

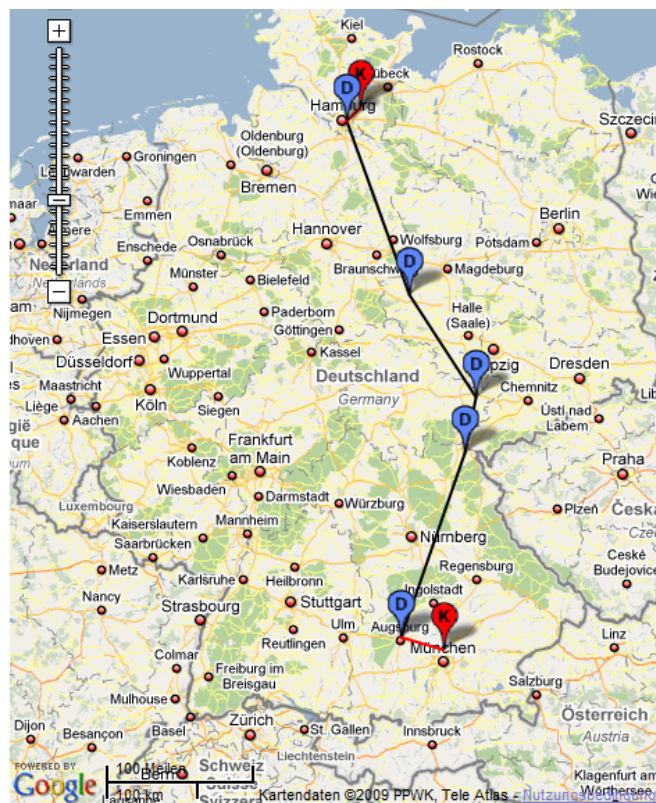
IWI Diskussionsbeiträge # 38 (11. Januar 2010)¹



ISSN 1612-3646

Plattformunabhängiges Softwareengineering eines Transportmodells zur ganzheitlichen Disposition von Strecken- und Flächenverkehren

Tim Rickenberg², Hans-Jörg von Mettenheim³ und Michael H. Breitner⁴



¹ Kopien oder eine PDF-Datei sind auf Anfrage erhältlich: Institut für Wirtschaftsinformatik, Leibniz Universität Hannover, Königsworther Platz 1, 30167 Hannover (www.iwi.uni-hannover.de).

² Cand. Dipl. Wirtsch.-Ing. Tim Rickenberg (rickenberg@iwi.uni-hannover.de).

³ Dr. rer. pol., Diplom-Mathematiker, Institut für Wirtschaftsinformatik (mettenheim@iwi.uni-hannover.de).

⁴ Professor für Wirtschaftsinformatik und Betriebswirtschaftslehre und Direktor des Instituts für Wirtschaftsinformatik (breitner@iwi.uni-hannover.de).

Inhaltsverzeichnis

1	EINFÜHRUNG	3
2	GRUNDLEGENDE STRUKTUR VON TRANSPORTNETZEN	4
3	KONZEPT.....	5
4	INTERNETPLATTFORM	6
5	IMPORTPROGRAMM JIMPORT	9
6	ROUTINGPROGRAMM JROUTER	9
6.1	GRAPHISCHES INTERFACE.....	10
6.2	VISUALISIERUNG.....	12
6.3	EINSTELLUNGEN.....	17
6.4	GRUNDLAGEN DES IMPLEMENTIERTEN ROUTINGALGORITHMUS	20
6.5	ABLÄUFE IM ROUTINGALGORITHMUS.....	25
6.6	SPEICHERN DER ROUTINGERGEBNISSE	29
7	FAZIT UND AUSBLICK	33

Abstract

Die vorliegende Arbeit befasst sich mit der automatisierten Generierung optimierter Transportketten zur Disposition von Strecken- und Flächenverkehren in komplexen Transportnetzen. Trotz fortschreitender Informationstechnologien wird die Disposition von Transportaufträgen im Allgemeinen noch manuell durch einen Disponenten durchgeführt. Im Laufe dieser Arbeit werden Operations Research Verfahren wie z.B. Heuristiken implementiert, um diese umfassende Aufgabenstellung automatisch zu lösen. Unter Berücksichtigung der Restriktionen und Nebenbedingungen (wie Kapazitäten und Zeitfenster) werden durch den erstellten Routingalgorithmus Potentiale zur Kostenreduzierung aufgedeckt. Um dem Nutzer die erstellten Forschungsergebnisse bereitzustellen, wurde eine Internetplattform erstellt, auf der die plattformunabhängige Software im Sinne des Open-Source Gedanken frei zugänglich gemacht wird. Die internetfähigen Java Programme ermöglichen Import und Erstellen von Logistikdaten sowie deren Verarbeitung im Routingalgorithmus und die Visualisierung.

Schlüsselwörter: Transportnetzwerk, Disposition, Restriktionen, A* (A Stern) Algorithmus, Generierung optimierter Transportketten, Heuristik

This paper deals with the automated generation of optimized transport chains for the disposition of near or far transfer orders within complex transport networks. Despite the progression of information technologies the disposition of transfer orders is generally still performed manually by a dispatcher. In the context of this study operations research methods were implemented to solve this comprehensive task automatically. Taking into consideration the restrictions and additional conditions (e.g. capacities and time frames) the routing algorithm is able to point out potentialities to reduce costs. In order to provide the research results to the user an internet platform was created on which the platform independent software can be accessed freely according to the open source concept. Those web-enabled java programs allow the import and creation of logistical data, plus calculation routes and their visualizations.

Keywords: Transport Network, Disposition, Restrictions, A* Algorithm, Conception of Optimized Transport Chains, Heuristics

1 Einführung

Die gerade gegenwärtige Wirtschaftskrise verdeutlicht wieder einmal, wie wichtig rationale und kostenminimale Prozesse für Unternehmen sind. Kein Unternehmen kann es sich auf Dauer leisten, unter einer Kostenstruktur zu operieren, die höher ist als die der Konkurrenz. Davon ist auch, oder gerade auch, die Logistikbranche nicht ausgenommen. Durch stark gesunkene Transportmengen müssen Logistikprozesse und Transportvorgänge optimiert werden und Kosten gesenkt werden, um wirtschaftlich und konkurrenzfähig zu arbeiten. Trotzdem wird oftmals in Sammelladungsspeditionen die Disposition manuell, also ohne den Einsatz von EDV allein auf Basis von persönlicher Erfahrung des Disponenten, durchgeführt⁵. Manuelle Disposition unter Zeitdruck erschwert das Finden einer ganzheitlichen Lösung und häufig wird das Zustandekommen der benutzten Route nicht hinterfragt.

Die Optimierung des Auftragspools in einem gegebenen Streckennetz durch Operations Research Verfahren ist ein wichtiger Ansatzpunkt für die Senkung der Kosten in der Logistikbranche. Dieser Ansatz wird in dem vorliegenden Diskussionsbeitrag, welcher im Laufe eines durch die AiF geförderten Projektes entstanden ist, untersucht. Ziel des Projektes mit dem Namen „Ganzheitliche Disposition von Strecken- und Flächenverkehren durch kombinierten Einsatz modifizierter Operations Research Verfahren“ (kurz „DiSFOR“) ist die automatische Generierung von komplexen Transportketten für den Güterversand durch Einsatz von Informations- und Kommunikationstechnologien. Um diese umfassende Aufgabe in einem begrenzten Zeitraum mit moderner IT lösen zu können, ist der Einsatz von Heuristiken unabdinglich. Die Nebenbedingungen des Transportproblems ergeben sich aus den Kapazitäten (bspw. Fläche und Gewicht) und den zeitlichen Restriktionen. Als Kostenparameter werden sowohl Hauptlaufkosten als auch Vor- und Nachlaufkosten sowie Umschlagkosten betrachtet. Die feste Zuordnung von Versendern und Empfängern zu Depots wird aufgrund des ganzheitlichen und zentralen Planungsansatzes aufgehoben. Dieser Ansatz unterstützt Netzwerkstrukturen mit mehrfach gebrochenen Hauptläufen durch Umschläge an Depots als auch Netzstrukturen mit Direktverkehren. Gerade häufig gebrochene Verkehre erhöhen die Komplexität und erschweren die manuelle Planung für den Disponenten.

Das Ergebnis dieser Studie soll die operative Disposition unterstützen und somit die Verteilung von Aufträgen auf ein bestehendes Transportnetz ermitteln. Dabei sollen wesentliche Potentiale zur Kostenreduzierung aufgedeckt werden. Vor allem kleinen und mittelständischen Unternehmen mit begrenztem Forschungsbudget kommen die Forschungsergebnisse des Projektes zu Gute, da die Ergebnisse als Open-Source Software zu Verfügung gestellt werden. Im Rahmen des DiSFOR Projektes wurde bereits ein lauffähiger, umfangreicher Demonstrator auf Basis von Microsoft Access erstellt. Dieser ist jedoch an Microsoft Access gebunden und nicht internetfähig. Um die Ergebnisse dieser Studie besser veröffentlichen zu können, wurden zwei plattformunabhängige und internetfähige Demonstratoren erstellt. Diese machen die Funktionalität dieser Studie dem Nutzer auf einer Internetplattform praktisch und komfortabel zugänglich.

Im nachfolgenden Kapiteln wird als Einstieg die grundsätzliche Struktur der zugrunde gelegten Transportnetze beschrieben und das Konzept dieser Studie skizziert. Anschließend werden die Internetplattform und beide Demonstratoren erläutert. Im Anschluss daran befindet sich ein Fazit mit einem Ausblick.

⁵ Vgl. Janz (2002).

2 Grundlegende Struktur von Transportnetzen

In Logistikbetrieben werden durch Disponenten und Planer Transportaufträge in einem Transportnetz generell manuell geplant, koordiniert und überwacht. Das Transportnetz besteht dabei in der Regel aus Depots (sogenannten primären Knoten) zwischen denen Transportfahrzeuge auf festgelegten Routen und meist auch unter vorgegebenen Zeitplänen verkehren (sogenannten Kanten). Von Kundenstandorten (sekundären Knoten) aus werden Transportaufträge versendet. Dazu werden die Transportgüter von Nahverkehrsfahrzeugen vom Standort des Versenders abgeholt und zum Startdepot gebracht. Dort werden sie auf Lastkraftwagen verladen (bzw. auf eine Kante umgeschlagen). Bei gebrochenen Verkehren kann das Transportgut auf dem Weg zum Zieldepot ein oder mehrmals an Depots umgeschlagen werden. Am Zieldepot angekommen, kann das Transportgut durch ein Nahverkehrsfahrzeug zum Empfänger befördert werden. Bei der Planung der Transportaufträge auf das Transportnetz müssen Restriktionen wie Zeit, Kosten und Kapazitäten beachtet und eingehalten werden. Wie hier deutlich wird, kann die manuelle Disposition in komplexen Transportnetzen schnell unübersichtlich oder sogar unüberschaubar werden.

Die Begrifflichkeiten Knoten und Kanten stammen aus der Graphentheorie. Knoten sind Quellen und Senken von Ladungen und sind über Kanten in Form von Netzwerkprozessen miteinander verknüpft. Die Folgende Abbildung zeigt schematisch den Aufbau eines Transportnetzes zusammen mit den in diesem Diskussionsbeitrag benutzen Begriffen:

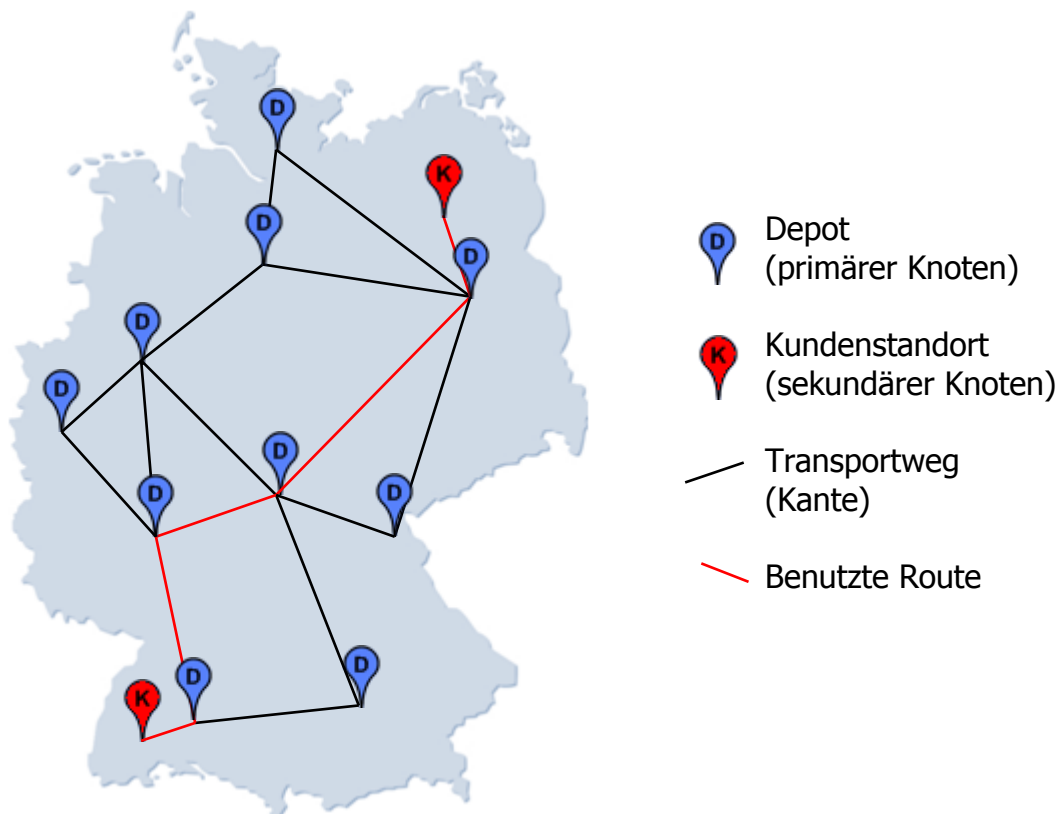


Abbildung 1: Schema eines Transportnetzes⁶

⁶ Karte auf Basis von Lenz (2009).

3 Konzept

Im Rahmen dieser Studie wurde eine Internetplattform zur Bereitstellung der Demonstratoren für den Endnutzer erstellt. Dieses sogenannte DiSFOR Internet-Portal kann unter folgender Internetadresse aufgerufen werden:

<http://www.iwi.uni-hannover.de/georouting/>

Auf diesem Portal werden die komplette Funktionalität der beiden Demonstratoren sowie ein Beispieldatenpool zum Testen zur Verfügung gestellt. Der genaue Aufbau des Internet-Portals kann dem nachfolgenden Kapitel entnommen werden.

Bei der Entwicklung der Demonstratoren und des Internet-Portals wurde auf folgende Aspekte besonderen Wert gelegt:

- Freeware und Open-Source: Die Nutzung von kostenfreier Software, Datenquellen und Schnittstellen wurde angestrebt. So wurde eine kostenlose Schnittstelle zum Import von Geodaten⁷ implementiert als auch bei der Entwicklung kostenfreie Produkte und Entwicklungsumgebungen eingesetzt. Dadurch kann eine Distribution ohne Lizenzgebühren erfolgen.
- Portierbarkeit: Das Programm wurde weitgehend plattformunabhängig⁸ gestaltet, so dass es auf verschiedenen Betriebssystemen (Microsoft Windows Betriebssysteme, Linux Distributionen) ausgeführt werden kann.
- Internetfähigkeit: Die Demonstratoren können direkt von der Internetseite aus dem Browserfenster gestartet werden. Die vorherige Installation oder das vorherige Konfigurieren der Programme ist nicht nötig.
- Variabler Import von Daten: Durch das Erstellen eines Importprogramms werden die einfache und formatunabhängige Einbindung von Daten eines Transportnetzes und deren Ausgabe in einem fest definierten Format ermöglicht. So können bereits bestehende Datensätze eines Transportnetzes automatisiert erfasst und zeitintensives manuelles Eingeben erübrigt werden.
- Erstellung eines leistungsstarken Berechnungsmodells: Im Routingprogramm JRouter wird ein Routingalgorithmus implementiert, der die Aufträge in einem gegebenen Transportnetz unter den gegebenen Einschränkungen mit Hilfe von Heuristiken in einer überschaubaren Zeitspanne optimiert.
- Visualisierung: Zur besseren Übersichtlichkeit, zum besseren Verständnis und zur Überprüfung der Ergebnisse können Depot-, Kundenstandorte, Kanten und Aufträge graphisch auf dem Bildschirm dargestellt werden. Auch komplette Auftragsabwicklungen vom Startkunden bis hin zum Zielkunden können dargestellt werden.
- Test an Hand von Realdaten: Nach Implementierung des Programms fand eine Verifizierung und Validierung des Programms an Hand von Realdaten statt.

Nachfolgend werden die Zusammenhänge zwischen den beiden erstellten Programmen mit ihren Datenflüssen erklärt, sowie weiter unten in einer Abbildung skizziert. Das Importprogramm JImport ermöglicht die einfache und formatunabhängige Einbindung von Transportnetzdaten. Als Datenquellen dienen Microsoft Access-Datenbanken und Microsoft Excel-Tabellen, die mit der Standardsoftware Microsoft Office erstellt werden können. Eine Installation der zugehörigen Microsoftprodukte ist jedoch nicht zwingend erforderlich. Für

⁷ Der Import von Geodaten erfolgt aus der Google Maps API. Vgl. Google (2009), Nazmul (2008).

⁸ Die Plattformunabhängigkeit wurde durch die Nutzung von Java als Programmiersprache erreicht. Grundlagen und Syntax dieser Sprache können der Literatur entnommen werden. Vgl. Baltes-Götz (2003), Lafore (2003), Ullenboom (2006), Wolmeringer (2007).

die Depot- und Kundenstandorte werden Geodaten sowie die Entfernungen der Kanten über eine kostenfreie Schnittstelle zur Verfügung gestellt. Auch das manuelle Erstellen und Bearbeiten von Transportnetzdaten wird unterstützt. Schließlich werden mit dem Programm die Daten des Transportnetzes in einem fest definierten Format ausgegeben. Die vom Importprogramm gespeicherten Daten können von dem Hauptprogramm JRouter aufgenommen und verarbeitet werden. Im Hauptprogramm ist ein Routingalgorithmus implementiert, der die Aufträge in einem gegebenen Transportnetz (Knoten und Kanten) unter Berücksichtigung der Nebenbedingungen (Zeit und Kapazitäten) optimiert. Hierbei wird auch der Kostenaspekt der Aufträge beachtet und die Visualisierung ermöglicht. Abschließend können die Routingergebnisse inklusive eines detaillierten Fahrplans und einer Visualisierung der Auftragsabwicklung ausgegeben werden. Gegebenenfalls kann der Routingalgorithmus auch in einem externen Programm angestoßen werden. In Frage kommen hier Optimierungsprogramme wie ILOG CPLEX als auch andere ausführbare Dateien.

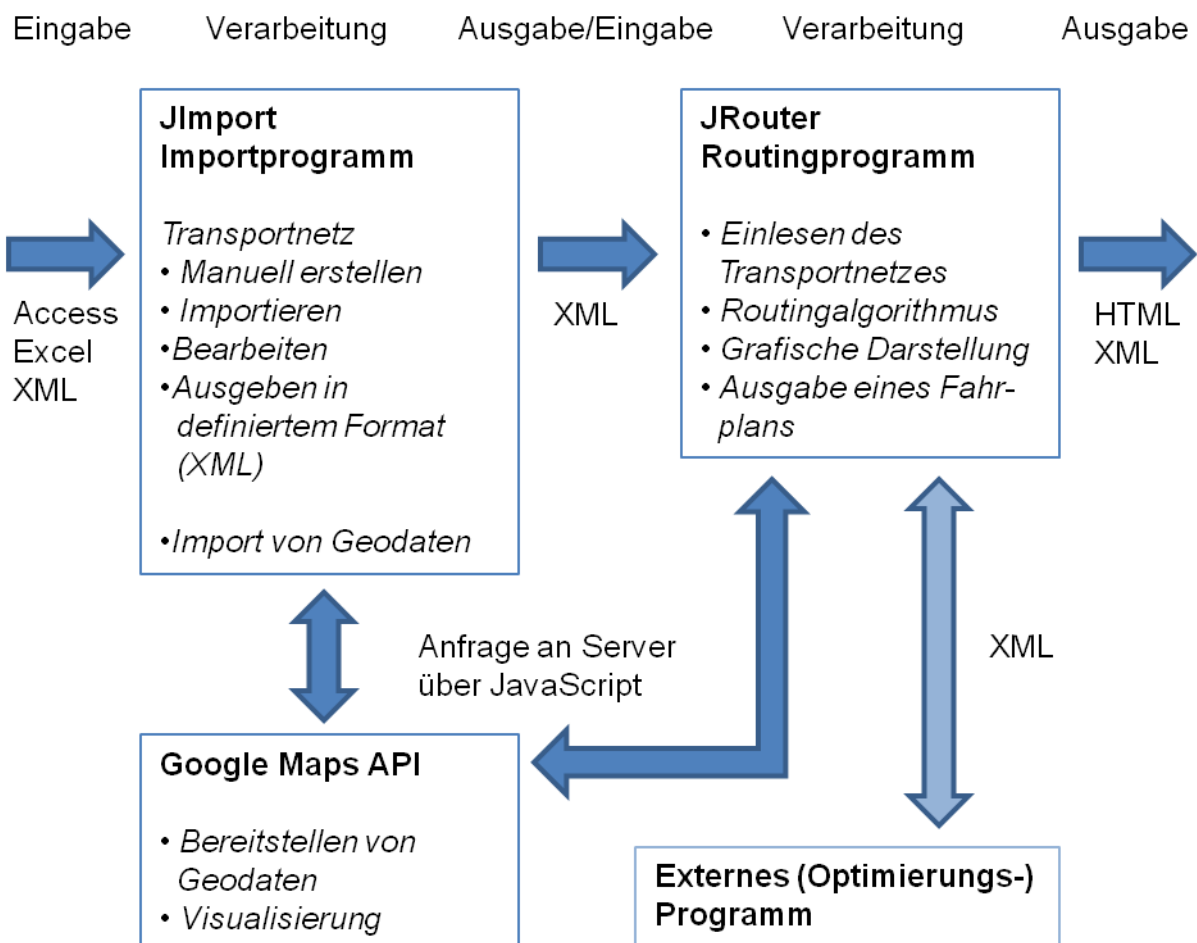


Abbildung 2: Zusammenhänge zwischen den erstellten Programmen

4 Internetplattform

Die Internetplattform auf HTML-Basis ist bewusst schlicht und übersichtlich gehalten, um die Funktionalität und die Plattformunabhängigkeit zu gewährleisten. Die nachfolgende Abbildung zeigt die Startseite des Internetportals:

DiSFOR internetfähige Software **JImport** und **JRouter**

Das DiSFOR Projekt

Im Rahmen des Forschungsvorhabens "Ganzheitliche Disposition von Strecken- und Flächenverkehren durch kombinierten Einsatz modifizierter Operations Research-Verfahren (DiSFOR)", gefördert durch die AiF, wird eine Methodik zur automatischen Generierung von Transportketten im Sammelladungsverkehr nach ganzheitlichen Kriterien entwickelt. Das Modell soll dabei nicht die günstigste Transportkette für einen einzigen Transport ermitteln, sondern einen Pool von Transportaufträgen so aufeinander abstimmen, dass gesamthaft ein effizientes Ergebnis erzielt wird.

Dieses Internet-Portal erlaubt ohne vorherige Installation die plattformunabhängige Nutzung der Java Webstart Programme. Eine umfangreichere Softwarelösung mit erhöhtem Funktions- und größerem Parametrisierungsumfang wurde im Verlaufe des Projektes auf Basis von Microsoft Access und Visual Basic for Applications erstellt.



Einleitung und Dokumentation

Die Dokumentation wird in Form von einer .pdf Datei angeboten und umfasst das Konzept dieser Studie sowie Nutzer- und Entwicklerdokumentation. Das Lesen vor der ersten Anwendung ist nicht zwingend erforderlich, jedoch sehr sinnvoll.

[Öffnen der Dokumentation](#)

JImport

Das Importprogramm ermöglicht eine einfache und formatunabhängige Einbindung von Logistikdaten und gibt die Daten in einem fest definierten Format aus. Für die Depot- und Kundenstandorte werden Geodaten über eine kostenfreie Schnittstelle zur Verfügung gestellt. Die gespeicherten Daten können von dem Hauptprogramm (JRouter) aufgenommen und verarbeitet werden.

[Starten von JImport](#)

JRouter

Im Hauptprogramm ist ein Routingalgorithmus implementiert, der die Aufträge in einem gegebenen Logistiknetz (Knoten und Kanten) unter Berücksichtigung der Nebenbedingungen (Zeit und Kapazitäten) optimiert. Hierbei wird auch der Kostenaspekt der Aufträge beachtet und die Visualisierung wird ermöglicht.

[Starten von JRouter](#)

Beispieldaten

Um die Funktionsweise der Programme zu testen werden hier Beispieldaten zur Verfügung gestellt. Die Excel Dateien können mit dem Importprogramm eingelesen werden, bearbeitet und in einer XML Datei ausgegeben werden. Die Visualisierung und das Routing erfolgen nach dem Einlesen der XML Datei in das Hauptprogramm JRouter. Weiterhin wird auf dieser Homepage auch ein umfangreicherer Testdatenpool in Form einer XML zur Verfügung gestellt.

[Beispieldaten](#)

Kontakt: [Carsten Schulte](#), [Tim Rickenberg](#), [Hans-Jörg v. Mettenheim](#)
Last update: 12.11.2009

Abbildung 3: Screenshot der Startseite der DiSFOR Internetplattform

Auf der Internetplattform befindet sich eine kurze Einführung in die Thematik des DiSFOR Projekts und Verweise auf die involvierten Forschungsinstitutionen. Anschließend wird die Dokumentation zum Download angeboten, welche die beiden Demonstratoren JImport und JRouter umfangreich erläutert. Hyperlinks ermöglichen das Starten der beiden Programme und den Zugang zu den Beispieldaten, welche von der hinterlegten Internetseite heruntergeladen werden können. In der Fußzeile der Internetseite werden schließlich die Kontaktinformationen und das Änderungsdatum dargestellt

Wie bereits oben dargestellt, kann der Downloadbereich durch Klicken auf den Hyperlink „Beispieldaten“ aufgerufen werden. Hier werden Beispieldaten in Form von Microsoft Excel-Tabellen, einer Microsoft Access-Datenbank und einer XML-Datei⁹ bereitgestellt. Weiterhin wird erläutert, welches Format Microsoft Excel-Tabellen haben müssen, um von dem Importprogramm lesbar zu sein. Prinzipiell sind die Microsoft Excel und Access Daten zum Testen und als Einstieg in das Importprogramm JImport gedacht. Die auf der Internetseite verfügbare XML-Datei enthält die Daten, die mit dem Importprogramm aus der hier bereitgestellten Microsoft Access-Datenbank extrahiert wurden und im programmspezifischen Format gespeichert wurden. Um das Routingprogramm zu testen und die Funktionalitäten kennen zu lernen, ist das Laden der hier bereitgestellten XML-Datei der einfachste Weg. Der Downloadbereich der Internetplattform wird nachfolgend dargestellt:

DiSFOR Beispieldaten

Excel Dateien

Um vom Importprogramm (JImport) verarbeitet werden zu können, müssen Excel Dateien diese Anforderungen erfüllen:

1. Zeile: Überschriften der Spalten bzw. Spaltennamen
 2. Zeile - EOF: Zu importierende Daten
- Zellenformat: Muss auf „Standard“ gestellt sein, ggf. muss dazu erst auf „Text“ und dann auf „Standard“ gestellt werden
Dateiformat: Muss .xls (Excel 1997-2003) sein

[Depots](#)

[Kanten](#)

[Aufträge](#)

[Firmen](#)

Access Datenbank

[Access Datenbank](#)

XML Dateien

[XML umfangreich](#)

Kontakt: [Carsten Schulte](#), [Tim Rickenberg](#), [Hans-Jörg v. Mettenheim](#)
Last update: 22.10.2009

Abbildung 4: Screenshot des Downloadbereichs der DiSFOR Internetplattform

⁹ Die Grundlagen von XML können der einschlägigen Literatur entnommen werden. Vgl. Vonhoegen (2007).

5 Importprogramm JImport

Das Importprogramm wurde zum Erzeugen eines Transportnetzes programmiert. Dieses kann entweder manuell erstellt werden oder durch Anbindungen an verschiedene Datenquellen importiert werden. Auch das Laden und Bearbeiten von bereits im Importprogramm erzeugten Daten ist möglich. Dabei ist zwischen dem Laden und dem Importieren unterscheiden: Das Laden von Transportnetzdaten ist nur aus der programmspezifischen XML-Datei möglich, wogegen der Import durch das Einspielen von Daten aus Datenquellen ohne fest definiertes Format beschrieben wird.

Während der Laufzeit spielt das Importprogramm Daten aus verschiedenen lokalen Quellen, aber auch aus dem Internet, ein. So wird für das Erstellen und Importieren von Depots, Kanten und Kundenstandorten eine leistungsstarke Internetverbindung benötigt, um Geodaten und Entfernungen automatisch zu bestimmen. Falls es dabei bspw. durch einen Proxy Server zu Problemen kommt, kann ein Administrator oder ggf. die IT-Fachabteilung zu Rate gezogen werden.

Letztendlich ist das Ziel der Nutzung des Importprogramms das Erstellen einer XML-Datei, die alle relevanten Daten für den Anstoß des Routingalgorithmus enthält. In diesem Diskussionsbeitrag wird der Fokus auf das Routingprogramm JRouter gelegt. Für weitere Informationen über das Importprogramm und das Erstellen bzw. Importieren von Daten eines Transportnetzes, kann die ausführliche Dokumentation von der vorgestellten Internetseite bezogen werden.¹⁰



Abbildung 5: Logo Importprogramm JImport

6 Routingprogramm JRouter

Das Routingprogramm wurde zur Optimierung der Disposition von Transportaufträgen auf ein gegebenes Transportnetzwerk unter Berücksichtigung der Nebenbedingungen erstellt. Die dafür erforderlichen Daten über das Transportnetz werden durch eine XML-Datei aufgenommen, die vorher durch das Importprogramm erstellt wurde.

Der Routingalgorithmus unterstützt die Variierung von Kostenparametern, zeitlichen Restriktionen und Transportparametern sowie die Anpassung der Heuristik und Knotenanbindung. Letztendlich ist das Ziel der Nutzung des Routingprogramms die Berechnung der kostenminimalen Abwicklung des Auftragspools. Das Visualisieren von Knoten, Kanten, Aufträgen oder einer Auftragsabwicklung sowie die Ausgabe der Routingergebnisse als XML oder HTML-Datei sind wichtige Funktionen des Programms. In den nachfolgenden Unterkapiteln wird zuerst das graphische Interface erläutert, die Möglichkeiten der Visualisierung erklärt, die Einstellungsmöglichkeiten erläutert, die Grundzüge und Abläufe innerhalb des Routingalgorithmus skizziert sowie das Speichern

¹⁰ Als Einstieg in die im Importprogramm verwendeten Datenstrukturen und –algorithmen kann Standardliteratur zu Rate gezogen werden. Vgl. Elmasri (2002), Lusti (2003), Ottmann (1996), Vossen (2008).

von Daten beschrieben. Schwerpunkt wurde in diesem Kapitel auf die Erklärung anhand von Screenshots und Beispielen gelegt, um dem Nutzer das Programm auf einfachem Wege zu verdeutlichen.

Das Routingprogramm nutzt lokal gehaltene Daten aus der erzeugten XML-Datei, bezieht aber auch Daten aus dem Internet. Zur Visualisierung werden dynamisch Google Maps Karten erzeugt. Dafür ist eine leistungsstarke Internetverbindung nötig. Zur reinen Berechnung des Routings ist keine Internetverbindung nötig, dies kann auch offline erfolgen. Falls es bei der Visualisierung bspw. durch einen Proxy Server zu Problemen kommt, kann ein Administrator oder ggf. die IT-Fachabteilung helfen.



Abbildung 6: Logo Routingprogramm JRouter

6.1 Graphisches Interface

Das Routingprogramm ist in Form einer Single-Frame Anwendung erstellt und hat somit nur ein Hauptfenster. Unten ist ein Screenshot des Programms in dem Zustand abgebildet, in dem es sich direkt nach dem Starten befindet:

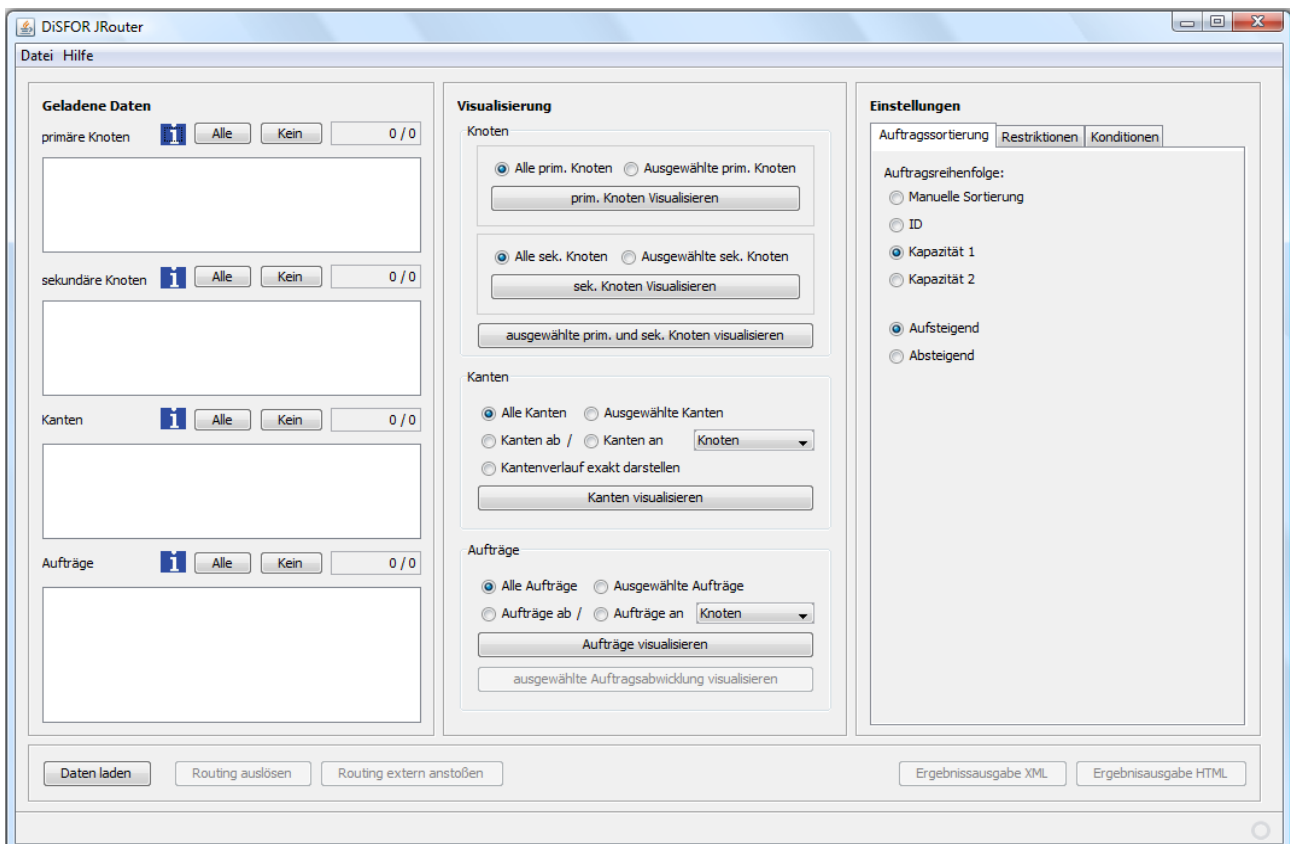


Abbildung 7: Hauptfenster Routingprogramm

Im linken Drittel des Fensters werden die geladenen Daten getrennt nach primären und sekundären Knoten sowie Kanten und Aufträgen dargestellt. Hier wurden noch keine Daten geladen, somit sind die Container leer. Im mittleren Drittel sind verschiedene Schaltflächen zur Visualisierung platziert. Oben sind Buttons zur Visualisierung von Depots und Kundenstandorten angeordnet, gefolgt von Buttons zur Visualisierung von Kanten und von Aufträgen. Im rechten Drittel befinden sich die Einstellungsmöglichkeiten des Programms, die durch mehrere Reiter unterteilt sind. Unterhalb der drei Bereiche sind Buttons zum Aufruf wichtiger Funktionen des Programms angeordnet. Dort lassen sich Daten laden, das Routing anstoßen oder auch die Ergebnisse des Routings in einer Datei ausgeben.

Durch das Klicken des Buttons „Daten laden“ können die Transportnetzdaten aus einer XML-Datei in das Programm eingespielt werden. Dadurch werden die Listen im linken Drittel des Programms mit den Daten aufgefüllt. Die nachstehende Abbildung zeigt den linken Teil der graphischen Oberfläche nach dem Laden von Daten:

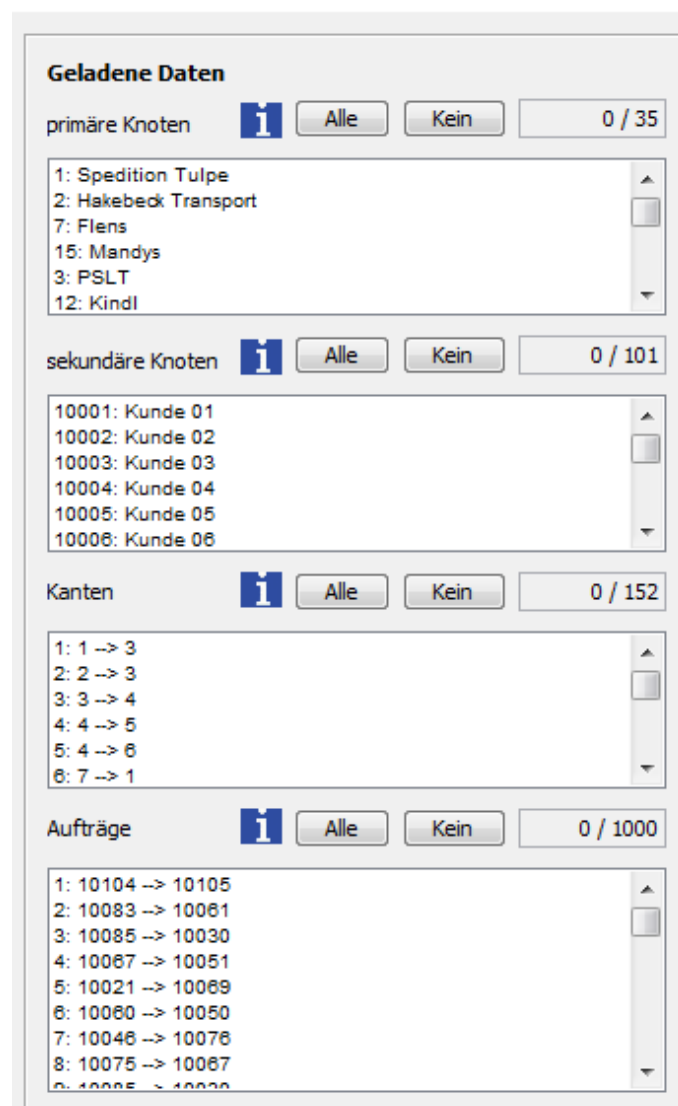


Abbildung 8: Schaltflächen „Geladene Daten“ JRouter

In den oben abgebildeten Listen können Objekte markiert werden. Die Gesamtanzahl an Objekten und die Anzahl an derzeit markierten Objekten werden jeweils in einem Textfeld angegeben. Außerdem können durch Knopfdruck entweder alle oder kein Objekt

markiert werden. Wenn nur ein Objekt in einer Liste markiert ist, können durch einen Klick auf den blauen Informationsbutton die Informationen zum markierten Objekt in einer Messagebox dargestellt werden. Weiterhin dient die Auswahl in den Listen zur Auswahl der zu visualisierenden Objekte.

In den nachfolgenden Unterkapiteln werden die Bereiche der Visualisierungs- und Einstellungsmöglichkeiten genauer betrachtet.

6.2 Visualisierung

Im mittleren Drittel des Programms befinden sich mehrere verschiedene Visualisierungsmöglichkeiten. Durch den Klick auf einen der Buttons wird eine Google Maps Karte mit der jeweilig gewünschten Visualisierung erzeugt. Das mittlere Drittel des Programms wird unten in Originalgröße zur besseren Übersichtlichkeit noch einmal dargestellt:

Visualisierung

Knoten

Alle prim. Knoten Ausgewählte prim. Knoten

prim. Knoten Visualisieren

Alle sek. Knoten Ausgewählte sek. Knoten

sek. Knoten Visualisieren

ausgewählte prim. und sek. Knoten visualisieren

Kanten

Alle Kanten Ausgewählte Kanten

Kanten ab / Kanten an Spedition Tulpe

Kantenverlauf exakt darstellen

Kanten visualisieren

Aufträge

Alle Aufträge Ausgewählte Aufträge

Aufträge ab / Aufträge an Kunde 01

Aufträge visualisieren

ausgewählte Auftragsabwicklung visualisieren

Abbildung 9: Schaltflächen „Visualisierung“ JRouter

Ganz oben ist die Visualisierung der Knoten angeordnet. Es können entweder alle oder nur die in der links auf der Oberfläche angeordneten Liste markierten primären Knoten auf einer Karte visualisiert werden. Gleiches gilt für die sekundären Knoten. Mit der

Schaltfläche darunter können die ausgewählten primären und die ausgewählten sekundären Knoten auf einer Karte dargestellt werden. Zu beachten ist, dass die maximale Anzahl von gleichzeitig visualisierbaren Knoten in allen Fällen auf 40 begrenzt ist. Die Visualisierung sieht dann bspw. folgendermaßen aus:



Abbildung 10: Visualisierung von primären und sekundären Knoten

Bei der Visualisierung werden die primären Knoten mit einem roten Marker und einem D (Depot) gekennzeichnet und die sekundären Knoten durch einem blauen Marker mit einem F (Firma) beschrieben.

Für die Visualisierung der Kanten gibt es auch mehrere Möglichkeiten. Es können wieder alle Kanten oder nur die markierten Kanten auf einer Karte abgebildet werden. Außerdem

können alle Kanten abgehend von oder ankommend an ein Depot visualisiert werden. Für die Visualisierung aller Kanten abgehend von einem Depot kann dies bspw. folgendermaßen aussehen:



Abbildung 11: Visualisierung von Kanten

Auch hier gibt es in jedem Fall eine Begrenzung auf maximal 18 Kanten pro Karte. Vereinfachend werden die Kanten in Form von Linien dargestellt, die der Luftlinie zwischen Start- und Zieldepot entsprechen. Falls der exakte Straßenverlauf einer Kante dargestellt werden soll, muss „Kantenverlauf exakt darstellen“ ausgewählt werden. In diesem Fall kann jeweils nur eine Kante visualisiert werden. Ein Beispiel dafür ist nachfolgend abgedruckt:



Abbildung 12: Visualisierung des Straßenverlaufs einer Kante

Die Visualisierungsmöglichkeiten für die Aufträge entsprechen denen der Kanten mit der Ausnahme, dass der exakte Verlauf nicht dargestellt werden kann, stattdessen aber eine komplette Auftragsabwicklung auf einer Karte angezeigt werden kann. Es kann immer nur eine Auftragsabwicklung dargestellt werden, weiterhin muss auch erst der Routingprozess abgeschlossen sein und für den Auftrag muss eine Route gefunden worden sein. Unten ist beispielhaft eine Auftragsabwicklung abgebildet:



Abbildung 13: Visualisierung einer Auftragsabwicklung

Die Kanten (Hauptläufe) sind durch schwarze Linien dargestellt und der Vor- und Nachlauf ist jeweils durch rote Linien gekennzeichnet. Die Linien entsprechen hier den Luftlinien und nicht dem exakten Straßenverlauf.

Wie auf den Karten ersichtlich ist, sind die üblichen Google Maps Kontrollelemente zum Verschieben und Zoomen der Karte oben links auf der Karte angeordnet. Unten auf den Karten befindet sich der Maßstab und oben rechts kann die Darstellungsart der Karte ausgewählt werden.

6.3 Einstellungen

Im rechten Bereich der graphischen Oberfläche kann eine Vielzahl von Einstellungen vorgenommen werden. Die Einstellungsmöglichkeiten sind auf drei verschiedenen Reitern namens Auftragsortierung, Restriktionen und Konditionen platziert.

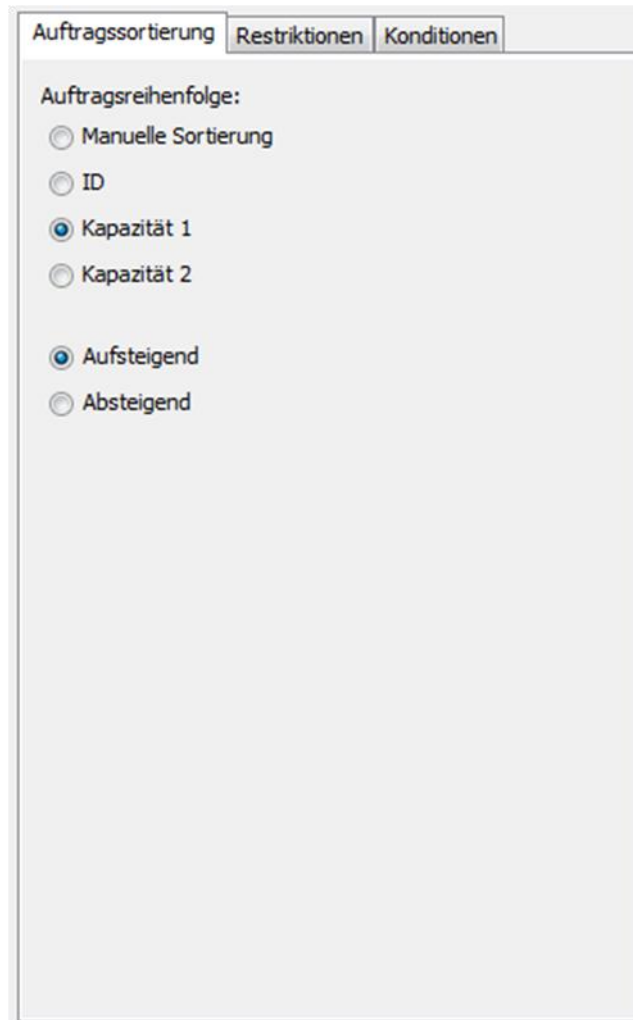


Abbildung 14: Schaltflächen „Einstellungen - Auftragsortierung“

Auf dem Reiter Auftragsortierung kann festgelegt werden, in welcher Reihenfolge die Aufträge vom Routingalgorithmus abgearbeitet werden sollen. Möglich ist die manuelle Sortierung sowie eine Sortierung nach AuftragID, Kapazität 1 oder Kapazität 2. Dabei kann noch festgelegt werden, ob die Sortierung aufsteigend oder absteigend erfolgen soll.

Unter dem Reiter Restriktionen können die Durchschnittsgeschwindigkeit im Hauptlauf sowie im Vor- und Nachlauf angepasst werden. Bei der Veränderung der Durchschnittsgeschwindigkeit im Hauptlauf muss beachtet werden, dass die Zeitfenster der Kanten für Abfahrt und Ankunft ggf. angepasst werden müssen. Außerdem kann auf diesem Reiter der Umrechnungskoeffizient für die Luftlinie zur Straßenentfernung und die Umschlagdauer angepasst werden.

Eine wichtige Einstellung ist das Anbinden von weiter entfernten Depots. Ist diese Option deaktiviert, so wird für jeden sekundären Knoten (Kundenstandort) immer nur der am nächsten gelegene primäre Knoten (Depot) angebunden. Dadurch wird die Anzahl der erfolgreich gerouteten Aufträge oft gesenkt. Ist diese Option jedoch aktiviert, werden für

jeden sekundären Knoten die drei am nächsten gelegenen primären Knoten angebunden. Dadurch wird ggf. die Entfernung des Vor- und Nachlaufes erhöht, jedoch können auch Aufträge abgewickelt werden, die sonst nicht realisierbar wären. Da für Vor- und Nachlauf die jeweils drei nächsten Depots angebunden werden, ergeben sich neun Kombinationen. Diese werden nacheinander im Routingalgorithmus abgearbeitet.

Das Aktivieren der Heuristik ist nicht zwingend erforderlich, aber sehr zu empfehlen. Die mögliche Anzahl der Durchläufe der Heuristik reicht von einem Durchgang bis zu neun Durchgängen. Als leistungsstarke Einstellung haben sich drei Durchläufe herauskristallisiert. Im Prinzip sperrt die Heuristik Kanten im Routingalgorithmus, die die erfolgreiche Abwicklung eines Auftrags verhindern und führt mit diesem veränderten Startvektor eine erneute Berechnung durch. Kanten können durch nicht passende zeitliche Restriktionen oder durch nicht ausreichende Kapazitäten gesperrt werden. Durch Aktivierung der Heuristik erhöht sich die Wahrscheinlichkeit, dass ein Auftrag erfolgreich geroutet werden kann. Jedoch erhöht sich, wie auch bei der Anbindung von weiter entfernten Depots, die Laufzeit des Algorithmus.

Ganz unten auf diesem Reiter können unterminierte Kanten aktiviert werden. Das bedeutet, dass die Zeitfenster für Abfahrt und Ankunft der Kanten sowie die Start- und Zielzeiten für die Aufträge mit Standardwerten überschrieben werden. Die Zeitfenster sind in diesem Falle sehr groß gehalten und die Kanten und Aufträge sind nicht vorterminiert. Mit dieser Option kann von der operativen Planung ein Schritt hin zur taktischen Planung gegangen werden. Die Abfahrts- und Ankunftszeiten sind über eine größere Zeitspanne variabel und somit können optimierte Abfahrt- und Ankunftszeiten für die Kanten automatisch generiert werden.

Die nachfolgende Abbildung zeigt die Einstellungsmöglichkeiten, die auf dem Reiter Restriktionen angeordnet sind:

Auftragsortierung	Restriktionen	Konditionen
Durchschnittsgeschwindigkeit im Hauptlauf (Kante)	<input type="text" value="65"/>	km/h
Durchschnittsgeschwindigkeit im Vor- und Nachlauf (Nahverkehr)	<input type="text" value="40"/>	km/h
Faktor Umrechnung Luftlinie	<input type="text" value="1,3"/>	
Umschlagsdauer	<input type="text" value="5,00"/>	min
Anbinden von weiter entfernten Depots erlauben (höhere Wahrscheinlichkeit für Routenfindung, längerer Vor-/Nachlauf)		
<input checked="" type="radio"/> Ja <input type="radio"/> Nein		
<input checked="" type="checkbox"/> Heuristik aktivieren		
Durchläufe	<input type="text" value="3"/>	
<input type="checkbox"/> Untermionierte Kanten (Untermionierte Kanten benötigen größere Zeitfenster)		
Früheste Abfahrt	<input type="text" value="2009-01-01 01:00:00.0"/>	
Späteste Abfahrt	<input type="text" value="2009-01-02 23:00:00.0"/>	
Früheste Ankunft	<input type="text" value="2009-01-01 01:30:00.0"/>	
Späteste Ankunft	<input type="text" value="2009-01-02 23:30:00.0"/>	
Start Auftrag	<input type="text" value="2009-01-01 06:00:00.0"/>	
Ende Auftrag	<input type="text" value="2009-01-02 20:00:00.0"/>	

Abbildung 15: Schaltflächen „Einstellungen - Restriktionen“

Die monetären Einstellungen befinden sich unter dem Reiter Konditionen. Dort können die Kosten für die Kanten und Umschläge sowie für die Vor- und Nachläufe angepasst werden. Für die Kanten können variable Kosten pro Kilometer und Kapazitätseinheit sowie pauschale Kosten angegeben werden. In Abhängigkeit davon werden dann unten in der Tabelle die Kosten für jede Kante generiert. Innerhalb der Tabellen können die Kosten manuell verändert werden, müssen dann aber mit dem Button „Kosten speichern“ für den Routingalgorithmus gesichert werden. Bei Veränderungen der variablen oder pauschalen Kosten für die Kanten oben auf dem Reiter müssen die Kantenkosten durch Klicken des Buttons „Kosten neu berechnen“ unten in der Tabelle aktualisiert werden. Auch für Vor- und Nachlauf sowie für den Umschlag können die variablen und pauschalen Kosten angegeben werden. Die variablen Kosten sind immer abhängig von der benötigten Kapazität des Auftrags und für den Vor- und Nachlauf auch von deren Entfernung. Dabei kann ausgewählt werden, mit welcher Kapazität die variablen Kosten korrelieren sollen. Die pauschalen Kosten sind immer unabhängig von der Kapazität und der Entfernung des Auftrags. Nachfolgend sind die Einstellungsmöglichkeiten bezüglich der Kosten abgebildet:

Auftragssortierung
Restriktionen
Konditionen

Kosten Kante pro km und Kapazitätseinheit

Kosten Kante pauschal

Kosten Vor- und Nachlauf pro km und Kapazitätseinheit

Kosten Vor- und Nachlauf pauschal

Umschlagkosten pro Kapazitätseinheit

Umschlagkosten pauschal

Kosten berechnen in Abhängigkeit von:

Kapazität 1

Kapazität 2

KantenID	Kosten
1	547.66
2	325.29
3	621.58
4	388.29
5	914.93
6	767.48
7	344.97
8	480.4

Abbildung 16: Schaltflächen „Einstellungen - Konditionen“

6.4 Grundlagen des implementierten Routingalgorithmus

Der Kern des Routingprogramms besteht aus einem Transportmodell und einem Routingalgorithmus. In der einschlägigen Literatur befinden sich viele Ansätze und Modelle, die hier in diese Arbeit eingeflossen sind oder in modifizierter Form implementiert wurden¹¹. Der im Hauptprogramm JRouter integrierte Routingalgorithmus benutzt zum Berechnen von Routen den A* (A Stern) Algorithmus. Dieser Algorithmus findet den kürzesten Pfad innerhalb von einem Netz von Knoten und gewichteten Kanten. Dabei wurde der Algorithmus an die hier vorliegende Aufgabenstellung angepasst. So werden die Restriktionen durch Kapazitäten und Zeitfenster sowie die Nebenbedingungen in die Optimierung mit einbezogen. Um die Laufzeit zu verringern greift der A* Algorithmus auf eine Heuristik zurück, die in jedem Fall die Lösung findet, falls eine existiert (vollständig). Außerdem wird immer die optimale Lösung unter kürzestem Zeitaufwand gefunden, falls eine besteht (optimal effizient). Die Idee und die Funktionsweise des Algorithmus kann der einschlägigen Literatur entnommen werden¹².

Neben der Heuristik des A* Algorithmus wurde in die Routenberechnung eine weitere Heuristik implementiert, die eine Kante für einen erneuten Berechnungsdurchlauf sperren

¹¹ Vgl. Domschke (1995), Domschke (2007), Eichner (1989), Suhl (2006).

¹² Vgl. Jenkins (2008), Lester (2006), Patel (2009), Otto (2004).

kann, falls diese die erfolgreiche Auftragsabwicklung verhindert. Somit wird der Startvektor verändert und ein weiterer Berechnungsdurchlauf des Auftrags kann mit veränderten Bedingungen durchgeführt werden. Die maximale Anzahl an Durchläufen der Heuristik ist im Programm einstellbar. Neben der hier implementierten Heuristik werden weitere interessante Ansätze in der Literatur beschrieben¹³.

Um ein besseres Verständnis für die Berechnungsweise einer Auftragsabwicklung zu erhalten, wird unten eine beispielhafte Auftragsabwicklung in einer Abbildung skizziert. Ausgehend vom Startkundenstandort findet der Vorlauf statt, durch den das Transportgut mit Nahverkehrsfahrzeugen zum Startdepot gebracht wird. Dabei muss die Startzeit des Auftrags beachtet werden. Am Startdepot wird das Transportgut in der Umschlagszeit umgeschlagen und auf eine Kante verladen. Die Abfahrtszeit der Kante wird durch die früheste (FAB) und die späteste Abfahrtszeit (SAB) determiniert, wobei die tatsächliche Abfahrtszeit immer zwischen diesen beiden liegen muss. Auch die Ankunftszeit dieser Kante wird analog dazu durch die früheste (FAN) und späteste Ankunftszeit (SAN) determiniert. Die Fahrtzeit der Kante wird bezüglich den Abfahrts- und Ankunftszeiten beachtet. Gegebenenfalls wird das Transportgut an weiteren Depots umgeschlagen und auf weitere Kanten verladen. Falls mehrere Kanten hintereinander geschaltet werden, müssen die Zeitfenster der Kanten zueinander beachtet werden. Selbstverständlich muss die vorherige Kante vor der Abfahrt der aktuellen Kanten angekommen sein. Sobald das Transportgut am Zieldepot ankommt, kann es auf ein Nahverkehrsfahrzeug umgeschlagen werden und durch den Nachlauf zum Zielkundenstandort gebracht werden. Dabei muss die späteste Zielzeit des Auftrags beachtet werden.

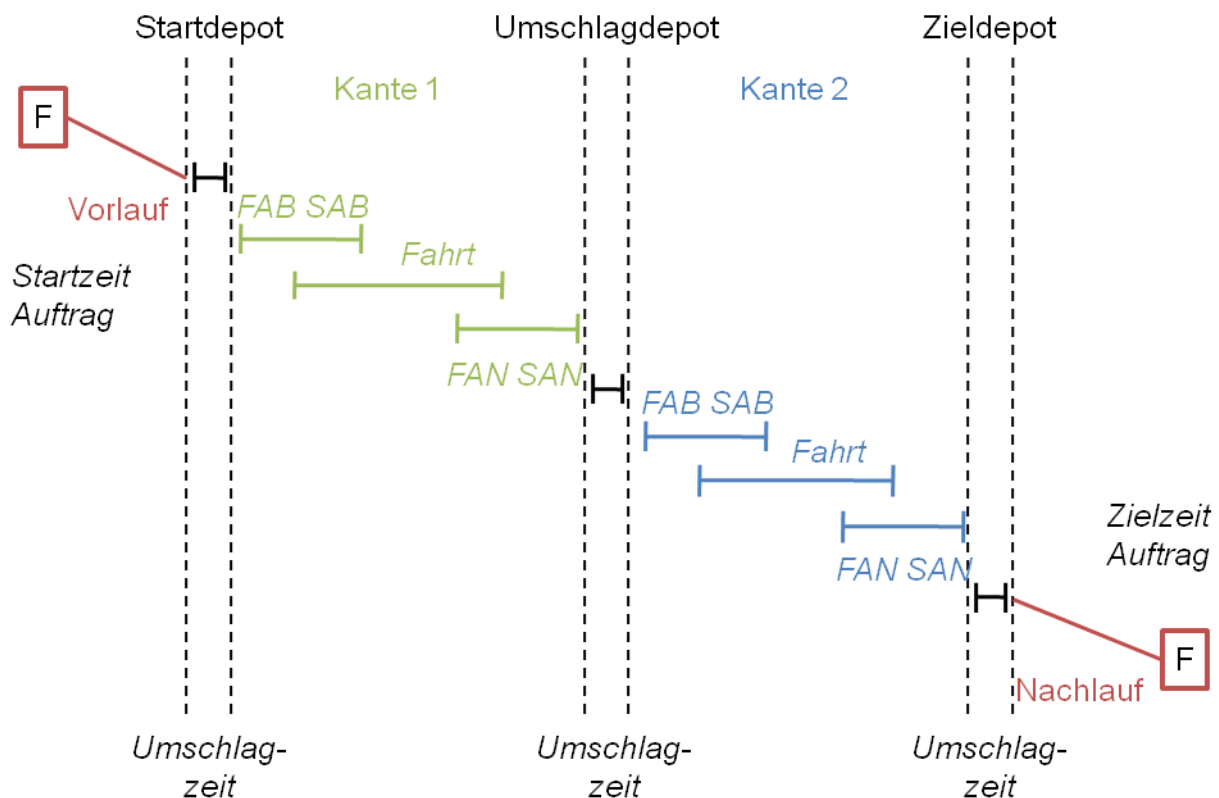


Abbildung 17: Darstellung des Prinzips einer Auftragsabwicklung

¹³ Vgl. Pankratz (2000), Möller (1991).

Während des Routingprozesses werden die Zeitfenster ständig überprüft. Falls grundsätzlich die Auftragsabwicklung mit der vorliegenden Route möglich ist, werden die Zeitfenster zueinander angepasst. Dabei werden die Zeitfenster der betroffenen Kanten soweit eingengt, so dass der jeweilige Auftrag erfolgreich abgewickelt werden kann. Durch die Anpassungen werden die früheste Abfahrts- und Ankunftszeit nach hinten geschoben und die späteste Abfahrts- und Ankunftszeit nach vorne geschoben. Falls es bei den Anpassungen zu Kollisionen kommt, wird durch die Heuristik eine erneute Routenberechnung angestoßen. Für diese wird dann durch die gesperrten Kanten zwangsläufig eine andere Route eingeschlagen. Vor allem für unterminierte Kanten werden viele Anpassungen der Abfahrts- und Ankunftszeiten vorgenommen und die Zeitfenster verschoben. Die Bedingungen für das Anpassen der Zeitfenster sind folgende:

$FAB \leq SAB$	Formel 1
$FAN \leq SAN$	Formel 2
$FAB + \text{Fahrzeit} = FAN$	Formel 3
$SAB + \text{Fahrzeit} = SAN$	Formel 4
$SAN_{\text{vorgaenger}} + \text{Umschlagzeit} \leq FAB_{\text{aktuell}}$	Formel 5
$SAN_{\text{aktuell}} + \text{Umschlagzeit} \leq FAB_{\text{nachfolger}}$	Formel 6

Nachdem oben stehend die Grundlagen des Routingalgorithmus grob umrissen wurden, wird nun die Ausführung dieser Funktion in dem Programm erläutert. Das Anstoßen des Routingprozesses kann erfolgen, nachdem die Daten über das Transportnetz aus einer XML-Datei geladen und sobald alle Restriktionen und Einstellungen im Programm wie gewünscht konfiguriert wurden. Die Schritte und der Fortschritt des Routingprozesses werden in einem sich öffnenden Fenster ausgegeben. Zuerst werden dort die Anbindung der sekundären Knoten an die primären Knoten und die ausgewählte Auftragsortierung ausgegeben und anschließend wird für jeden Auftrag die Routenberechnung schrittweise detailliert ausgegeben. Nach Beendigung des Routingprozesses werden eine Übersicht über die Kanten und eine Zusammenfassung der Routingergebnisse dargestellt. Der Berechnungsfortschritt wird dabei ständig über einen Fortschrittsbalken unten im Fenster abgebildet. Ein Screenshot dieses Fensters ist nachfolgend abgedruckt:

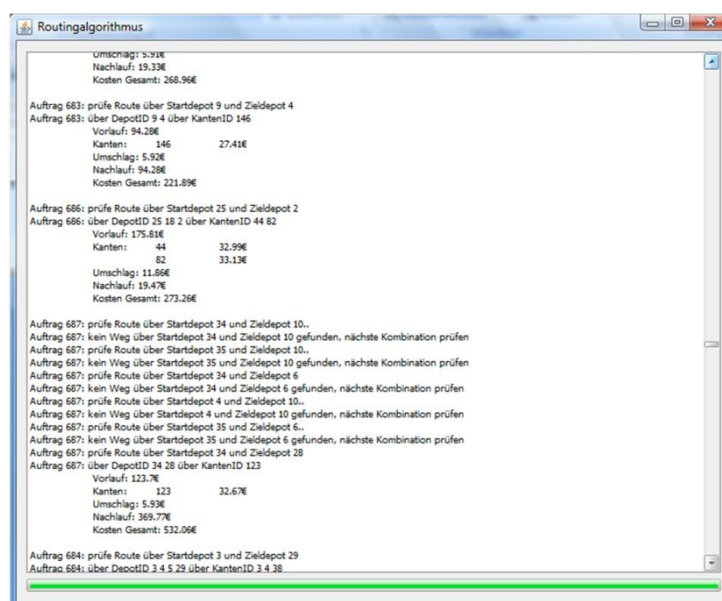


Abbildung 18: Fenster Berechnungsfortschritt Routing

Der Routingprozess wird in einer endlichen Zeit beendet. Unter Betrachtung einer gegebenen Netzwerkstruktur und der Kapazitäten hängen die Laufzeit des Programms und die Anzahl der erfolgreich berechneten Aufträge stark von den benutzten Einstellungen und den Nebenbedingungen ab. Ausschlaggebend sind Auftragsortierung, Geschwindigkeitseinstellungen, die Zeitfenster der Kanten, Konfiguration der Heuristik und Aktivierung der Anbindung von weiter entfernten Depots. Um einen Überblick über die benötigte Laufzeit des Programms zu erhalten, wird nachfolgend eine Übersicht über Laufzeittests gegeben. Als erstes wird die für die Tests benutzte Transportnetzstruktur aufgezeigt, dann wird die Rechenleistung des benutzten Systems beschrieben und anschließend werden die Ergebnisse des Benchmarks abgebildet.

Tabelle 1: Komplexität des Transportnetzes

Transportnetzobjekt	Anzahl
Primäre Knoten	35
Sekundäre Knoten	101
Kanten	152
Aufträge	1000

Das hier betrachtete Transportnetz besteht aus selbst konstruierten Testdaten und ist auf dem DiSFOR Internet-Portal als XML-Datei verfügbar. Somit sind Laufzeitvergleiche mit anderen Rechnern unkompliziert durchführbar. In diesem Auftragspool werden auch Aufträge versucht zu berechnen, die unter den gegebenen Restriktionen und Nebenbedingungen nicht realisierbar sind. Dies verschlechtert die Laufzeit des Programms. Das Verhältnis von erfolgreich berechneten Aufträgen zu der Gesamtanzahl an Aufträgen beläuft sich in dem oben genannten Transportnetz unter optimalen Einstellungen auf etwa 84%.

Nachfolgend werden die Leistungsdaten des zum Benchmark benutzten Computers abgebildet:

Tabelle 2: Leistung des zum Benchmark eingesetzten Computers

Systemkomponente	Leistungskapazität
Prozessor	Intel Core 2 Duo T5750 @ 2 GHz
Arbeitsspeicher	4 GB
Betriebssystem	Microsoft Vista Home Premium, 32-Bit mit SP2

Generell muss nun unterschieden werden, ob das Anbinden von weiter entfernten Depots aktiviert ist. Wie unten durch Vergleich der Performancetabellen sichtbar wird, verschlechtert sich durch die Aktivierung die Laufzeit stark, jedoch erhöht sich die Anzahl der erfolgreich berechneten Aufträge. Unten sind die Laufzeitergebnisse in Tabellen abhängig von der Anzahl an Durchläufen der Heuristik aufgetragen. Dabei entspricht die Einstellung mit einem Durchlauf der Heuristik einer deaktivierten Heuristik, da es in beiden Fällen nur einen Durchlauf gibt. Als erstes wird die Tabelle mit aktivierter Anbindung der weiter entfernten Depots dargestellt:

Tabelle 3: Performancetabelle (Anbindung weiter entfernter Depots aktiviert)

Anzahl Durchläufe der Heuristik	1	2	3	4	5	6	7	8	9
Dauer [s]	25,3	33,6	37,4	43,4	48,9	54,3	64,4	71,3	77,8
Gelöste Aufträge [%]	79,2	82,7	83,5	83,9	83,9	83,9	82,7	82,7	82,7
Verbesserung	0	+3,5	+4,3	+4,7	+4,7	+4,7	+3,5	+3,5	+3,5

Die Laufzeit verdreifacht sich von einem Durchgang mit deaktivierter Heuristik gegenüber einem Durchgang mit neun Durchläufen der Heuristik. Die Anzahl an erfolgreich gelösten Aufträgen erhöht sich durch Aktivierung der Heuristik, fällt hier aber bei zu vielen Durchläufen wieder ab. Die Verbesserung der erfolgreich gelösten Aufträge ergibt sich oben in der Tabelle im Vergleich zu einem Durchgang mit deaktivierter Heuristik und wird in Prozentpunkten angegeben.

Nachfolgend werden die Ergebnisse des Routingalgorithmus mit deaktivierter Anbindung weiter entfernter Depots präsentiert:

Tabelle 4: Performancetabelle (Anbindung weiter entfernter Depots deaktiviert)

Anzahl Durchläufe der Heuristik	1	2	3	4	5	6	7	8	9
Dauer [s]	9,2	12,4	14,1	16,1	17,9	19,4	20,9	22,4	23,9
Gelöste Aufträge [%]	42,6	48,7	50,8	51,2	51,3	51,3	51,3	51,3	51,3
Verbesserung	0	+6,1	+8,2	+8,6	+8,7	+8,7	+8,7	+8,7	+8,7

Bei Betrachtung des gesamten Spektrums an möglichen Durchläufen verdreifacht sich auch hier die Laufzeit nahezu. Die Anzahl der erfolgreich berechneten Aufträge steigt zuerst an, bleibt dann auf einem konstanten Niveau und fällt nicht ab.

Die Ergebnisse dieses Benchmarks werden anschließend durch Diagramme visualisiert. Das erste Diagramm zeigt den Zusammenhang zwischen der Anzahl an Durchläufen der Heuristik und der Laufzeit:

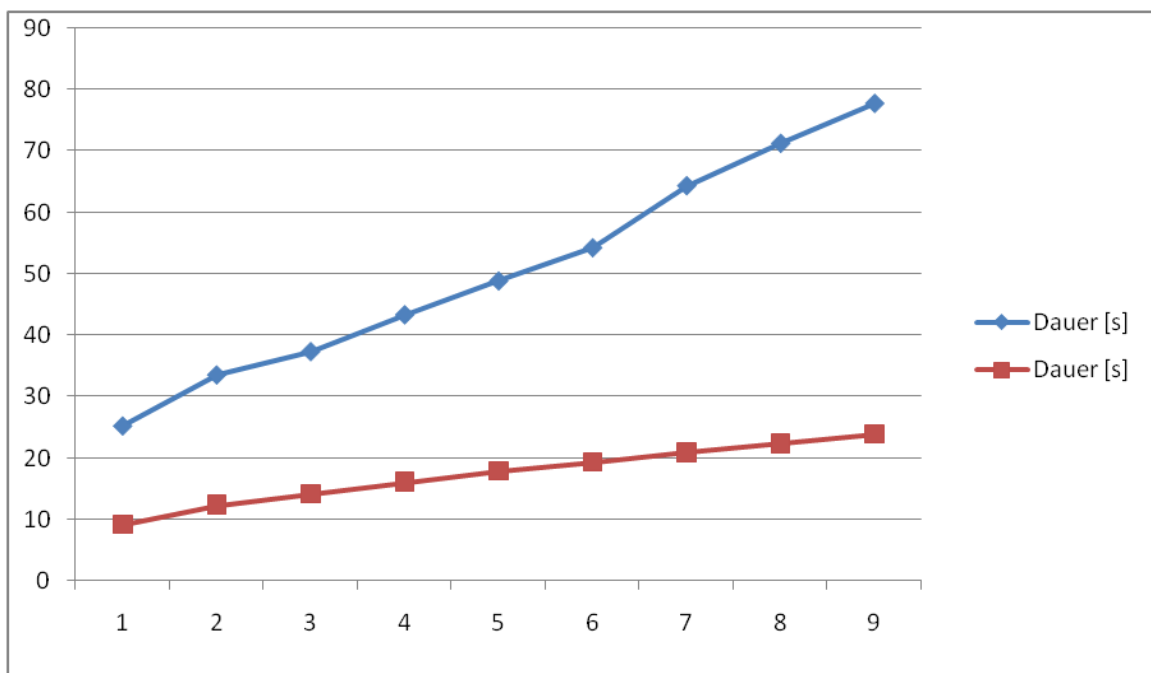


Abbildung 19: Diagramm Laufzeit Routingvorgang

Die Zeitdauern der Routingvorgänge mit aktivierter Anbindung weiter entfernter Depots sind blau markiert, die Zeiten ohne Anbindung weiter entfernter Depots sind rot markiert. Durch das Diagramm wird deutlich, dass die Erhöhung der Durchläufe der Heuristik einen linearen Anstieg der Laufzeit mit sich zieht. In dem folgenden Diagramm wird die Veränderung der Anzahl der erfolgreich abgewickelten Aufträge in Abhängigkeit zu den Durchläufen der Heuristik gesetzt. Die Veränderung wird gegenüber einem Durchlauf mit deaktivierter Heuristik in Form von Prozentpunkten ausgedrückt.

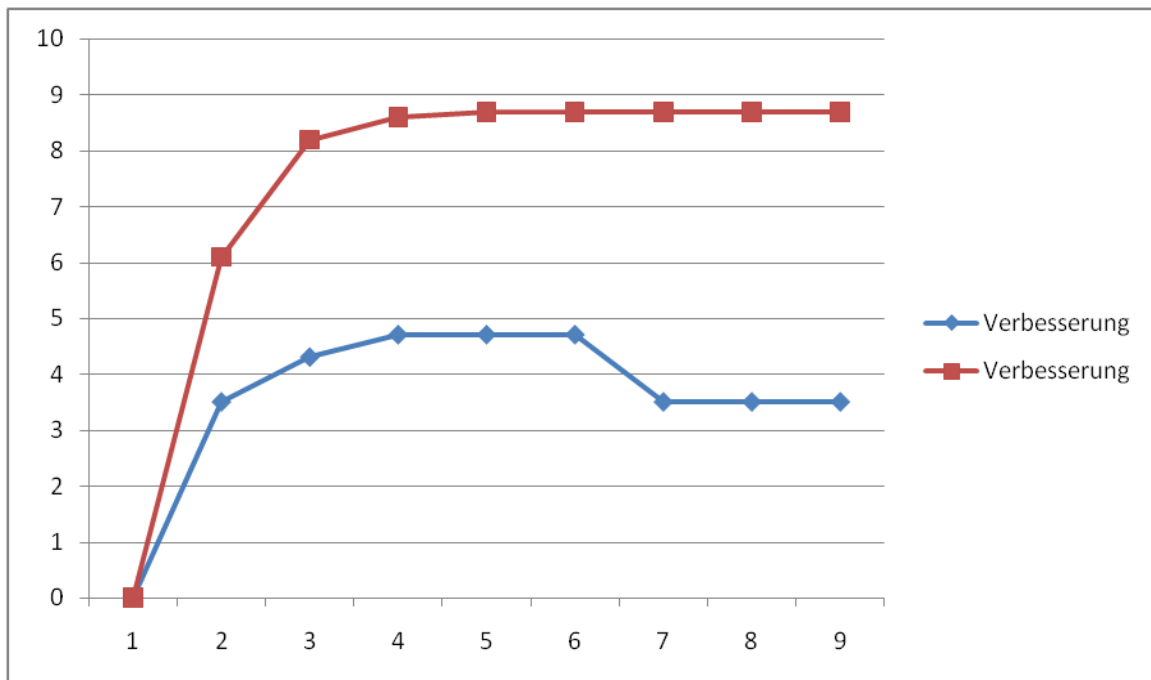


Abbildung 20: Diagramm Veränderung Anzahl erfolgreicher Auftragsabwicklungen

Die farblichen Markierungen in diesem Diagramm entsprechen denen des obigen. Hier wird deutlich, dass eine zu hohe Anzahl an Durchläufen keine Auswirkungen hat oder zu negativen Auswirkungen führen kann. Als optimaler Wert für dieses Transportnetz haben sich drei bis vier Durchläufe herausgestellt.

6.5 Abläufe im Routingalgorithmus

Der Routingalgorithmus ist der umfangreichste und komplexeste Algorithmus dieses Programms. Er wurde mit Java implementiert und umfasst etwa 2000 Quelltextzeilen. Je nach Anzahl an Aufträgen, Komplexität des Transportnetzes sowie den gewählten Programmeinstellungen und der Rechenleistung des Systems kann der Routingprozess bis zu mehrere Minuten in Anspruch nehmen. Der Routingalgorithmus kann dabei in mehrere Teilschritte unterteilt werden, welche nachfolgend aufgeführt und kurz erläutert sind. Da der Quelltext umfangreich ist, wird auf die komplette Darstellung des Routingalgorithmus verzichtet. Bei Unklarheiten oder für weiterführende Informationen kann der kommentierte Quelltext des Algorithmus zu Rate gezogen werden.

Teilschritte des Routings:

0. Initialisieren des Routings und der Variablen
1. Anbindung der sekundären Knoten an die primären
2. Sortieren der Aufträge

3. Schleife mit Routing der einzelnen Aufträge
 - 3.1 Variablen initialisieren und vorbereiten
 - 3.2 Abwicklung über ein Depot prüfen
 - 3.3 A* Algorithmus
 - 3.3.I A* terminiert
 - 3.3.I.a Route rekonstruieren
 - 3.3.I.b Zeitfenster überprüfen
 - 3.3.I.b.a terminierte Kanten
 - 3.3.I.b.β unterminierte Kanten
 - 3.3.I.c Restriktionen eingehalten, Route möglich
 - 3.3.II A* Expand Node
 - 3.4 Keine Lösung gefunden?
 - 3.4.I Kombination festlegen
 - 3.4.II Abwicklung über ein Depot prüfen
4. Auswertung und Ausgabe
 - 4.1 Kantendaten schreiben
 - 4.2 Auswertung Routingergebnisse
5. GUI aktualisieren

Im Teilschritt 0 werden das GUI und die Kostenvariablen gesetzt. Außerdem werden die Objektdaten initialisiert. Anschließend werden im 1. Schritt des Routings die Kundenstandorte an die drei jeweils am nächsten liegenden Depots angeschlossen. Dazu wird eine Klasse namens Knotenanbindung aufgerufen, die die Entfernungen für jeden Kundenstandort bestimmt und diese zusammen mit den drei am nächsten liegenden Depots zurückliefert. Im 2. Schritt werden die Aufträge je nach ausgewählter Option sortiert. Dies erfolgt durch den implementierten Quicksort-Algorithmus.

Im dritten Schritt wird nun über die Aufträge iteriert. In dieser umfassenden Schleife werden zuerst in Schritt 3.1 die für die Berechnung der Auftragsabwicklung benötigten Variablen initialisiert sowie die Vor- und Nachlaufkosten für die jeweils am nächsten liegenden Depots berechnet. Es kann vorkommen, dass ein Auftrag über ein Depot abgewickelt werden kann, falls Start- und Zieldepot identisch sind. Dies wird in Schritt 3.2 überprüft. Falls dies der Fall ist, ist die Berechnung für diesen Auftrag abgeschlossen und die Berechnung des nächsten Auftrags kann erfolgen. Die Abläufe zur Berechnung einer Auftragsabwicklung sind zur besseren Übersicht unten stehend in einer Abbildung skizziert. Größtenteils können Aufträge nicht über das gleiche Depot abgewickelt werden und somit wird in Schritt 3.3 der A* Algorithmus zum Bestimmen einer Route gestartet. Dieser Algorithmus wurde für die hier vorliegende Aufgabenstellung angepasst und kann mehrmals mit einem veränderten Startvektor für denselben Auftrag wiederholt werden. Letztendlich wird die Auftragsberechnung erst durch eine Boolean-Variable abgebrochen, sobald der Auftrag erfolgreich berechnet wurde oder die maximale Anzahl an Iterationen der Heuristik erreicht wurde und der Auftrag somit nicht erfolgreich gelöst werden konnte. Beim Start des A* Algorithmus werden die benötigten Variablen initialisiert. So werden Listen („openlist“ und „closedlist“) für den A* Algorithmus erstellt und die Kosten für Vor- und Nachlauf berechnet. Es schließt sich eine weitere Schleife an, in der solange das Transportnetz nach dem kürzesten Weg durchsucht wird, bis dieser gefunden wurde oder kein Weg nach der Überprüfung aller Knoten gefunden werden konnte. Innerhalb dieser Schleife wird zuerst in Schritt 3.3.I überprüft, ob die Lösung bereits gefunden wurde und der A* Algorithmus somit bereits terminiert. Dies ist der Fall, wenn der aktuell betrachtete

Knoten dem Zielknoten entspricht. Falls dieser Fall vorliegt, werden in Schritt 3.3.I.a die zurückgelegte Route über zwei Arrays von Zeigern rekonstruiert und anschließend in Schritt 3.3.I.b die Zeitfenster überprüft und angepasst. Dieser sehr umfangreiche Schritt wird unterteilt in terminierte (3.3.I.b.a) und unterminierte Kanten (3.3.I.b.β). Prinzipiell werden die Zeitfenster der Kanten schon beim Durchsuchen des Transportnetzes überprüft, jedoch werden sie erst in diesem Schritt angepasst. Dabei kann es vorkommen, dass die Anpassung einer Kante nicht möglich ist, weil sonst andere Aufträge negativ beeinflusst werden könnten. Ist die Auftragsabwicklung mit den vorliegenden Kanten und Zeitfenstern nicht möglich, wird die ausschlaggebende Kante für einen möglichen weiteren Durchlauf gesperrt. Dadurch werden die Anfangsbedingungen bzw. der Startvektor geändert und der A* Algorithmus kann erneut gestartet werden. In Schritt 3.3.I.c wird überprüft, ob die Routenberechnung erfolgreich war oder es zu Kollisionen zwischen den Zeitfenstern der Kanten gekommen ist. Bei erfolgreicher Berechnung werden nun die Kosten für den Auftrag berechnet, die Routingergebnisse im Auftrag gespeichert und die Kapazitäten der Kanten aktualisiert. Anschließend wird mit der Berechnung des nächsten Auftrags fortgefahren.

Wenn der aktuell betrachtete Knoten (noch) nicht der Zielknoten ist, wird mit dem systematischen Durchsuchen des Transportnetzes in Schritt 3.3.II fortgefahren. In diesem Schritt werden ausgehend vom aktuell betrachteten Knoten nach Kanten gesucht, die sinnvoll zum Zielknoten hinführen. Dabei wird durch die ständige Berechnung der Kosten der kostenminimale Weg ermittelt. Es wird solange im Transportnetz gesucht, bis der Zielknoten erreicht wurde, oder keine weiteren noch nicht besuchten Knoten vorliegen. Genau dieser Zustand wird in Schritt 3.4 überprüft. Falls dieser Zustand eingetreten ist, kann je nach vorgenommenen Einstellungen für Heuristik und Anbindung der Knoten ein weiterer Durchlauf mit geänderten Anfangsbedingungen gestartet werden oder die Suche wird endgültig erfolglos beendet. Um die Anfangsbedingungen zu ändern wird in Schritt 3.4.I entweder die Kombination aus Start- und Zieldepot geändert oder es werden Kanten aus der Berechnung herausgenommen, die eine erfolgreiche Abwicklung des Auftrags verhindern. Nach der Veränderung des Startvektors wird erneut in Schritt 3.4.II überprüft, ob der Auftrag über dasselbe Start- und Zieldepot überprüft werden kann. Falls dies der Fall ist, wird die Auftragsberechnung erfolgreich abgeschlossen und der nächste Auftrag kann berechnet werden. Andernfalls wird erneut mit der Berechnung des A* Algorithmus mit den geänderten Anfangsbedingungen begonnen (Rücksprung zu Schritt 3.3). Wurden jedoch alle Kombinationen aus Start- und Zieldepot geprüft und ist die maximale Anzahl an Durchläufen der Heuristik für die letzte Kombination von Depots erreicht, konnte der Auftrag endgültig nicht erfolgreich berechnet werden. Es folgt die Berechnung des nächsten Auftrags.

Nachdem über alle Aufträge iteriert wurde, wird in Schritt 4 die Auswertung gestartet. Dabei werden in Schritt 4.1 die Kantendaten berechnet und in aktualisierter Form ausgegeben sowie in Schritt 4.2 eine Zusammenfassung der Routingergebnisse zusammengestellt und ausgegeben. Abschließend wird in Schritt 5 das GUI aktualisiert.

Nachfolgend werden die Abläufe bei der Berechnung eines Auftrags vereinfacht in einer Abbildung dargestellt. Dabei wurden bewusst alle dazu nicht relevanten Teile des Routingprozesses ausgespart, um nur die Vorgänge bei der Auftragsberechnung zu verdeutlichen.

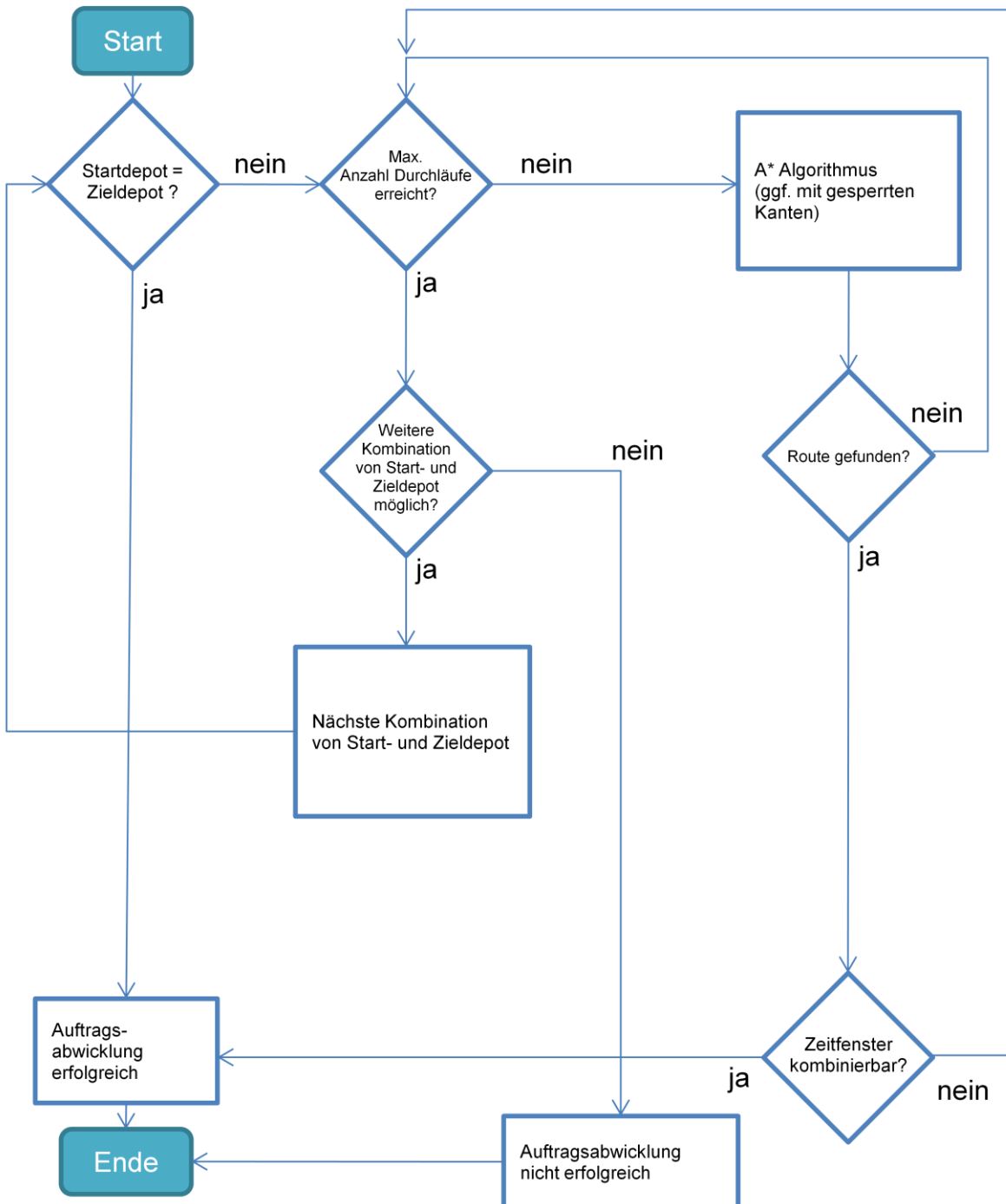


Abbildung 21: Skizzierung der Abläufe bei einer Auftragsberechnung

Um das Verständnis über die Abläufe im Routingprozess zu verbessern, werden diese hier anhand der oben stehenden Grafik nochmals erläutert. Zuerst wird überprüft, ob die Abwicklung des Auftrags über ein Depot möglich ist. Dies ist der Fall, wenn das Startdepot dem Zieldepot entspricht. Ist die Abwicklung über ein Depot nicht möglich, dann wird der A* Algorithmus zur Routensuche gestartet. Falls eine Route gefunden wurde und die Zeitfenster der Kanten ohne Kollision kombinierbar sind, kann die Auftragsberechnung erfolgreich beendet werden. Kann keine Route unter den gegebenen Nebenbedingungen gefunden werden oder können die Zeitfenster nicht angepasst werden, so kann ein

weiterer Durchlauf des A* Algorithmus gestartet werden. Dies erfolgt, wenn die maximale Anzahl an Durchläufen der Heuristik noch nicht erreicht wurde. Wurde die maximale Anzahl jedoch erreicht, so kann eine andere Kombination aus Start- und Zieldepot festgelegt werden. Nun wird erneut überprüft, ob die Abwicklung des Auftrags über ein einziges Depot möglich ist. Falls nicht, wird der A* Algorithmus erneut gestartet. Erst wenn keine weiteren Kombinationen aus Start- und Zieldepot mehr möglich sind (maximal 9 Kombinationen) und die maximale Anzahl an Durchläufen der Heuristik für die letzte Kombination von Depots erreicht wurde, wird die Auftragsberechnung erfolglos beendet.

6.6 Speichern der Routingergebnisse

Nachdem der Routingprozess erfolgreich beendet wurde, können die Routingergebnisse in Form von einer XML-Datei oder einer HTML-Datei ausgegeben werden¹⁴. Generell eignen sich XML-Dateien sehr gut zur automatisierten Verarbeitung bspw. durch ein anderes Programm. Für den Menschen ist eine im Webbrowser dargestellte HTML-Datei besser lesbar als eine XML-Datei. Weitere Informationen über den Aufbau der hier eingesetzten XML-Dateien können der umfangreichen Dokumentation entnommen werden, welche auf dem Internetportal erhältlich ist. In diesem Kapitel soll der Fokus auf der nutzerfreundlichen HTML-Datei liegen.

Nachfolgend wird der Aufbau einer vom Routingprogramm erzeugten HTML-Datei erläutert. Wie auf folgendem Screenshot sichtbar ist, fängt jede dieser Dateien durch eine Zusammenfassung über die Routingergebnisse und die zugrunde liegenden Einstellungen an:

¹⁴ Die Syntax und die Eigenschaften vom XML und HTML können bspw. Kemper (2006) entnommen werden.

JRouter

Durchschnittsgeschwindigkeit im Hauptlauf	65.0km/h
Durchschnittsgeschwindigkeit im Vor-/Nachlauf	40.0km/h
Kosten pro km und Kapazitätseinheit Hauptlauf	0.05€
Kosten pauschal Hauptlauf	100.0€
Kosten pro km und Kapazitätseinheit Vor-/Nachlauf (variabel)	1.5€
Kosten Vor-/Nachlauf pauschal (fix)	5.0€
Kosten Umschlag pro Kapazitätseinheit (variabel)	1.0€
Kosten Umschlag pauschal (fix)	4.0€
Umschlagdauer	5.0min
Umrechnungsfaktor Luftlinie	1.3
Untermionierte Kanten	false
Auftragssortierung	Sortierung nach Kapazität 1 (aufsteig
Andere Depots anbinden	true
Heuristik aktiviert	true
Anzahl Durchläufe Heuristik	3

Gefahren Kanten	78/152	51.31%
Erfolgreiche Auftragsabwicklung	835/1000	83.5%
über Nahverkehr	142/835	17.0%
über eine Kante	407/835	48.74%
über zwei Kanten	185/835	22.15%
über drei oder mehr Kanten	101/835	12.09%

Abbildung 22: Screenshot HTML-Datei – Zusammenfassung

Anschließend wird für jeden Auftrag die Auftragsabwicklung Schritt für Schritt detailliert aufgeführt. Dabei wird auch ein Hyperlink zur Visualisierung der Auftragsabwicklung eingefügt. In der unten stehenden Abbildung ist neben der schrittweisen Darstellung einer Auftragsabwicklung auch die Visualisierung dieser Auftragsabwicklung rechts daneben dargestellt, welche durch den Hyperlink geöffnet werden kann:

6: Visualisierung Auftragsabwicklung

Von Kunde 10060 (Kunde 60) zu Kunde 10050 (Kunde 50)

Benötigte Kapazität 1 0.02

Benötigte Kapazität 2 10.0

Früheste Startzeit 2008-01-01 06:00:00.0

Späteste Zielzeit 2008-01-02 20:00:00.0

Abwicklung über mehrere Depots:

0 Vorlauf

Von Firma 10060 Kunde 60

Nach Depot 3 PSLT

Kosten 6.71€

Entfernung 57km

geschätzte Fahrtzeit 111.15min

Abfahrt 2008-01-01 12:17:27.68

Ankunft 2008-01-01 14:08:52.352

Umschlagdauer 5.0min

1

Von Depot 3 PSLT

Nach Depot 9 Killepitsch

Über Kante 9

Kosten 0.34€

Entfernung 273.95km

geschätzte Fahrtzeit 252.88min

Abfahrt 2008-01-01 14:16:16.144

Ankunft 2008-01-01 18:28:49.92

Umschlagdauer 5.0min

Wartedauer 32min

2

Von Depot 9 Killepitsch

Nach Depot 34 Karo

Über Kante 131

Kosten 0.34€

Entfernung 279.75km

geschätzte Fahrtzeit 258.23min

Abfahrt 2008-01-01 19:01:11.0

Ankunft 2008-01-01 23:19:22.496

Umschlagdauer 5.0min

3 Nachlauf

Von Depot 34 Karo

Nach Firma 10050 Kunde 50

Kosten 6.23€

Entfernung 41km

geschätzte Fahrtzeit 79.95min

Abfahrt 2008-01-01 23:25:55.712

Ankunft 2008-01-02 00:46:45.376

Kosten:

Vorlauf: 6.71€ (57km)

Kanten: 0.68€

Umschläge: 8.04€

Nachlauf: 6.23€ (41km)

Gesamtkosten: 21.66€

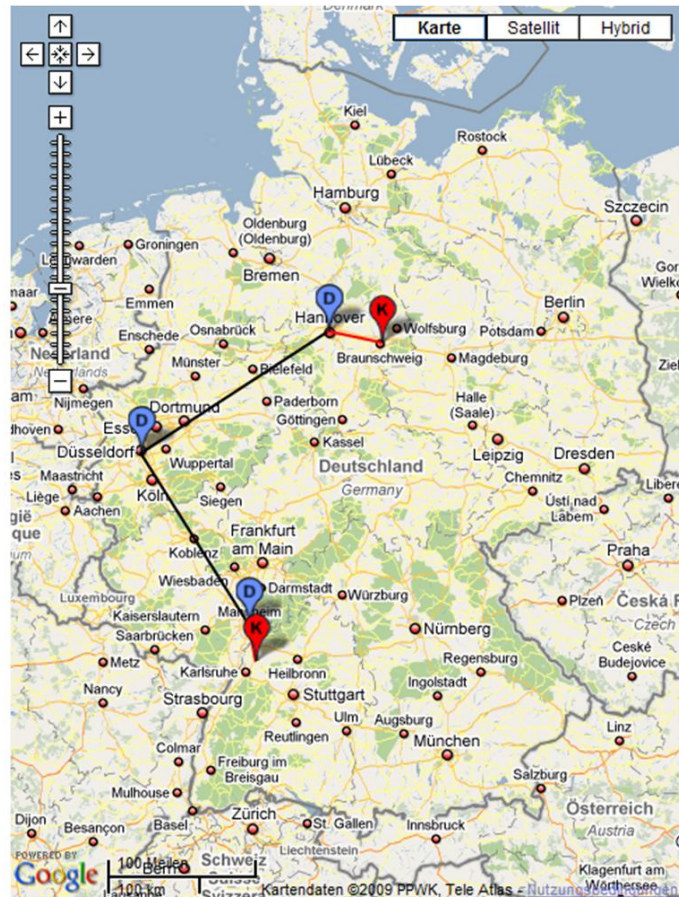


Abbildung 23: Screenshot HTML-Datei – Auftragsabwicklung

Bei der Berechnung eines Auftrags kann es vorkommen, dass ein Auftrag nicht abgewickelt werden kann oder über ein einziges Depot abgewickelt wird. In beiden Fällen ist dies bei der Ausgabe des jeweiligen Auftrags in der HTML-Datei vermerkt. Im Anschluss an die Auflistung aller Auftragsabwicklungen werden in der HTML-Datei die Daten für alle Kanten Kante für Kante ausgegeben. Auch hierbei werden Links zur Visualisierung der Kanten in die HTML-Datei geschrieben.

150: [Visualisierung Kante](#)

Von Depot 29 (Lessing) nach Depot 23 (Zentrum)

Auslastung Kapazität 1: 16.74 / 33.0
 Auslastung Kapazität 2: 7692.0 / 26000.0
 wird gefahren: true
 Entfernung: 477.92km
 Geschätzte Fahrtzeit 441 min (7h 21min)
 Früheste Abfahrt: 2008-01-01 09:05:13.344
 Tatsächliche Abfahrt: 2008-01-01 15:29:40.172
 Späteste Abfahrt: 2008-01-01 21:54:07.0
 Früheste Ankunft: 2008-01-01 16:26:29.888
 Tatsächliche Ankunft: 2008-01-01 22:50:58.56
 Späteste Ankunft: 2008-01-02 05:15:27.232
 Kosten der Kante: 888.57€

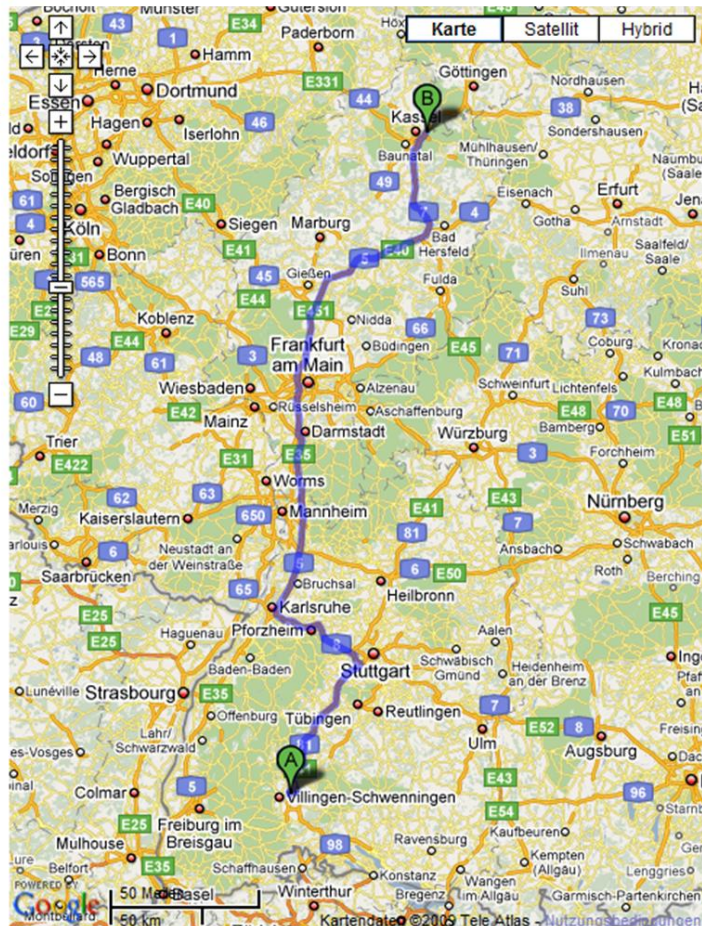


Abbildung 24: Screenshot HTML-Datei – Kanten

Abgeschlossen wird die Darstellung der HTML-Datei durch eine Fußzeile mit dem Erstellungsdatum der jeweiligen Datei und den Kontaktdaten der Mitarbeiter der Forschungsstellen.

Eine dieser dynamisch generierten HTML-Dateien kann je nach Umfang des Transportnetzes schnell mehrere Megabytes groß werden. Dadurch kann es beim Laden einer solchen Datei im Webbrowser zu einer kurzen Wartezeit kommen. Generell kann der Microsoft Internet Explorer in der aktuellen Version gut mit diesen Dateien umgehen.

7 Fazit und Ausblick

Im Rahmen dieser Studie wurde ein Transportmodell erstellt, mit dem durch automatische Generierung von Transportketten die Disposition von Strecken- und Flächenverkehren in komplexen Transportnetzen optimiert wird. Dabei werden unter Berücksichtigung der Nebenbedingungen und Restriktionen die kostenminimalen Routen zur Auftragsabwicklung bestimmt. Innerhalb von wenigen Sekunden können so für einen vollständigen Auftragspool die Routen über die vorliegenden Knoten und Kanten in einem Ausmaß berechnet werden, wie es durch manuelle Planung nicht möglich wäre. Verschiedene Einstellungsmöglichkeiten und Parameter können zur Anpassung des Modells an das reale Transportgeschäft genutzt werden. Die Visualisierung von Knoten, Kanten und Aufträgen erhöht die Übersichtlichkeit und dient außerdem zur graphischen Überprüfung der Routingergebnisse. Die hier erarbeiteten und im Internet veröffentlichten Forschungsergebnisse sollen Logistik- und Transportunternehmen einen Anstoß geben, ihre bisherigen Vorgehensweisen bei der Disposition zu überdenken. Das Importprogramm JImport ermöglicht den einfachen Import des Transportnetzes und der Transportaufträge eines Unternehmens. Diese Daten können dann von dem Routingprogramm JRouter aufgenommen werden und durch heuristische Verfahren optimiert werden. Somit sind Unternehmen in der Lage, ohne großen Aufwand und Kosten ihre Routenplanung mit den Ergebnissen dieser Studie zu verifizieren. Folgende kritische Fragen könnten dem Disponenten helfen, Kostensenkungspotentiale aufzudecken:

- Wie werden im Unternehmen Transportaufträge geplant?
- Wird bei der zur Routenplanung eingesetzten Systematik die Kostenminimierung als Kernkonzept angesehen?
- Ist die Disposition von Transportaufträgen personenunabhängig?
- Steht bei der Disposition von Aufträgen genug Zeit zur Verfügung, um den Auftragspool ganzheitlich zu analysieren und kostenoptimal zu planen?
- Werden bestehende Routen regelmäßig auf ihre Effizienz überprüft?
- Wird geprüft, wie und warum Routen zustande gekommen sind?
- Werden Routen nur so gefahren, weil sie immer so gefahren wurden?

Durch die automatisierte Berechnung und Optimierung des Auftragspools in dem Routingprogramm kann eine von manueller Disposition losgelöste Referenzlösung erstellt werden. Sie stellt unabhängig und neutral das berechnete Optimum dar. Dieses kann dann mit der realen Disposition verglichen werden und Abweichungen bilateral kritisch hinterfragt werden.

Die Forschungsergebnisse wurden in Form einer Internetplattform der Öffentlichkeit frei zur Verfügung gestellt. Der Open-Source Gedanke und die Plattformunabhängigkeit dieser Studie soll zur Weiterentwicklung des erstellten Modells motivieren. Gerade KMUs können so auf dieser Plattform aufbauen und sie an ihre Anforderungen anpassen und erweitern. Weitere Möglichkeiten ergeben sich durch die universelle XML-Schnittstelle, durch die auch Optimierungssysteme und -programme wie bspw. ILOG CPLEX angesteuert werden können. Als Ausblick für weitere Entwicklungen ist die Implementierung von verschiedenen Verkehrsarten (Linienverkehr, Begegnungsverkehr, Rundläufe, etc.), das Erstellen von Restriktionen für Vor- und Nachlauf sowie das Begrenzen von Depotkapazitäten zu nennen. Auch das Erzeugen eines optimierten Startvektors für unterminierte Kanten durch eine Heuristik scheint erfolgsversprechend. Letztendlich wird

durch unterminierte Kanten ein Schritt von der operativen Planung hin zur taktischen gegangen. Die vorliegende Studie unterstützt die operative Tourenplanung durch die Optimierung der Auftragsabwicklung in einem größtenteils gegebenen Transportnetz. Strategisch betrachtet ist die Optimierung des Transportnetzes (Depotstandorte, Routenführung, Verkehrsart, Zeitfenster, etc.) unter Betrachtung einer großen Menge von Aufträgen hochrelevant und eröffnet großes Kostensenkungspotential.

Literaturverzeichnis

1. Balthes-Götz, B. (2003). Einführung in das Programmieren mit Java. Trier: Universitäts-Rechenzentrum Trier.
2. Domschke, W. (1995). Logistik: Transport. München: Oldenbourg.
3. Domschke, W., & Drexl, A. (2007). Einführung in Operations Research. Berlin: Springer-Verlag.
4. Elchner, J. (1989). Die Tourenkonstruktion im heterogenen Fuhrpark. Regensburg: CH-Verlag & Druckerei.
5. Elmasri, R., & Navathe, S. (2002). Grundlagen von Datenbanksystemen. München: Pearson Education Deutschland GmbH.
6. Google Maps-API-Konzepte. (2009). Abgerufen am 6.11.2009 von Google Maps: <http://code.google.com/intl/de/apis/maps/documentation/index.html>
7. Janz, O. (2002): Integriertes Transportnetzmanagement. Dissertation Universität Mannheim.
8. Jenkins, R. (30.11.2008). A-Star Algorithm in Java. Abgerufen am 6.11.2009 von Memoization: <http://memoization.com/2008/11/30/a-star-algorithm-in-java/>
9. Kemper, A., & Eickler, A. (2006). Datenbanksysteme. München: Oldenbourg Wissenschaftsverlag GmbH.
10. Lenz, B. (25.02.2009). landkarte-deutschland01-klein.jpg. Abgerufen am 9.03.2009 von Landkarte Direkt: <http://www.landkarte-direkt.de>
11. Lafore, R. (2003). Data Structures & Algorithms in Java. Indianapolis: Sams Publishing.
12. Lester, P., & Weber-Groß, W. (6.03.2006). A* Pfadfindung für Anfänger. Abgerufen am 6.11.2009 von http://www.policyalmanac.org/games/aStarTutorial_de.html
13. Lusti, M. (2003). Dateien und Datenbanken. Berlin: Springer-Verlag.
14. Möller, R., & Hotz, L. (1991). Suchalgorithmen und Interaktionstechniken für Fahrplan Informationssysteme. Labor für Künstliche Intelligenz , Bericht LKI-M-91/3.
15. Nazmul. (21.04.2008). Geocoding tutorial – Accessing Google Static Maps from Java. Abgerufen am 27.11.2009 von Developer Life: <http://developerlife.com/tutorials/?p=274>
16. Ottmann, T., & Widmayer, P. (1996). Algorithmen und Datenstrukturen. Heidelberg: Spektrum Akademischer Verlag GmbH.
17. Otto, A., & Gräbel, P. (19.04.2004). Visualisierte Suchalgorithmen . Abgerufen am 6.11.2009 von Suchalgorithmen in der Informatik: <http://kik.informatik.fh-dortmund.de/visual/>
18. Pankratz, G., & Gehring, H. (2000). Ein genetischer Algorithmus für ein dynamisches expeditionelles Dispositionsproblem. Operations Research Proceedings , 420-425.
19. Patel, A. (1.06.2009). Amit's A* Pages. Abgerufen am 6.11.2009 von Amit's Game Programming Site: <http://theory.stanford.edu/~amitp/GameProgramming/>
20. Suhl, L., & Mellouli, T. (2006). Optimierungssysteme. Berlin: Springer-Verlag.
21. Ullenboom, C. (2006). JAVA ist auch eine Insel. Bonn: Galileo Press.
22. Vonhoegen, H. (2007). Einstieg in XML. Bonn: Galileo Press.
23. Vossen, G. (2008). Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme. München: Oldenbourg Wissenschaftsverlag GmbH.
24. Wolmeringer, G. (2007). Java 6 lernen mit Eclipse. Bonn: Galileo Press.

IWI Discussion Paper Series/Diskussionsbeiträge

ISSN 1612-3646

Michael H. Breitner, *Rufus Philip Isaacs and the Early Years of Differential Games*, 36 p., #1, January 22, 2003.

Gabriela Hoppe and Michael H. Breitner, *Classification and Sustainability Analysis of e-Learning Applications*, 26 p., #2, February 13, 2003.

Tobias Brüggemann und Michael H. Breitner, *Preisvergleichsdienste: Alternative Konzepte und Geschäftsmodelle*, 22 S., #3, 14. Februar, 2003.

Patrick Bartels and Michael H. Breitner, *Automatic Extraction of Derivative Prices from Webpages using a Software Agent*, 32 p., #4, May 20, 2003.

Michael H. Breitner and Oliver Kubertin, *WARRANT-PRO-2: A GUI-Software for Easy Evaluation, Design and Visualization of European Double-Barrier Options*, 35 p., #5, September 12, 2003.

Dorothee Bott, Gabriela Hoppe und Michael H. Breitner, *Nutzenanalyse im Rahmen der Evaluation von E-Learning Szenarien*, 14 S., #6, 21. Oktober, 2003.

Gabriela Hoppe and Michael H. Breitner, *Sustainable Business Models for E-Learning*, 20 p., #7, January 5, 2004.

Heiko Genath, Tobias Brüggemann und Michael H. Breitner, *Preisvergleichsdienste im internationalen Vergleich*, 40 S., #8, 21. Juni, 2004.

Dennis Bode und Michael H. Breitner, *Neues digitales BOS-Netz für Deutschland: Analyse der Probleme und mögliche Betriebskonzepte*, 21 S., #9, 5. Juli, 2004.

Caroline Neufert und Michael H. Breitner, *Mit Zertifizierungen in eine sicherere Informationsgesellschaft*, 19 S., #10, 5. Juli, 2004.

Marcel Heese, Günter Wohlers and Michael H. Breitner, *Privacy Protection against RFID Spying: Challenges and Countermeasures*, 22 p., #11, July 5, 2004.

Liina Stotz, Gabriela Hoppe und Michael H. Breitner, *Interaktives Mobile(M)-Learning auf kleinen End-geräten wie PDAs und Smartphones*, 31 S., #12, 18. August, 2004.

Frank Köller und Michael H. Breitner, *Optimierung von Warteschlangensystemen in Call Centern auf Basis von Kennzahlenapproximationen*, 24 S., #13, 10. Januar, 2005.

Phillip Maske, Patrick Bartels and Michael H. Breitner, *Interactive M(obile)-Learning with UbiLearn 0.2*, 21 p., #14, April 20, 2005.

Robert Pomes and Michael H. Breitner, *Strategic Management of Information Security in State-run Organizations*, 18 p., #15, May 5, 2005.

Simon König, Frank Köller and Michael H. Breitner, *FAUN 1.1 User Manual*, 134 p., #16, August 4, 2005.

Christian von Spreckelsen, Patrick Bartels und Michael H. Breitner, *Geschäftsprozessorientierte Analyse und Bewertung der Potentiale des Nomadic Computing*, 38 S., #17, 14. Dezember, 2006.

Stefan Hoyer, Robert Pomes, Günter Wohlers und Michael H. Breitner, *Kritische Erfolgsfaktoren für ein Computer Emergency Response Team (CERT) am Beispiel CERT-Niedersachsen*, 56 S., #18, 14. Dezember, 2006.

Christian Zietz, Karsten Sohns und Michael H. Breitner, *Konvergenz von Lern-, Wissens- und Personalmanagementssystemen: Anforderungen an Instrumente für integrierte Systeme*, 15 S., #19, 14. Dezember, 2006.

Christian Zietz und Michael H. Breitner, *Expertenbefragung „Portalbasiertes Wissensmanagement“: Ausgewählte Ergebnisse*, 30 S., #20, 5. Februar, 2008.

IWI Discussion Paper Series/Diskussionsbeiträge

ISSN 1612-3646

Harald Schömburg und Michael H. Breitner, *Elektronische Rechnungsstellung: Prozesse, Einsparpotentiale und kritische Erfolgsfaktoren*, 36 S., #21, 5. Februar, 2008.

Halyna Zakhariya, Frank Köller und Michael H. Breitner, *Personaleinsatzplanung im Echtzeitbetrieb in Call Centern mit Künstlichen Neuronalen Netzen*, 35 S., #22, 5. Februar, 2008.

Jörg Uffen, Robert Pomes, Claudia M. König und Michael H. Breitner, *Entwicklung von Security Awareness Konzepten unter Berücksichtigung ausgewählter Menschenbilder*, 14 S., #23, 5. Mai, 2008.

Johanna Mählmann, Michael H. Breitner und Klaus-Werner Hartmann, *Konzept eines Centers der Informationslogistik im Kontext der Industrialisierung von Finanzdienstleistungen*, 19 S., #24, 5. Mai, 2008.

Jon Sprenger, Christian Zietz und Michael H. Breitner, *Kritische Erfolgsfaktoren für die Einführung und Nutzung von Portalen zum Wissensmanagement*, 44 S., #25, 20. August, 2008.

Finn Breuer und Michael H. Breitner, *„Aufzeichnung und Podcasting akademischer Veranstaltungen in der Region D-A-CH“: Ausgewählte Ergebnisse und Benchmark einer Expertenbefragung*, 30 S., #26, 21. August, 2008.

Harald Schömburg, Gerrit Hoppen und Michael H. Breitner, *Expertenbefragung zur Rechnungseingangsbearbeitung: Status quo und Akzeptanz der elektronischen Rechnung*, 40 S., #27, 15. Oktober, 2008.

Hans-Jörg von Mettenheim, Matthias Paul und Michael H. Breitner, *Akzeptanz von Sicherheitsmaßnahmen: Modellierung, Numerische Simulation und Optimierung*, 30 S., #28, 16. Oktober, 2008.

Markus Neumann, Bernd Hohler und Michael H. Breitner, *Bestimmung der IT-Effektivität und IT-Effizienz service-orientierten IT-Managements*, 20 S., #29, 30. November, 2008.

Matthias Kehlenbeck und Michael H. Breitner, *Strukturierte Literaturrecherche und -klassifizierung zu den Forschungsgebieten Business Intelligence und Data Warehousing*, 10 S., #30, 19. Dezember, 2009.

Michael H. Breitner, Matthias Kehlenbeck, Marc Klages, Harald Schömburg, Jon Sprenger, Jos Töller und Halyna Zakhariya, *Aspekte der Wirtschaftsinformatikforschung 2008*, 128 S., #31, 12. Februar, 2009.

Sebastian Schmidt, Hans-Jörg v. Mettenheim und Michael H. Breitner, *Entwicklung des Hannoveraner Referenzmodells für Sicherheit und Evaluation an Fallbeispielen*, 30 S., #32, 18. Februar, 2009.

Sissi Eklun-Natey, Karsten Sohns und Michael H. Breitner, *Buildung-up Human Capital in Senegal - E-Learning for School drop-outs, Possibilities of Lifelong Learning Vision*, 39 S., #33, July 1, 2009.

Horst-Oliver Hofmann, Hans-Jörg von Mettenheim und Michael H. Breitner, *Prognose und Handel von Derivaten auf Strom mit Künstlichen Neuronalen Netzen*, 34 S., #34, 11. September, 2009.

Christoph Polus, Hans-Jörg von Mettenheim und Michael H. Breitner, *Prognose und Handel von Öl-Future-Spreads durch Multi-Layer-Perceptrons und High-Order-Neuronalnetze mit Faun 1.1*, 55 S., #35, 18. September, 2009.

Jörg Uffen und Michael H. Breitner, *Stärkung des IT-Sicherheitsbewusstseins unter Berücksichtigung psychologischer und pädagogischer Merkmale*, 37 S., #36, 24. Oktober, 2009.

Christian Fischer und Michael H. Breitner, *MaschinenMenschen – reine Science Fiction oder bald Realität?*, 36 S., #37, 13. Dezember, 2009.

