwerder	4
4. Problemstellung	6
4.1. Einführung	6
4.2. Das Modell	7
5. Der Algorithmus	9
5.1. Übersicht	9
5.2. Physikalische Eigenschaften prüfen	9
5.3. Kürzesten Weg mit Zeitfenster bestimmen	11
5.3.1. Initialisierung.....	12
5.3.2. Label expandieren	12
5.3.3. Dominanztest.....	12
5.4. Zeitfenster aktualisieren	13
5.5. Zur Komplexität	13
6. Schlussbemerkung	15
Literaturverzeichnis	16

Abbildungsverzeichnis

Abb. 1: Darstellung eines Live- und Deadlocks	6
Abb. 2: Gerichteter Graph als Modell des AGS	8
Abb. 3: Übersicht über die Schritte des Algorithmus.....	9
Abb. 4: Ausschnitt eines Graphen mit Polygonen und Transpondern	11
Abb. 5: Der Algorithmus in Pseudo-Code	13

Abkürzungsverzeichnis

AGV	Automated Guided Vehicle
FTF	Fahrerloses Transportfahrzeug
AGS	Automated Guided System
CTA	Container Terminal Altenwerder
HHLA	Hamburger Hafen- und Logistik AG
SPP	Shortest Path Problem
SPPTW	Shortest Path Problem with Time-Windows

1. Einführung

In dieser Arbeit wird ein Algorithmus vorgestellt, der für AGVs, so genannte Automated Guided Vehicles (fahrerlose Transportfahrzeuge) in Echtzeit eine konfliktfreie Route berechnet. Der Algorithmus wurde im Rahmen eines Projektes des Bundesministeriums für Bildung und Forschung, der Hamburger Hafen- und Logistik AG (HHLA) und der TU-Berlin speziell für den Container Terminal Altenwerden (CTA) in Hamburg entwickelt. Der Unterschied zu dem bereits eingesetzten Verfahren ist, dass jegliche Kollisionsverhütung zum Zeitpunkt der Routenberechnung erfolgt und dadurch keine späteren Maßnahmen in dieser Hinsicht nötig sind. Bis jetzt ist es möglich und wahrscheinlich, dass ein AGV auf seiner Route durch das Automated Guided System (AGS) anderen automatischen Fahrzeugen begegnet. Das heißt, obwohl eine Route berechnet und dem AGV bekannt ist, eine Kollisionsverhütung während des Fahrens dauerhaft nötig ist. In Altenwerden wird derzeit dies Problem gelöst, indem die im Asphalt versenkten Transponder, die als nächstes angesteuert werden und an denen sie die AGV sich orientieren, für alle anderen AGVs gesperrt werden. Dabei kann es aber zu so genannten Deadlocks und Livelocks kommen, denen wiederum Lösungsstrategien entgegengestellt werden müssen. Es soll gezeigt werden, dass der vorgestellte Algorithmus eine konfliktfreie Route in angemessener Zeit berechnen kann und, dass sich dadurch die durchschnittliche Fahrtzeit der AGVs reduziert.

2. Autonomes Fahren

2.1. Einführung

Allgemein wird unter „Autonomen Fahren“ das automatische, führerlose Fahren von Fahrzeugen verstanden. Autonom Fahren heißt, dass sich das Fahrzeug ohne einen menschlichen Fahrer oder Führer zielgerichtet bewegt. Personen sind in keiner Weise an der Bedienung des Fahrzeugs beteiligt. Das Schalten, Kuppeln, Bremsen, Lenken, die Routenwahl und die Hinderniserkennung werden selbstständig von einer Fahrzeugsteuerung erledigt.

Sinnvollerweise wird zwischen FTF (fahrerlosen Transportfahrzeugen), die auf einem vermessenen und bekannten Gelände (Indoorbereich) unterwegs sind und normalen Autos, die Personen befördern und sich im öffentlichen Straßenverkehr mit teilweise sehr unterschiedlichen Verkehrsteilnehmern und Bedingungen bewegen, unterschieden. FTF im Indoorbereich gibt es nämlich schon seit den 50er Jahren, während Autonomes Fahren im öffentlichen Straßenverkehr bislang nicht mehr ist als eine Zukunftsvision, von der die Realität noch Jahrzehnte entfernt ist. Nicht von ungefähr ist Autonomes Fahren auf öffentlichen Straßen gesetzlich verboten. Vor allem die Hinderniserkennung muss deutlich bessere Resultate liefern, als dies durch den Menschen möglich ist. Gerade hier liegt in der Flexibilität der menschlichen Urteilskraft ein großer Vorteil. Es gibt lediglich verschiedene Mischformen, wie die Busbahn oder die elektronische Deichsel, die die Hinderniserkennung nach wie vor einem menschlichen Fahrer bzw. die Spurführung einer automatischen Steuerung überlässt.

Das Ziel für den Einsatz von FTFs im gewerblichen Betrieb ist die Erhöhung der Produktivität und der damit verbundenen Kostenreduzierung. Unter Umständen ist ein schnellerer, sicherer und zuverlässigerer Betrieb ohne Einschränkung hinsichtlich Ermüdung und Ruhepausen möglich. Für den privaten, öffentlichen Personenbetrieb steht vor allem die steigende Sicherheit (z. B. automatische Geschwindigkeitsbegrenzung) und die Steuerung und Lenkung des Verkehrsaufkommens im Vordergrund.

2.2. Techniken zur Steuerung automatischer Fahrzeuge

Damit sich ein AGV überhaupt in einem Verkehrssystem kontrolliert fortbewegen kann, muss es Informationen darüber erhalten, wo die Fahrbahnen entlangführen, um entsprechende Kurskorrekturen an die Lenkung zu übermitteln. Zudem müssen Vorkehrungen getroffen werden um Kollisionen mit anderen Verkehrsteilnehmern oder unerwarteten Hindernissen zu verhindern.

Bekannte Techniken für die Spurführung sind Leitdraht, Transponder, Magnet, Scanner (Laser), Bildverarbeitung und DGPS. Der Leitdraht ist ein im Asphalt verlegtes Kabel, dessen Magnetfeld zur Orientierung benutzt wird. Diese Technik hat sich bewährt, ist aber relativ unflexibel und simpel. Sind Transponder im Einsatz, übertragen diese bei Annäherung und Aktivierung durch ein Energiefeld ihre Position, die dem AGV eine absolute Orientierung in einem bekannten Gelände ermöglicht. Transponder sind relativ teuer; wobei Magnete eine billigere Alternative darstellen. Bei der Steuerung durch Bildverarbeitung werden z. B. Markierungsstreifen auf dem Asphalt ausgelesen und ausgewertet. Dies ist eine sehr flexible und billige Lösung, da die Streifen einfach übermalt werden können. Für ein zentimetergenaues Fahren liefert das GPS zu ungenaue Positionsdaten. Daher werden diese Daten meist über eine Referenzstation in unmittelbarer Umgebung, deren Position exakt bekannt ist, korrigiert, so dass mit einer Genauigkeit +/- ein Zentimeter gerechnet werden kann. Ein Scanner orientiert sich an künstlich aufgestellten Fahrbahnmarkierungen. Dabei misst ein Laser den Winkel und den Abstand zu diesen Marken. Diese Technik ist zwar flexibel und genau, aber sehr anfällig gegen Verschmutzungen und daher nur eingeschränkt für den Outdoorbereich geeignet.

Um Kollisionen zu verhindern ist es sinnvoll, Kommunikation zwischen Verkehrsteilnehmern möglich zu machen oder Abschnitte für andere zu sperren. Kommuniziert werden kann über induktive Datenübertragung, Infrarot oder Datenfunk, wobei sich letzterer immer mehr durchsetzt. Gründe hierfür sind die Übertragungsgeschwindigkeit und die Reichweite.

Das Hauptproblem sind sicherlich unerwartet auftretende Hindernisse, wie sie auf öffentlichen Straßen häufig auftreten können. Techniken zur Hinderniserkennung sind Radar, Laser und Bildverarbeitung.

3. Der Container Terminal Altenwerder

Der Container Terminal Altenwerder (CTA) ist einer der größten und modernsten Containerhäfen weltweit. Er wird von der CTA GmbH betrieben, die zu 74,9 % der Hamburger Hafen- und Logistik AG (HHLA) und zu 25,1 % der Hapag-Lloyd AG gehört.

Der Containerhafen ist in verschiedenen Bauabschnitten von 1997 bis 2005 am Westufer des Köhlbrands, südlich der Köhlbrandbrücke, errichtet worden. Für den Bau musste das Fischerdorf Altenwerder aufgekauft und abgerissen werden. Mit der Planung des Hafens wurde bereits in den 50er Jahren begonnen und ab 1960 wurden die ersten Grundstücke aufgekauft und die darauf befindlichen Häuser abgerissen. Allerdings weigerten sich einige Eigentümer bis 1998 ihre Grundstücke zu verkaufen und wurden schließlich durch die Androhung von Enteignung zur Aufgabe gezwungen.

Seit der Fertigstellung des letzten Bauabschnitts (März 2005) sind vier Liegeplätze mit insgesamt 14 großen Containerbrücken (Super-Post-Panamax-Brücken) vorhanden. Zusätzlich gibt es einen weiteren Liegeplatz mit einer kleinen Containerbrücke für so genannte Feeder, die Container in Häfen verschiffen, die zu klein für große Containerschiffe sind oder deren Anlaufen nicht lohnend ist. Im CTA sollen nach Beendigung des letzten Bauabschnittes bis zu 2,4 Mio. Container im Jahr, das sind 25 % der gesamten Kapazität des Hamburger Hafens, umgeschlagen werden. Ungefähr 50% der Container werden auf der Straße weiter transportiert, 25% per Feeder und 25% per Bahn.

Die Geschwindigkeit mit der ein Schiff entladen wird, ist entscheidend für dessen wirtschaftlichen Betrieb. Im Hafen entstehen den Betreibern hohe Liegeplatzkosten, weshalb jede Reederei versucht, diese Hafenzeiten so kurz und so günstig wie möglich zu halten. Daher bedeutet eine höhere Geschwindigkeit beim Be- und Entladen und geringere Liegeplatzkosten pro Stunde einen Standortvorteil für Containerhäfen.

Das Besondere an Altenwerder ist – und das macht ihn zu einem der modernsten Häfen der Welt – ist der hohe Automatisierungsgrad beim Abfertigen eines Schiffes. Die Fahrzeuge, die die Container zwischen dem Schiff und den Lagerplätzen hin und her transportieren, fahren führerlos bzw. automatisch. In Altenwerden sind bis

zu 73 dieser AGVs im Einsatz, die beim Entladen eines Schiffes die Container von den Containerbrücken (diese werden teilautomatisch betrieben) abholen und sie zu einem vollautomatischen Portalkranpaar (Double Rail Mounted Gantry – DRMG) transportieren, die den Container abladen und in einem Zwischenlager lagern. Beim Beladen eines Schiffes bringen die AGVs die Container von den Lagerplätzen zu den Containerbrücken.

Das Gelände des Container Terminals Altenwerden auf dem die AGVs fahren, ist 1400 Meter lang und ca. 100 Meter breit. Es ist millimetergenau vermessen. Im Asphalt sind 13.000 passive Transponder installiert, die von einem AGV bei Annäherung durch ein Energiefeld aktiviert werden und ihre exakte Position an das Fahrzeug übermitteln. Die AGVs kommunizieren mit einem Tower über Funk und werden zusätzlich mit Hilfe von DGPS überwacht. Da die Geschwindigkeit der AGVs 25 km/h beträgt, ist der Fahrbereich der AGVs für Personen gesperrt (gesetzliche Vorgabe bei Anwesenheit von Menschen: höchstens 6 km/h). Auf eine Hinderniserkennung wurde daher vollständig verzichtet. Ein AGV kann mit einem vollen Tank 48 Stunden fahren. Koordiniert werden die Aktionen durch eine Terminallogistik- und Steuerung (TLS).

4. Problemstellung

4.1. Einführung

Die Produktivität der AGVs ist maßgeblich von der Steuerung durch die Verkehrsleitung abhängig. Das allgemeine Problem ist als Kürzeste-Wege-Problem (Shortest Path Problem - SPP) bekannt. Für dieses Problem gibt es den bekannten und beliebten Algorithmus von Dijkstra, der den kürzesten Weg auch in großen Graphen schnell findet (siehe Paper). Dieses Konzept wird zurzeit im Container Terminal Altenwerden umgesetzt und wird als „static approach“ bezeichnet. Der Nachteil ist, dass die Routen nicht konfliktfrei sind, das heißt, es muss eine zusätzliche Kollisionsverhütung zur Echtzeit existieren. Darüber hinaus müssen Strategien implementiert werden, die Deadlocks und Livelocks verhindern. Ein Deadlock entsteht, wenn zwei AGVs dieselben Transponder gleichzeitig für sich beanspruchen, so dass beide ihre Fahrt nicht fortsetzen können. Livelocks entstehen, wenn ein AGV über einen längeren Zeitraum seine Fahrt nicht fortsetzen kann, weil seine Route durch einen anhaltenden Fluss anderer AGVs gesperrt wird.

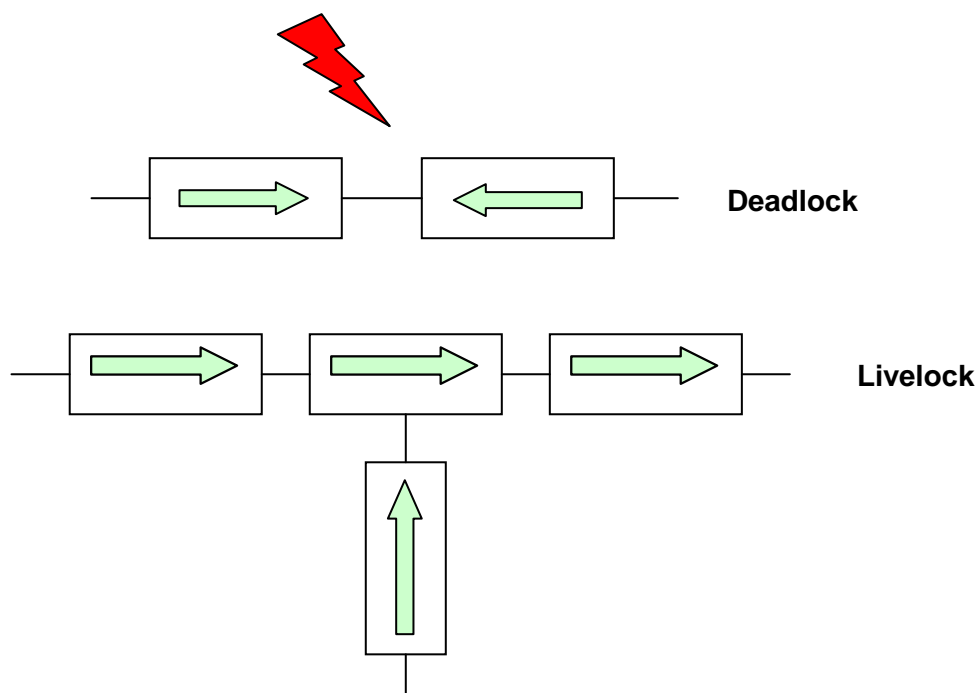


Abb. 1: Darstellung eines Live- und Deadlocks

Dead- und Livelocks können sich unter Umständen äußerst negativ auf die Performanz des Systems auswirken. Die Aufgabe des neuen Algorithmus ist daher, eine konfliktfreie Route (keine Kollisionen, Deadlocks oder Livelocks) zum Zeitpunkt der Routenberechnung zu finden. Abschließend muss gezeigt werden, dass dadurch bessere Resultate erzielt werden, als bisher durch den static approach. Die Komplexität der Routen-Berechnung steigt in jedem Fall an. Es ist daher allgemein fraglich, ob eine Berechnung in angemessener Zeit (Echtzeit) überhaupt möglich ist. Um eine konfliktfreie Route berechnen zu können, sind insbesondere die physikalischen Eigenschaften der AGVs und die Zeitpunktbezogenheit der Route zu berücksichtigen. Das Ziel ist, einen kürzesten Weg zu finden, der die Zeitfenster berücksichtigt, in denen eine Route befahren werden kann, ohne dass es zu einer Kollision kommen kann.

4.2. Das Modell

Die Wege des Automated Guided Systems (AGS) werden als gerichteter Graph $G = (V, A)$ modelliert. Die Kanten sind die Wege, denen jeweils eine Fahrzeit $\tau(\alpha)$ zugeordnet wird. Diese Zeit gibt an, wie lange ein AGV benötigt, um die Strecke zu bewältigen. Zudem gibt es für jede Kante eine Menge Zeitfenster F , in der die Zeiten in Form von Zeitintervallen $[t_1, t_2]$ gespeichert werden, zu denen die Kante frei ist – so genannte Zeitfenster. Zu den übrigen Zeiten – den so genannten Blockings – ist die Kante gesperrt, sei es, weil diese Kante befahren wird oder weil sie von einer befahrenen Kante geographisch abhängig ist und deshalb gesperrt wird.

Ein Weg (oder eine Route) ist eine Folge von Kanten, die von einem Startknoten zu einem Zielknoten führt. Natürlich soll dieser Weg möglichst kurz sein, um den Durchsatz der AGVs zu erhöhen, wobei die reine Fahrzeit eine, aber nicht die einzige Rolle spielt. Zu den reinen Fahrzeiten müssen die Wartezeiten hinzuaddiert werden, da die Route zeitpunktbezogen ist. Zeitpunktbezogen meint, dass für die Beschreibung einer Route nicht nur die Reihenfolge der Kanten angegeben werden müssen, sondern auch Zeitintervalle, in denen die Kanten befahren werden dürfen. Damit existiert zum Zeitpunkt der Routenberechnung eine Route mit einer garantierten Reisezeit, Wartezeit und Ankunftszeiten.

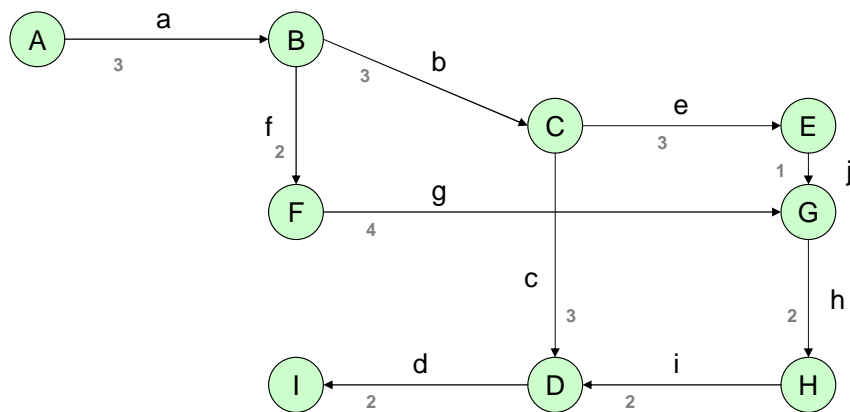


Abb. 2: Gerichteter Graph als Modell des AGS

Das Problem das sich damit stellt ist bekannt als Kürzeste-Wege-Problem mit Zeitfenstern (Shortest Path Problem with Time-Windows - SPPTW). Da dies ein NP-schweres Problem ist, bedarf es zusätzlicher Anstrengungen, die Berechnung der kürzesten Wege und der Zeitfenster in polynomialer Zeit durchzuführen. Warum dies trotzdem möglich ist, wird weiter unten besprochen.

5. Der Algorithmus

5.1. Übersicht

Der entwickelte und hier beschriebene Algorithmus ist ein generalisierter kanten-basierter Dijkstra-Algorithmus, der mit Labeln arbeitet. Der Algorithmus ist in folgende Abschnitte gegliedert:

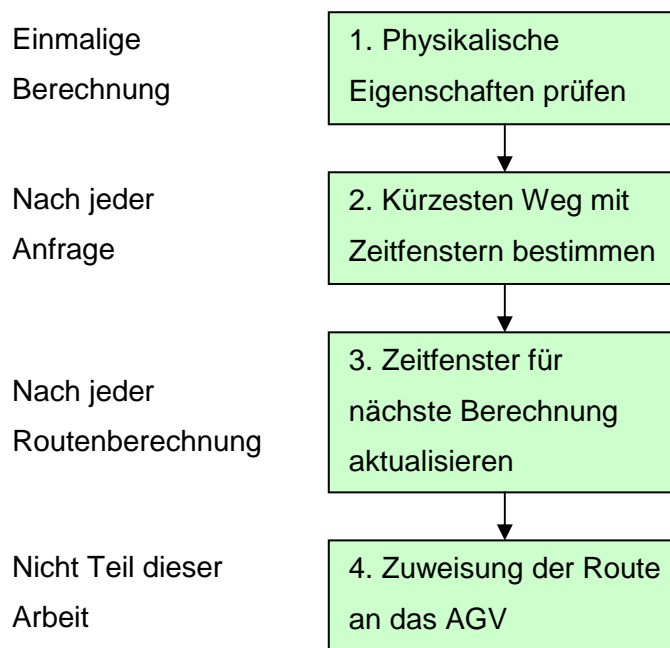


Abb. 3: Übersicht über die Schritte des Algorithmus

Der Algorithmus nimmt eine Sequenz $\sigma = r_1, \dots, r_n$ von Anfragen der Gestalt $r_j = (s_j, t_j, \theta_j)$ entgegen und beantwortet die Anfragen ohne Informationen über einen weiteren Verlauf des Verkehrsaufkommens. Wobei s der Startknoten, t der Zielknoten und θ die gewünschte Startzeit ist.

5.2. Physikalische Eigenschaften prüfen

Um die physikalischen Eigenschaften der AGVs zu berücksichtigen (und damit wirklich eine konfliktfreie Route berechnen zu können), wird für jede Kante ein

Polygon (Vieleck) erstellt. Diese Polygone sind natürlich abhängig vom realen System, das durch den Graphen modelliert wird und von den Dimensionen der eingesetzten AGVs. (Es ist davon auszugehen, dass die Polygone auch die Knoten „überdecken“, also, dass die Knoten infinitesimal klein sind.) Die Polygone werden so gewählt, dass Breite und Länge der AGVs sowie ein zusätzlicher Sicherheitsabstand zur Seite in Geradeaus- und in Kurvenfahrt existiert. Sinnvollerweise verlaufen parallele Strecken, die an der Position der Transponder festgemacht sind so, dass sich die Polygone gerade nicht überschneiden. Es ist überdies davon auszugehen, dass die Kanten des Modells real nicht allzu lang ausfallen, also z. B. keinesfalls ganze gerade Steckenteile über mehrere hundert Meter umfassen. Dies würde zu erheblichen Effizienzverlusten führen, da sich zur selben Zeit immer nur ein AGV auf einer Kante befinden kann. Daher wäre es sinnvoll die Kantenlänge folgendermaßen zu wählen: Länge des AGVs plus Sicherheitsabstand nach vorne und hinten.¹

In einem Preprocessing-Schritt werden diese Polygone paarweise auf Überschneidungen überprüft. Sobald die Koordinaten des AGS vermessen und die Routen bekannt sind, erfolgt diese Berechnung einmalig. Das Resultat sind so genannte Konfliktmengen $\text{confl}()$ für jede Kante des Graphen. Überschneiden sich zwei Polygone $P(a)$ und $P(b)$, so werden diejenigen Kanten des Graphen g jeweils der Menge $\text{confl}(a)$ und $\text{confl}(b)$ hinzugefügt. Das heißt, dass es zu keiner Kollision kommen kann, wenn diese Kanten nicht gleichzeitig von zwei AGVs befahren werden. Befindet sich ein AGV auf einer Kante a , werden für diesen Zeitraum alle Kanten aus der Menge $\text{confl}(a)$ für andere AGVs gesperrt. Das bedingt eine weitere Menge „Zeitfenster“ $F(a)$ pro Kante, in der die blockierten und freien Zeiten gespeichert werden. Wird nun ein kürzester Weg gesucht, darf dieser sich nur der freien Zeiten bedienen.

Zusätzlich wird für jede Kante α eine Menge $\text{Out}(\alpha)$ angelegt, in der alle Kanten hinzugefügt werden, die erlaubter Weise gewählt werden können, nach dem Kante α die physikalischen Eigenschaften des AGVs berücksichtigt hat.

¹ bei 25 km/h wären dies 20 Meter für eine Kante, davon 15 Meter für das AGV und jeweils 2,5 Meter für die Sicherheitsabstände. Die Reisezeit beliefe sich dann auf drei Sekunden pro Kante. Sind Kurven oder Beschleunigungen zu berücksichtigen, kann sich die Reisezeit leicht vergrößern. Wie viele Transponder dieser Kante zugrunde liegen, hängt von den technischen Rahmenbedingungen ab. Es muss lediglich garantiert werden, dass ein AGV nicht signifikant vom Fahrkurs abweicht, bis es mit Hilfe des nächsten Transponders den Kurs korrigieren kann.

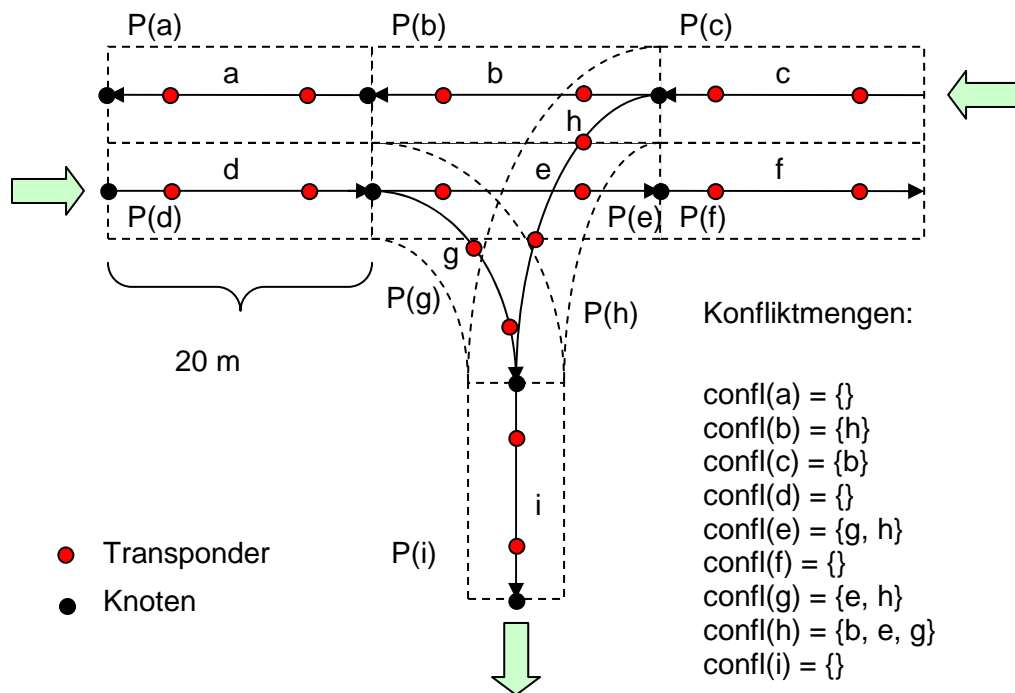


Abb. 4: Ausschnitt eines Graphen mit Polygonen und Transpondern

Es ist notwendig, dass, falls die nachfolgende Kante gesperrt ist, eine Warteposition (mit ausreichend Sicherheitsabstand nach vorne) auf einer Kante eingenommen wird, die das jeweilige Polygon dieser Kante nicht überschreitet. Im oben angeführten Beispiel ist es beispielsweise möglich, dass ein AGV unmittelbar vor dem letzten Transponder der Kante anhält.

5.3. Kürzesten Weg mit Zeitfenster bestimmen

Für den Ablauf des Algorithmus werden Labels definiert: Ein Label $L = (\alpha_L, d_L, I_L, \text{pred}_L)$ der Kante α_L besteht aus einem Distanzwert d_L , dem Zeitintervall I_L , zu dem eine Ankunft beim Ende von α_L möglich ist und einen Vorgänger (Predecessor) pred_L . Jedes Label repräsentiert einen Weg vom Startknoten s zum Ende von α_L . Die Labels werden in einer Vorrangwarteschlange H nach kleinstem d gespeichert.

5.3.1. Initialisierung

Es wird ein Label $(\alpha, 0, (0, \infty), \text{nil})$ erstellt und allen Kanten aus der Menge $\text{Out}(s)$ zugeordnet.

5.3.2. Label expandieren

Geht zum Beispiel zum Zeitpunkt null für Kante a das Zeitfenster f von null bis ∞ und beträgt die Fahrzeit $\tau(a)$ drei Zeiteinheiten, so ist ein Expandieren des Labels grundsätzlich möglich, da das Zeitfenster f vom Zeitpunkt null bis drei frei ist und die drei Zeiteinheiten für die Fahrt belegt werden können. Das Labelintervall wird aber noch weiter expandiert, bis zum Ende des Zeitfensters f , also bis ∞ . Das neue Labelintervall lautet nun $(3, \infty)$. Danach werden die Kosten auf d addiert: in diesem Beispiel sind es drei Zeiteinheiten, da keine Wartezeiten hinzukommen. Grundsätzlich sind beides, Fahrt- und Wartezeit, zu berücksichtigen. Abschließend wird der Vorgänger von a gesetzt.

Das fertig expandierte Label lautet nun: $(a, 3, (3, (3, \infty)), s)$

5.3.3. Dominanztest

Nachdem das Label expandiert wurde, muss geprüft werden, ob es den Kanten α aus $\text{Out}(a)$ hinzugefügt wird oder nicht. Das neue Label wird den Kanten hinzugefügt, wenn es von keinem anderen Label der Kante α dominiert wird. Ein Label L dominiert ein Label L' genau dann wenn, $d_L \leq d_{L'}$ und $I_{L'} \subseteq I_L$. Der Algorithmus läuft nun wie folgt ab:

```

* Initialisierung.
* Schleife
  Beginn
    * Label L mit kleinstem d aus H auswählen.
    * Wenn kein Label in H vorhanden,
      dann kein Weg, Stop.
    * Wenn Ziel erreicht,
      dann Pfad zurückgeben, Stop.
    * Für jedes Zeitfenster f der Kante  $\alpha_L$ 
      * Label expandieren.
      * Dominanztest.

  Ende.
* Zeitfenster aktualisieren.

```

Abb. 5: Der Algorithmus in Pseudo-Code

5.4. Zeitfenster aktualisieren

Nach dem eine Route unter Berücksichtigung der Zeitfenster berechnet wurde, werden die Kanten der Route zu den Reisezeiten gesperrt und die davon geographisch abhängigen Kanten aus den Mengen $\text{confl}(\alpha)$. Nicht mehr benötigte Blockings und Zeitfenster werden gelöscht.

5.5. Zur Komplexität

Normalerweise kann das SPPTW, solange P ungleich NP nicht in polynomialer Zeit ausgeführt werden. In diesem Fall aber besteht die Kostenfunktion nicht nur aus Fahrtzeiten, sondern auch aus den Wartezeiten. Daher liegt das Problem für diesen Spezialfall in P und ist mit dem angeführten Algorithmus in Polynomialer-Zeit lösbar.

Die Kosten (Fahrt- und Wartezeiten) korrelieren mit der unteren Grenze der Labelintervalle. Die Differenz ist immer eine additive Konstante, nämlich die Startzeit. Dadurch dominiert ein Label ein anderes alleine, wenn dieses über dasselbe Zeitfenster expandiert wurde und einen geringeren Distanzwert d als jenes ausweist.

Die Anzahl der Labels pro Kante ist dadurch begrenzt: nämlich durch die Anzahl der Zeitfenster aller eingehenden Kanten. Dadurch ist aber auch die Anzahl der Iterationen in jeder Schleife begrenzt, nämlich, entweder durch die Anzahl der Kanten oder durch die Anzahl der Zeitfenster, so dass der Algorithmus in polynomialer Zeit mit einem kürzesten Weg terminiert oder mit der Benachrichtigung, dass kein Weg existiert.

Das Ergebnis eines Tests mit einem Graphen mit 30.000 Kanten bestätigt, dass der dynamic approach dem static approach überlegen ist. Die Rechenzeit für eine dynamische, konfliktfreie Route lag im Durchschnitt deutlich unter einer halben Sekunde. Damit ist der Algorithmus für ein Echtzeitsystem geeignet. Die Overall-Transit-Time lag bei der dynamischen Routenberechnung unter derjenigen der statischen Routenberechnung.

6. Schlussbemerkung

Der angeführte Algorithmus berechnet einen zeitpunktbezogenen Kürzesten Weg in polynomialer Zeit. Im Plenum kam die Behauptung auf, dass unter Umständen der static approach eine schnellere Route berechnet, da die Kanten nicht unnötig lange für andere AGVs gesperrt werden und sich dadurch die Fahrtzeit verkürzt. Dies ist nur anzunehmen, wenn davon auszugehen ist, dass die Fahrtzeiten der Kanten übermäßig groß gewählt werden. Liegen diese aber z. B. im Sekundenbereich ist nicht mit einer Verzögerung zu rechnen, da die gesperrte Kante damit der Länge des AGVs mit einem Sicherheitsabstand entspricht. Beim static approach werden zwar keine Kanten gesperrt, aber das Sperren der Transponder (während sich das AGV durch das AGS bewegt) ist letztlich genau dasselbe. Wird also das Modell in dieser Hinsicht optimiert, ist mit keiner Effizienzeinbuße zu rechnen. Stattdessen wird durch die unnötig gewordenen Maßnahmen zur Verhinderung oder Auflösung von Dead- und Livelocks und durch eine flexible und ausgewogene Verkehrsführung die durchschnittliche Fahrdauer der AGVs reduziert.

Literaturverzeichnis

Literatur

- [1] Möhring, Rolf H. / Köhler, Ekkehard / Gawrilow, Ewgenij, Conflict-free Real-Time AGV Routing, TU-Berlin, 2005
- [2] Bernd, Thomas, Autonomes Fahren, Universität Koblenz-Landau, 2003
- [3] Stenzel, Björn, Schneller ans Ziel: wie die Mathematik den Weg weist. TU-Berlin, 2006
- [4] Container-Terminal-Altenwerder, <http://www.terminal-altenwerder.com>, Download am 17. Mai 2006, 12.30
- [5] Wikipedia - CTA, http://de.wikipedia.org/wiki/Containerterminal_Altenwerder, Download am 17. Mai 2006, 12.30
- [6] Wikipedia - AGV, http://de.wikipedia.org/wiki/Fahrerloses_Transportfahrzeug, Download am 17. Mai 2006, 12.50
- [7] HHLA AG, <http://www.hhla.de/Geschaeftsfelder/index.jsp>, Download am 24. Mai 2006, 9.00
- [8] COGA, <http://www.math.tu-berlin.de/coqa/projects/traffic/agvrouting>, Download am 24. Mai 2006, 8.50