Universität Bremen
Angewandte Informatik — FB 3
Institut für Informatik und Verkehr (IIuV)

Lutz Zegartowski

Definition, Teilimplementation und Verifikation eines vollautomatischen Vermittlungssystems für den Personentransport

Dissertation zur Erlangung des akademischen Grades *Dr.-Ing.*

—

Vorgelegt im Fachbereich Mathematik/Informatik der Universität Bremen

Diese Arbeit wurde im Stipendienprogramm der Alfried Krupp von Bohlen und Halbach-Stiftung zur Förderung von Doktoranden auf dem Gebiet der Verkehrswissenschaften gefördert.

Dissertation mit dem Thema:

Definition, Teilimplementation und Verifikation eines vollautomatischen Vermittlungssystems für den Personentransport

Gutachter: Prof. Dr. K. Haefner, Universität Bremen
Prof. Dr. G. Marte, Universität Bremen

Das Kolloquium fand am 24. April 1998 im Raum MZH 1160 der Universität Bremen statt.

Satz: LaTeX
Gedruckt auf chlorfrei gebleichtem Papier

Gefördert von der Alfried Krupp von Bohlen und Halbach-Stiftung
Institut für Informatik und Verkehr, Fachbereich 3, Universität Bremen, Germany

Inhaltsverzeichnis

Zusammenfassung

Ziel dieser Dissertation ist es, eine auf dem Konzept des Schlanken Verkehrs ([HM94]) beruhende *PersonenLogistik* zu entwerfen und in Ansätzen zu implementieren sowie diese auf Korrektheit zu überprüfen. Es soll gezeigt werden, mit welchen Algorithmen, Heuristiken und Systemstrukturen ein für diese Aufgabe geeignetes System realisiert werden kann.

Die unterstellte Ausgangssituation ist die, daß ein VerkehrsSystemManager (Kapitel 2) seine Arbeit aufgenommen hat und nun mittels der zu erstellenden PersonenLogistik Angebote und Nachfragen nach Personentransportleistung handhaben soll. Es geht dabei *nicht* um Verkehrserzeugung, Verkehrsmittelwahl o.ä., sondern um ein System, welches in der Lage ist, mit einer großen Anzahl an privaten und öffentlichen Transportanbietern und Nachfragern umzugehen. Zu Testzwecken wurde ein Generator implementiert, der unter Zuhilfenahme verschiedener Parameter eine definierte Anzahl an Personen in einer Region plazieren kann, um die angewendeten Algorithmen und Heuristiken auf ihre Leistungsfähigkeit und Robustheit zu überprüfen. In einer möglichen Realität – wenn das System eingesetzt werden soll – geben real existierende Personen ihre Fahrtwünsche dem VerkehrsSystemManager bzw. der PersonenLogistik über definierte Schnittstellen bekannt. Der Kern des CORONA-Systems unterscheidet nicht zwischen den generierten Testdaten und zukünftigen realen Daten, da CORONA eine Datenbank nutzt, die ihre Daten entweder vom Generator oder realen I/O-Komponenten erhält.

Das Ergebnis der geforderten Teilimplementation – welche mittlerweile ein eigenständig lauffähiges System darstellt – ist das CORONA-System[1]. Mittels dieses Systems ist es *auf der Basis vektorisierter Straßenkarten* erstmals möglich, für jede beliebige Stadt eine PersonenLogistik bereitzustellen, welche ihre Aufgaben gemäß der Definition des Schlanken Verkehrs wahrnimmt. CORONA ist in der Lage, individuelle Angebote und Nachfragen nach Transportleistung im Öffentlichen Individualverkehr (ÖIV), im ÖPNV und im gebrochenen Verkehr abzugleichen. Dabei verwendet das System verschiedene Präferenzen der einzelnen Personen unter der Berücksichtigung der individuellen *Start- und Reisezeit-Toleranz*.

Weiterhin wurden einige Zusatzprogramme entwickelt, die es ermöglichen, vektorisierte Straßenkarten im GDF-Format in ein für CORONA benutzbares Format zu konvertie-

[1] CORONA = Computer Organized Routematching and Navigation

ren und die Daten mit tageszeit- und straßenabhängigen Geschwindigkeiten zu versehen. Ebenfalls ist es mit einem dieser Programme möglich, Fahrpläne des ÖPNV zu erstellen und beliebig zu modifizieren. Es wurde ferner ein spezieller Generator entwickelt, mit dessen Hilfe es möglich ist, das CORONA-System zu testen. Dieser Generator stellt realitätsnahe Daten von Anbietern und Nachfragern dem CORONA-System zur Verfügung. Somit geht CORONA weit über die Anwendungsgebiete des EFA-Systems[2] oder der technischen Möglichkeiten der Mitfahrzentralen hinaus — dies wird im Laufe der vorliegenden Dissertation deutlich werden.

Die in dieser Arbeit entwickelte PersonenLogistik ist nach dem derzeitigen Stand des Wissens ein völlig neues System, welches aufgrund seiner Komplexität ebenfalls *neuartige Heuristiken* verwendet; es legt somit den operativen Grundstein für ein VerkehrsSystemManagement im Sinne des Schlanken Verkehrs.

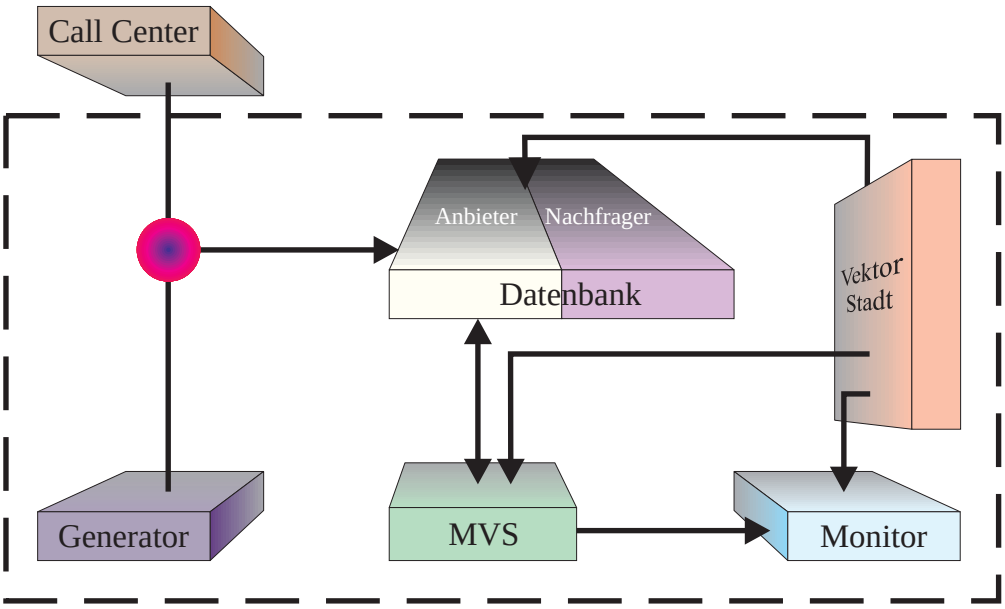


Abbildung 0.1: Das CORONA-System, bestehend aus einem Generator, einem MitfahrVermittlungsSystem (MVS), einer Datenbank und einer VektorStadt. Die VektorStadt beinhaltet eine digitale, vektorisierte Straßenkarte und ein digitales ÖPNV-Netz sowie alle relevanten Fahrpläne.

Die Abbildung 0.1 zeigt das Systemschema von CORONA. Es existiert eine *Datenbank* mit den Anbieter- und Nachfragereigenschaften, wobei zu den Anbietern die mitnahmewilligen Personen sowie der ÖPNV zählen. Die Datenbank erhält ihre Datensätze über eine Schnittstelle entweder aus einem *Call Center* oder – wie in dieser Arbeit realisiert – von dem hier implementierten *Generator*. Das *MitfahrVermittlungsSystem* (MVS) ist für den Abgleich zwischen Angebot und Nachfrage unter den geforderten Bedingungen

[2]EFA = Elektronische Fahrplan Auskunft (siehe `http://www.efa.de`).

zuständig, während der *Monitor* zur Überprüfung der Systemfunktionen und -ergebnisse dient. Wesentlicher Bestandteil der *PersonenLogistik* CORONA ist die *Vektor Stadt*, welche aus einer auf das System zugeschnittenen digitalen, vektorisierten Straßenkarte besteht.

Die im Rahmen der Dissertation getätigte Systemimplementation wird durch die gestrichelte Linie eingegrenzt und stellt somit den aktuellen Stand von CORONA dar.

1. Motivation

Wenn wir uns in den Städten und auf den Straßen umschauen, müssen wir feststellen, daß wir in den letzten Jahren ein stetiges Verkehrswachstum verzeichnen konnten. Dieses Wachstum wird sich in Zukunft weiter fortsetzen. Nun gibt es diverse Überlegungen, wie man dieses Problem lösen könnte. Grundsätzlich können zweierlei Strategien verfolgt werden:

- Höhere Ausnutzung der Infrastruktur und
- höhere Auslastung der Fahrzeuge.

Bei der erstgenannten Strategie wird versucht, durch den Bau von entlastenden Straßen sowie den Straßenrückbau in Innenstädten die Pkw um die Städte umzu-, bzw. durch die Städte hindurchzuführen. Im allgemeinen führt dieses zu induziertem Verkehr (neue Straßen) oder zu Mobilitätsverlusten (Straßenrückbau). Die Strategie der höheren Fahrzeugausnutzung hingegen versucht, das System Verkehr als Ganzes besser zu gestalten. Hierzu existieren ebenfalls diverse Ideen und Konzepte, von denen das folgende näher betrachtet werden soll.

Im Jahre 1994 wurde von K. Haefner und G. Marte das Konzept des Schlanken Verkehrs in Buchform veröffentlicht ([HM94]). Dieses Konzept sieht vor, in einer Stadt oder Region Fahrzeugkilometer zu reduzieren und Mobilität zu erhalten. Dieses Ziel kann mittels Rationalisierungsmaßnahmen im Verkehr erreicht werden, so daß die Pkw sowie der Öffentliche Personennahverkehr (ÖPNV) besser ausgelastet werden. Weiterhin sollen zusätzliche Verkehrsdienste geschaffen werden (Bürgerbusse, Anwinktaxen etc.), um ein breiteres Angebot an Transportleistung zu ermöglichen. Damit eine spürbare Reduktion von Fahrzeugkilometern erreicht werden kann, ist es unerläßlich, daß sich viele Menschen an diesem System beteiligen. Laut Definition des Schlanken Verkehrs handelt es sich hierbei um mindestens 20% aller Pkw-Besitzer in dem zu betrachtenden Gebiet. Bei dieser Anzahl an Personen (in Bremen ca. 60.000 Personen) ist es notwendig, ein vollautomatisches, computergestütztes System zu entwickeln, welches die Vielzahl von mitnahme- und mitfahrwilligen Menschen untereinander vermittelt. Herkömmliche Mitfahrzentralen sind hierbei strukturell überlastet, da diese relativ wenig Vermittlungen durchführen - und dieses halbautomatisch mit Hilfe einfacher Tabellen. Weiterhin bearbeiten diese Zentralen nur Wünsche nach sporadischen Fahrten über große Entfernungen (z.B. Bre-

men - Frankfurt). Im Schlanken Verkehr handelt es sich um tägliche Fahrten zur Arbeit, zum Einkaufen, zum Sport etc., so daß man in diesen Fällen um ein vollautomatisches System nicht mehr herumkommt. Dieses Vermittlungssystem hat die Aufgabe, Personen von A nach B zu transportieren und dabei möglichst viele Wünsche zu berücksichtigen wie z.B. Reisezeit-Toleranzen, Startzeit-Toleranzen und Gepäcktransportmöglichkeiten. In Analogie zur Güterlogistik, mittels derer ähnliche Anforderungen erfüllt werden, soll das zu erstellende System als eine *PersonenLogistik* betrachtet werden.

Ziel dieser Dissertation ist es, zu untersuchen, ob zum heutigen Zeitpunkt ein computergestütztes Vermittlungssystem (PersonenLogistik) unter verhältnismäßig geringem finanziellen Aufwand implementiert werden kann und ob dieses System in der Lage ist, eine große Anzahl von Personen und Diensten ohne nennenswerten Zeitverlust zu vermitteln. Weiterhin soll mit CORONA überprüfbar sein, ob das Konzept des Schlanken Verkehrs in einer simulierten Realität seine Ziele bezüglich der Fahrleistungsreduktion und des Mobilitätserhaltes erfüllen kann. Dabei wird von der Annahme ausgegangen, daß kein induzierter Verkehr entsteht. Nach verkehrlichen Gesichtspunkten ist es so, daß durch eine Reduktion die Personen (durch freie Straßen) längere Wege zurücklegen. In dieser Arbeit wird von konstanten Wegen ausgegangen und die Reduktion bezieht sich auf die ursprünglichen, ohne Vermittlung gefahrenen Wege.

Zur Überprüfung soll das implementierte System Testläufe unter realistischen Bedingungen durchführen und anhand der gemessenen Werte Daten zur Verfügung stellen können. Dieser Test, der als Verifikation des Computersystems gilt, ist als Micro-Simulation anzusehen, da hier keine Verkehrsflüsse in einer Stadt betrachtet werden, sondern jedes Fahrzeug einzeln.

In Kapitel 2 wird zunächst der Schlanke Verkehr näher erläutert; dann werden einige Betrachtungen diverser Systeme aufgeführt, die ähnliche Aufgaben wahrnehmen (Kapitel 3). Diese Systeme wurden bezüglich der angestellten Überlegungen und Wünsche untersucht und in Hinsicht auf die Tauglichkeit für die Problemstellung beurteilt. In Kapitel 4 werden die Anforderungen an das neu entwickelte System definiert. Es folgt die Beschreibung der vektorisierten Karte (Kapitel 5) inklusive der hier notwendigen Erweiterungen, eine Auflistung der notwendigen neuen Heuristiken und Algorithmen (Kapitel 6) sowie ein Vergleich des Routenfindungsalgorithmus mit einem kommerziellen Navigationssystem. In Kapitel 7 wird das erstellte CORONA-System beschrieben und in Kapitel 8 bezüglich Korrektheit, Leistungsfähigkeit und Robustheit getestet. Im letzten Kapitel dieser Dissertaion werden schließlich die zentralen Ergebnisse und zukünftigen Erweiterungen erörtert.

2. Der Schlanke Verkehr — eine Einführung

Der Schlanke Verkehr ist ein in sich konsistentes, verkehrspolitisches Konzept, welches im Institut für Informatik und Verkehr – Universität Bremen von K. Haefner und G. Marte im Jahre 1994 in Buchform veröffentlicht wurde ([HM94]).

Dieses Konzept beschreibt die Methodik, Fahrzeugkilometer in einer Stadt oder Region zu *reduzieren* und die Mobilität dabei weitestgehend zu erhalten. Zu den weiteren Zielen zählen die Reduzierung von Unfallzahlen durch weniger Verkehr, das Schonen der Umwelt, die Gewährleistung von Urbanität sowie eine Umsatzsteigerung der Wirtschaft, da bei weniger Verkehr auch die sonst im Stau stehenden Transportfahrzeuge kostengünstiger an ihr Ziel gelangen. Diese Punkte definieren den angestrebten Zielvektor **UUMUU**([HM94]).

Zum Erreichen der fünf Ziele werden u.a. neue Verkehrsdienste (ÖIV – Öffentlicher Individualverkehr) zusätzlich zum bestehenden ÖPNV und MIV definiert. Zu den neuen Diensten zählen unter anderem

- organisierte Mitfahrten,
- Bürgerbusse und
- Anwink-Sammeltaxen.

Wird ein Weg mittels mindestens zwei unterschiedlichen Verkehrsdiensten zurückgelegt, so wird dieses als gebrochener Verkehr bezeichnet.

Die Motivation zur Einführung dieser neuen Verkehrsdienste kommt daher, daß der MIV lediglich einen Besetzungsgrad von nahezu einer Person pro Pkw hat. Dieses hat zur Folge, daß pro Pkw drei Sitzplätze weitestgehend unbesetzt bleiben und somit freie Kapazitäten im Pkw-Verkehr existieren. Bei ca. 40 Millionen Pkw in Deutschland ergibt das eine ungenutzte Transportkapazität von 120 Millionen freien Sitzplätzen. Im ÖPNV sieht die Lage ähnlich aus: Im Mittel ist im ÖPNV jeder fünfte Sitzplatz besetzt. Im Schlanken Verkehr geht es nun darum, einige dieser freien Kapazitäten regional zu nutzen, um somit die Zahl der Personenkilometer zu erhalten oder sogar zu erhöhen und die Fahrleistung der Pkw zu reduzieren.

Das ausführende Organ des Schlanken Verkehrs ist der *VerkehrsSystemManager* (VSM). Dieser ist ein privatwirtschaftlicher Unternehmer, der mit der Stadt/Region einen VSM-Vertrag schließt, in dem die einzusparenden Fahrzeugkilometer festgehalten sind. Ferner werden dem VSM von der Stadt/Region gewisse Rechte übertragen, mit deren Hilfe er seine Aufgaben erledigen kann. Der VSM wird regelmäßig von der Stadt/Region überprüft, ob die vertraglich festgehaltene Fahrleistungsersparnis erbracht wurde. Hier ist auch die Meßgröße gegeben, mittels derer der Erfolg dieses Unternehmens ermittelt werden kann: Die Fahrleistung. Anhand dieser Größe ist es dem VSM wie der Stadt/Region möglich – ähnlich zu Kenngrößen wie die der Produktivität –, meßbare Werte zu ermitteln und entsprechend darauf zu reagieren. Zudem schließt der VSM einen Vertrag mit jedem Teilnehmer von organisierten Mitfahrten. Diese Mitglieder des „Clubs der Miteinander Fahrenden“ (CMF) verpflichten sich gegenüber dem VSM eine gewisse Transportleistung zu erbringen — also andere Personen mitzunehmen und so deren Fahrleistung einzusparen. Diese Art der Mitnahme muß zusammen mit Angeboten weiterer Transportleistungen (z.B. ÖPNV) in einer organisierten Form erfolgen. Die operationelle Basis hierfür ist eine sogenannte *PersonenLogistik*, die – ähnlich zur Güterlogistik – Angebot und Nachfrage von Routen und Personen effizient abgleicht. Ferner dient diese Personen-Logistik dazu, die Fahrleistungsreduktion zu ermitteln und so die geforderte Kenngröße bereitzustellen.

Um die Menschen zum Umdenken bzw. zur Teilnahme zu bewegen, stehen dem Verkehrs-SystemManager im Schlanken Verkehr diverse *push-* und *pull-*Maßnahmen zur Verfügung. So kann er beispielsweise den Teilnehmern im CMF eine Prämie von mehreren hundert Mark pro Jahr in Abhängigkeit ihrer erbrachten Transportleistung bzw. Fahrleistungsreduktion anbieten. Im Gegensatz zu dieser pull-Maßnahme steht dem VSM auch die Bewirtschaftung von Parkraum zu. Somit kann er durch die Einnahmen aus dieser Parkraumbewirtschaftung sein Unternehmen finanzieren und zugleich den Nicht-Teilnehmer[1] mit dieser push-Maßnahme „bestrafen“.

In dieser Dissertation wird nicht auf die push- und pull-Strategien eingegangen, da diese Maßnahmen Bestandteil des VSM-Vertrages sind und individuell zwischen VS-Manager und der jeweiligen Stadt/Region ausgehandelt werden müssen. Für Informationen bezüglich dieser Maßnahmen sei auf die Arbeiten des IIuV für das Land Brandenburg sowie für die *üstra Hannoversche Verkehrsbetriebe AG* hingewiesen ([HRHS94], [HMH96]). Hier wird schlicht von einer existierenden Population von Pkw-Besitzern ausgegangen, die entweder mitnehmen oder mitfahren wollen bzw. sich im gebrochenen Verkehr bewegen.

[1] Damit sind die Menschen gemeint, die weiterhin alleine in ihrem Pkw fahren wollen.

3. Existierende Systeme

In diesem Kapitel soll ein Blick auf existierende, vergleichbare Systeme geworfen werden. Da dieses Themengebiet relativ neu ist, gibt es in der Literatur äußerst wenig Hinweise auf diese Systeme. Es ist dennoch nach aufwendiger Literaturrecherche und intensiver Bemühungen seitens K. Haefners und mir gelungen, wissenschaftlich fundierte Informationen zu besorgen. Alle aufgeführten Systeme beschäftigen sich mit Simulationen von Verkehr auf der Micro-Ebene bzw. dynamischen Vermittlungen zwischen Anbietern und Nachfragern. Da das von mir zu spezifizierende System u.a. eine vollautomatische Vermittlung durchführen soll und die Ergebnisse überprüft werden sollen, muß selbstverständlich diese Vermittlung simuliert und analysiert werden. Dementsprechend kann von einer Ähnlichkeit dieser Systeme zu dem später vorgestellten, von mir entwickelten System gesprochen werden. An dieser Stelle sei darauf hingewiesen, daß die Anforderungsdefinition bereits um den Jahreswechsel 1995/1996 fertiggestellt war, und einige Systeme um diese Zeit in Anfängen implementiert wurden.

3.1 Real-Time Ridesharing

Unter dem Stichwort *Real-Time Ridesharing* oder auch *Dynamic Ridesharing* wird das Vermittlungsverfahren zusammengefaßt, mit welchem ich mich ebenfalls beschäftige. Es geht darum, innerhalb kürzester Zeit individuelle Vermittlungen aus einem Pool an Menschen bzw. Pkw diejenigen zusammenzubringen, die innerhalb eines bestimmten Gebietes ihre jeweiligen Start- und Zielpunkte haben. Dabei geht es lediglich um die reine Vermittlung von Personen – nicht um Konzepte wie den Schlanken Verkehr o.ä. In diesem Abschnitt sollen verschiedene Systeme vorgestellt werden, die in den Vereinigten Staaten von Amerika zum Einsatz gekommen sind. Alle diese Systeme haben jedoch *keinen gemeinsamen* Nenner, so daß es um so mehr notwendig erscheint, ein System zu spezifizieren, welches alle gewünschten Eigenschaften vereint und eine optimale Vermittlungsstrategie gewährleistet. Wie nämlich später zu sehen sein wird (Kapitel 8, Seite 91), konnten diese Systeme aus jetzt verstandenen Ursachen nicht zum vollen Erfolg gelangen. Weiterhin sind bei diesen Systemen gewisse technische Voraussetzungen nicht

erfüllt, um sinnvolle Vermittlungen durchführen zu können[1]. Alle folgenden Systeme sind der Komplettbeschreibung aller Projekte [Joh96] sowie [GG98] entnommen.

3.1.1 Bellevue, Washington

Das *Bellevue Smart Traveller (BST)* System wurde von der *Bellevue Transportation Management Association* und der *University of Washington* entwickelt. Der Zugang zum BST-System erfolgt über Telefon und – mittlerweile auch in Deutschland verfügbare – Pager. Die Teilnehmer wurden auf Grund ihrer Wohnlage, ihrer Fahrtroute und eventuellen Zwischenstop-Punkten zu Gruppen zusammengefaßt. Ergibt sich eine Vermittlung, so werden nur die entsprechenden Gruppenmitglieder bzw. deren Pager benachrichtigt. Laut Bericht haben sich 134 Personen als Mitglieder registrieren lassen, jedoch wurden nur 53 regelmäßig vermittelt; alle anderen „paßten“ in keine Gruppe hinein[2]. Im Zeitraum von November 1993 bis März 1994 wurden 496 Fahrten angeboten. Einige Fahrten wurden von den Mitgliedern selbständig als konstante Mitfahrten arrangiert, deren Zahl allerdings nicht bekannt ist.

3.1.2 Sacramento, Kalifornien

Gegen Ende des Jahres 1994 begann in der Stadt Sacramento und deren mittlerer Umgebung (bis 100 km Radius) das System *Rideshare Express* seinen Dienst aufzunehmen. Dabei wurden als Mitglieder dieses Systems 360 Personen gezählt. Die Personen konnten ihre Start- und Zielpunkte in Form von *ZIP-Codes* angeben[3]. Ferner wurden die Routen nach zeitlicher Ankunft des Fahrzeugs vermittelt. Bis zum Oktober 1995 wurden ca. 10(!) gültige Anfragen durch das System gezählt. Das Hauptproblem bei dieser äußert schlechten Annahme war die mangelhaft durchgeführte Information der Betreiber. Viele Menschen wußten nichts von diesem System, und die 360 Teilnehmer (registrierte Fahrer) fühlten sich oft zu unsicher. Als weitere Punkte für die schlechten Ergebnisse wurden die geographische Verteilung der Personen sowie die Einteilung der Start- und Zielpunkte in die angesprochenen ZIP-Codes genannt. Die anschließende Phase des Projektes beschäftigt sich mit den geographischen Informationssystemen (GIS). Diese sollen es erlauben, Personen unabhängig von ihrem ZIP-Code zu vermitteln, sondern rein auf Grund ihrer aktuellen Position.

[1] Hier sei auf die *vektorisierte Straßenkarte* hingewiesen, die von den Amerikanern bisher nicht verwendet werden konnte. Sie wird im Kapitel 5 näher erläutert.

[2] In der Analyse (Kapitel 8) wird dieses Problem nochmals angesprochen bzw. erklärt; die Topographie ist eine der beiden großen Probleme neben den Zeitfenstern. Diese Probleme werden in den anderen, unten beschriebenen Systemen wieder auftauchen.

[3] *ZIP-Codes* sind – ähnlich zu unseren fünfstelligen Postleitzahlen – fest definierte Gebietseinteilungen. Ihre Ausdehnung ist allerdings kleiner und besser strukturiert als bei uns.

3.1.3 Los Angeles, Kalifornien

Das *Smart Traveller*-Projekt in Los Angeles wurde nur für drei Monate (Juli – September 1994) ins Leben gerufen und sollte Personen vermitteln, die auf Grund des kurz zuvor stattgefundenen Erdbebens nicht ihre gewohnte Mobilität aufrecht erhalten konnten. Es waren ca. 60.000 Leute von dem Beben betroffen, die als Pendler zur Arbeit nach L.A. fahren. Per Telefon konnten sich die Personen dem System mitteilen und Fahrtangebote bzw. -wünsche abgeben. Pro Woche gab es im Mittel 30 Anrufe; wieviele Vermittlungen stattgefunden haben, ist nicht bekannt.

3.1.4 Seattle, Washington

Die *University of Washington* in Seattle hat das *Seattle Smart Traveller*-System (SST) Anfang des Jahres 1996 gestartet. Dabei hatten alle Angestellten und Studenten der Universität die Möglichkeit, mittels des *World Wide Web* am System teilzunehmen und ihre jeweiligen Fahrtrouten bzw. Fahrtwünsche anzugeben. Die Fahrdaten können dabei jederzeit variiert werden – das System reagiert unmittelbar auf entsprechende Änderungen. Täglich wiederkehrende Fahrten können komfortabel interaktiv definiert werden. Die Benachrichtigungen über Vermittlungen erfolgen über *e-mail* oder Pager. Zusätzlich erhalten die vermittelten Personen eine Kontaktnummer, um telefonische Bestätigungen einzuholen. Start- und Zielpunkte können Kreuzungen oder bekannte Punkte (Bahnhof, Rathaus, Park etc.) sein. Um mögliche Personen für Vermittlungen einzugrenzen, wird eine *Matchlist* generiert, welche Personen aus einem Radius von 15% der Wegelänge heraussucht[4]. Dieser Radius kann optional auf 25% der Wegelänge erhöht werden, um die Chance einer Vermittlung zu erhöhen. Das SST bietet dabei für ein Ridesharing-System die idealen Voraussetzungen, da „nur“ die Universitätsangestellten und Studenten teilnehmen: Somit ergeben sich ideale Zielgebiete, da quasi alle Personen das gleiche Ziel haben. Insgesamt konnten ca. 35.000 Personen teilnehmen, wobei allerdings ca. 70% dieser Menge auf dem Campus oder in Fußmarsch-Nähe der Universität wohnten.

Ergebnisse dieses Systems liegen mir zu diesem Zeitpunkt noch nicht vor, da das beschreibende Papier [Joh96] keine Auskunft darüber gibt.

3.1.5 Houston, Texas

Im Rahmen des *Houston Smart Communter*-Projektes gibt es ein Teilprojekt, das sich mit Real Time Ridesharing in einem in Richtung Stadtrand gelegenen Arbeitsgebiet beschäftigt und Mitte 1996 in Betrieb genommen werden sollte. Die teilnehmenden Personen lassen sich in einer Art *Call Center* telefonisch oder per Fax registrieren und

[4]Der hier erwähnte Radius wird im Kapitel 6.5.2 (Abbildung 6.22) wieder auftauchen. Dabei geht es um die Frage nach einer optimalen Reduktion aller Systemteilnehmer auf diejenigen, die für einen bestimmten Vermittlungswunsch in Frage kommen.

können anschließend Angebote und Nachfragen auf dem gleichen Wege tätigen. Die Angabe der Start- und Zielpunkte erfolgt dabei über die bereits erwähnten ZIP-Codes. Zusätzlich zu diesen Angaben können persönliche Präferenzen wie Geschlecht des Anbieters/Nachfragers, Raucher/Nicht-Raucher etc. angegeben werden[5]. Dabei ist es geplant, den Service zunächst manuell – also mit Bedienpersonal – zu gestalten und später die Vermittlung und Kommunikation vollautomatisch zu gewährleisten.

Analog zum SST-System in Seattle stehen über die Erfolge z.Zt. keinerlei Daten zur Verfügung.

3.1.6 Ontario, Kalifornien

Für den Zeitraum von Februar 1996 bis Januar 1997 wurde das Projekt *ATHENA* geplant und implementiert. Ähnlich wie bei den bereits oben beschriebenen Systemen können registrierte Personen sich beliebig oft mit dem System in Verbindung setzen und Angebote sowie Nachfragen individuell mitteilen. Der Computer in der Vermittlungszentrale kennt die genauen Aufenthaltsorte der Pkw, welche ihre Position mittels GPS[6] dem Rechner mitteilen. Anfragen und Mitteilungen schreiben bzw. erhalten die Anbieter über *Personal Digital Assistants* (PDA) in ihren Pkw; dieser PDA kommuniziert über Digitalfunk mit dem Zentralrechner. Bei einer erfolgreichen Vermittlung gibt dieser Rechner eine Nachricht an den Anbieter und weist ihn an, den entsprechenden Nachfrager mitzunehmen. Die Nachfrager setzen sich zunächst telefonisch mit dem System in Verbindung, wobei eine spätere Kommunikation über Pager möglich sein soll. Erste Planungen gehen davon aus, daß ca. 100 Anbieter nach der Halbzeit des Versuchs verfügbar sein werden.

Da dieser Versuch bis Anfang 1997 laufen sollte, sind ebenfalls noch keine Ergebnisse verfügbar.

3.2 Schlußfolgerung

Als wichtigstes Ergebnis ist festzuhalten, daß keines der Systeme eine digitale, vektorisierte Straßenkarte verwendet, um Fahrten von Haus-zu-Haus zu vermitteln; wenn überhaupt, wurden ZIP-Codes verwendet, um die Topographie Lage der Anbieter und Nachfrager einzugrenzen.

Alle hier angesprochenen Systeme stellen Einzellösungen für eine relativ kleine Menge an Personen dar. Was die Systeme angeht, die keinen Zulauf an Anbietern und Nachfragern erhielten, so stellte sich heraus, daß viele von den angebotenen Systemen nichts wußten,

[5] Diese Wünsche können u.a. auch bei dem in dieser Arbeit spezifizierten System CORONA angegeben werden (Kapitel 4 und 7). Für eben dieses System wurde die Anforderungsdefinition bereits um den Jahreswechsel 1995/1996 niedergeschrieben, so daß die Idee zu dieser komfortablen Vermittlung zeitgleich mit den Amerikanern erfolgte.

[6] GPS = Global Positioning System

es nicht bedienen konnten oder wollten und daß es zu umständlich sei, sich für spontane Fahrten (oder Fahrtwünsche) mit dem System auseinanderzusetzen.

Es ist ebenfalls aus diesen Darstellungen der existierenden Systeme zu sehen, daß es keinerlei Einheit gibt: Jedes System hat gewisse Ideen und Vorzüge, aber es gibt keinerlei „Verschmelzung“ zu einem konsistenten vollautomatischen System. Alle vorgestellten Einzellösungen arbeiten zum Teil halbautomatisch und besitzen somit eine recht lange Antwortzeit. Ferner fehlen diverse Komponenten wie die bereits in der Einleitung zu diesem Kapitel angesprochene vektorisierte Straßenkarte. Ohne diese Karte kann keine vernünftige Vermittlung und Berechnung der Routen erfolgen. Es existiert nach dem derzeitigen Stand des Wissens kein vollautomatisches System, welches die im nächsten Kapitel dargelegten Anforderungen bezüglich der Aufgaben und Genauigkeit erfüllt — das CORONA-System stellt somit tatsächlich eine Neuerung auf diesem Gebiet dar.

4. Anforderungsdefinition für eine PersonenLogistik

Das zu erstellende System muß eine Reihe von grundlegenden Anforderungen erfüllen, die hier in vier große Bereiche aufgeteilt werden. Diese Bereiche werden im einzelnen beschrieben und sollen als Eckwerte für die spätere Systementwicklung dienen.

4.1 Funktionale Anforderungen

Das System muß in der Lage sein, die folgenden Funktionen zu gewährleisten:

- Verwaltung der Straßen einer Stadt/Region in Form eines *digitalisierten, vektorisierten* Stadtplans.
- Verwaltung des ÖPNV-Netzes inkl. Fahrpläne, Fahrzeit, Preise etc.
- Aufnehmen und Löschen von Anbietern und Nachfragern nach Transportleistung sowie deren komplette Verwaltung.
- Rücksichtnahme auf Wünsche von Anbietern und Nachfragern.
- Schnittstelle zu Anbietern und Nachfragern zwecks Vermittlungswunsch.
- Vermittlung von Mitfahrten im Pkw. Es soll nur die Vermittlung von einer Person pro Pkw (2er Fahrten) betrachtet werden.
- Vermittlung im gebrochenen Verkehr (Mitfahrt/ÖPNV).
- Rückmeldung der Vermittlung an Anbieter bzw. Nachfrager.
- Finden von Ausweichmöglichkeiten bei Ausfall oder Verspätung einer Transportmöglichkeit sowie Benachrichtigung an Anbieter bzw. Nachfrager.
- Komplette Protokollierung aller Vorgänge, um Berechnungen bezüglich der Fahrzeugkilometerreduktion, der Straßenauslastung, der Fahrzeugauslastung etc. anstellen und visualisieren zu können.

4.2 Leistungsanforderungen

Die vorher geforderten Funktionalitäten müssen auf eine bestimmte Art und Weise ihre Dienste bereitstellen, deren Leistung in den folgenden Punkten definiert wird.

- Die digital-vektorisierten Daten des Stadtplans müssen so aufbereitet sein, daß die Suche eines Weges von A nach B schnell abgeschlossen werden kann (ca. eine Sekunde). Dabei müssen Verkehrsflüsse ebenso berücksichtigt werden, wie geschwindigkeitsbegrenzte Zonen. Die Fahrzeit muß mit der Ausgabe des Weges erfolgen. Ferner müssen die Straßennetze graphisch dargestellt werden können, um das Straßennetz im Ganzen zu überblicken.

- Ebenso wie die Straßendaten der Stadt müssen die Routen und Zeiten des ÖPNV digital erfaßt und verwaltet werden. Die Antwortzeit zur Vermittlung von Haltestelle zu Haltestelle muß ebenfalls kurz sein (eine Sekunde). Die Kosten müssen kalkuliert und mit der Route ausgegeben werden.

- Das System muß in der Lage sein, einen Großteil der Bevölkerung in einer Region (ca. 100.000) als Anbieter und Nachfrager zu verwalten. Neue Teilnehmer müssen dem System sofort zur Verfügung stehen, damit es diese in die Vermittlung einbeziehen kann. Ebenso müssen Personen, die dem Vermittlungssystem nicht mehr angehören wollen, schnell aus diesem entfernt werden können. Anbieter und Nachfrager sollen sich ebenso selbständig dem System bekanntmachen, um eine sofortige Mitgliedschaft zu erreichen und vermittelt werden zu können.

- Jedem Anbieter und Nachfrager soll es möglich sein, von mehreren Orten dem System einen Vermittlungswunsch mitzuteilen. Diese Schnittstelle muß eindeutig und benutzungsfreundlich sein, damit jede Person (auch ohne technische Kenntnisse) sich dem System mitteilen kann. Die Schnittstelle darf nicht allein auf die Funktion der Fahrtenvermittlung beschränkt sein, sondern muß derart offen sein, daß weitere Dienste in Anspruch genommen werden können (z.B. Telefonauskunft).

- Bei einer Vermittlung muß das System auf spezifische Wünsche der Teilnehmer achten und entsprechend handeln. Da nicht auf alle Wünsche geachtet werden kann, sei an dieser Stelle auf den Abschnitt Restriktionen auf Seite 17 hingewiesen.

- Das System muß jede Vermittlung so schnell durchführen, daß es keine größeren Wartezeiten gibt. Die Antwortzeit des Systems sollte innerhalb weniger Sekunden liegen. Sollten Alternativrouten gewünscht werden, so muß das System diese ebenso schnell bereitstellen.

- Damit Anbieter und Nachfrager eine Bestätigung erhalten, muß ein entsprechendes Dokument an jede Partei ausgehändigt werden. Der Inhalt des Dokumentes

bezieht sich dabei auf Zeiten, Routen und Preise. Die Erstellung eines solchen Dokumentes muß ebenfalls innerhalb von Sekunden erfolgen, um große Wartezeiten zu vermeiden.

- Für den Fall des Ausfalls eines Verkehrsmittels oder eines Nachfragers muß eine Mitteilung an die Beteiligten erfolgen, um Umwege oder Wartezeiten zu sparen. Zudem muß bei Ausfall der Mitfahrgelegenheit ein Ersatzplan bereitgestellt werden, um die Mobilität zu gewährleisten. Diesen Plan muß das System in Sekunden bereitstellen und alle beteiligten Personen benachrichtigen. Der Dialog, der bei einem Ausfall zu führen ist, muß klar, einfach und eindeutig sein; er muß schneller geführt werden können, als bei normaler Vermittlung, damit nicht unnötig Zeit verloren geht.

- Da das System mittels Vermittlung von Personen helfen soll, Fahrzeugkilometer einzusparen und die Straßen zu entlasten, müssen alle relevanten Vorgänge protokolliert werden. Dazu zählen insbesondere alle gefahrenen Wege der Teilnehmer, auch die Wege, die ohne erfolgreiche Vermittlung gefahren werden würden. Aus diesem Verhältnis muß sich die jeweilige Straßenbelastung sowie die Ausnutzung der Kapazitäten dieser Fahrzeuge berechnen lassen. Zusätzlich müssen diese Werte graphisch dargestellt werden, um interpretierbare Aussagen machen zu können. Die Statistiken müssen jederzeit abrufbar sein und den aktuellen Stand aufweisen.

- Im System müssen Vermittlungen im gebrochenen Verkehr (umsteigen ÖIV in den ÖPNV) realisiert werden. Dabei sind ebenso die Startzeit-Toleranzen wie Umsteigezeiten und Wegezeiten zu/von der Haltestelle zu berücksichtigen. Ebenso soll die Priorität bei der Vermittlungsreihenfolge angegeben werden können (ÖIV, ÖPNV, gebrochener Verkehr).

4.3 Handhabungsanforderungen

Das zu erstellende System kann bezüglich der Handhabung für zwei Benutzergruppen definiert werden. Zum einen die *SysOp-Ebene*, welche die Systembenutzung im VSM-Betrieb beinhaltet, zum anderen die *Benutzer-Ebene*, die sich an den normalen Menschen wendet, der die Dienste des Systems in Anspruch nehmen will.

4.3.1 SysOp-Ebene

Das System soll weitestgehend vollautomatisch arbeiten. Anbieter und Nachfrager tragen sich individuell in die Personenliste ein oder aus. Vermittlungswünsche werden ebenfalls von den jeweiligen Gruppen dem System mitgeteilt. Diese Mitteilung wird durch ein Terminal realisiert, welches mehrere Dienste anbieten kann. In der VSM-Zentrale, in der

die Rechenanlage die Vermittlungen durchführt, müssen System-Operatoren die Funktionsfähigkeit gewährleisten. Das Vermittlungssystem muß einfach zu handhaben sein und alle Informationen zur Verfügung stellen. Ebenso muß es möglich sein, daß Anbieter, Nachfrager, Vermittlungswünsche etc. direkt eingespeist werden können, da nicht jede Person über ein Terminal verfügen kann (z.B. Vermittlungswunsch von einer Person in einer Telefonzelle).

4.3.2 Benutzer-Ebene

Da das System für jeden zugänglich sein soll, muß ein einfach zu führender Systemdialog existieren. Ein solcher Dialog ist zwar per Telefon denkbar, vermittelt allerdings nur unzureichende Rückmeldungen und ist z.B. für die Stand- und Zielorteingabe nicht sinnvoll. Eine zufriedenstellende Lösung stellt eine Art Terminal dar, welches aus einem flachem Bildschirm und einer einfachen Tastatur besteht. Dieses Terminal ist per Telefonleitung mit dem System verbunden und soll über eine einfache, übersichtliche Maske einen kurzen, aber ausreichenden Dialog mit dem Benutzer gewährleisten. Ein kleiner Drucker an der Terminalseite kann auf Wunsch eine Bestätigung der Vermittlung, sowie diverse andere Daten (Fahrpläne, Preise etc.), ausdrucken. Dieses Benutzungssystem muß eine Hilfe- sowie eine Korrektur-Funktion beinhalten, damit Fehleingaben und Unwissenheit keine negativen Auswirkungen haben. Durch den multifunktionalen Charakter des Terminals können weitere Dienste verfügbar gemacht werden. Die Systemführung muß für alle Dienste ähnlich sein, so daß ein Wiedererkennungseffekt eintritt und die Benutzung nicht erschwert wird. Zu dem Thema der Ein- und Ausgabe sei auf den Abschnitt 4.6 hingewiesen, der sich explizit mit der Kommunikation zwischen Mensch und System befaßt. Aus Zeitgründen ist die Ein-/Ausgabekomponente lediglich auf eine direkte Kommunikation zwischen Benutzer und Oberfläche realisiert worden.

4.4 Einbettungsanforderungen

Personen, die die Dienste des Systems in Anspruch nehmen, sollen die Interaktion per Terminal o.ä. durchführen. Ein solches Medium stellt einen eindeutigen Dialog sicher, so daß Unklarheiten von vornherein ausgeräumt sind und die Benutzer eine Rückmeldung (durch Ausdruck auch in schriftlicher Form) erhalten. Das Datenübertragungsmedium kann hierbei die Telefonleitung sein; wie bei Mail-Box- oder T-Online-Diensten, da diese flächendeckend verfügbar sind; ebenso kann eine Übertragung per Funk erfolgen, so daß ortsunabhängige Eingaben oder Benachrichtigungen realisierbar sind. Beispielsweise kann in jedem Haushalt, in dem ein Telefon installiert ist, ein Terminal angeschlossen werden. Die Einrichtung eines Kommunikationsmediums muß für die benutzende Person so kostengünstig wie möglich gemacht werden. Da aus Zeitgründen eine Realisierung von Teilen der Kommunikation nicht möglich war, soll in der späteren Implementation nur die direkte Kommunikation zwischen Benutzer und Oberfläche betrachtet werden.

4.5 Restriktionen

Das System muß sehr flexibel in Bezug auf die Wünsche der Kunden reagieren und möglichst alle Erwartungen erfüllen, da ansonsten die Vermittlung zum Scheitern verurteilt wird. Niemand wird ein System nutzen wollen, welches nur Unbequemlichkeiten verursacht. Dennoch sind nicht alle Wunsch-Variationen realisierbar, und es müssen Restriktionen in Kauf genommen werden. Die folgenden gewünschten Eigenschaften bei Mitfahrten muß das System auf jeden Fall beinhalten, wobei Kombinationen ebenfalls möglich sein sollen:

- Mitnehmer / Mitfahrer ist Raucher / Nichtraucher
- Ist das Fahrzeug straßen- bzw. schienengebunden, fährt es nach einem bestimmten Fahrplan
- Geschlecht des Mitnehmers / Mitfahrers (wichtig, um beispielsweise eine „Frauen-Mitnahme“ zu realiseren)
- Alter des Mitnehmers / Mitfahrers
- Abfahrzeit / Ankunftzeit
- δRZ (Reisezeit-Toleranz) in Abhängigkeit von der angebotenen Route
- δSZ (Startzeit-Toleranz) in Abhängigkeit von den Verkehrsdiensten
- gebrochener Verkehr
- 2er-Fahrten

Auf diese Punkte wird in Abschnitt 6.1 gesondert eingegangen.

4.6 Dialog

Ein großes Problem stellt die interaktive Kommunikation zwischen Benutzer und System dar. Da die Vermittlungen (weitestgehend) vollautomatisch erfolgen sollen, muß das System die menschliche Sprache und Intuitionen verstehen. In den gängigen vollautomatischen Diensten, zum Beispiel D2-ISY-Fax oder die Bahnauskunft in Aachen, können per natürlicher Spracheingabe oder DTMF-Töne des Telefons (fast) beliebige Informationen abgerufen werden. Im Fall von D2-ISY-Fax (Mannesmann) können sogar Dokumente an jedes Faxgerät vollautomatisch gesendet werden.

Der Dialog des Systems sollte alle sinnvollen Kommunikationsmöglichkeiten bereitstellen.

- **Spracheingabe und -erkennung per Telefon:**
  Wie oben beschrieben, können die Vermittlungswünsche mittels natürlicher Sprache erfolgen. Dabei wird eine strikte Menüführung eingehalten, so daß der Benutzer keine Fehleingaben tätigen kann und das System vor falschen Interpretationen geschützt wird. Das System fragt nach Name, Adresse etc., Start- und Zielort, Abfahrt- bzw. Ankunftzeit, Reisezeit-Toleranz und den Mitfahrwünschen, je nach Mitnehmer oder Mitfahrer.

- **DTMF-Menüwahl per Telefon:**
  Diese Methode stellt ein Menü sprachlich zur Verfügung und die Teilnehmer können die Wünsche, Zeiten etc. durch die vorgestellten Menüpunkte identifizieren und per DTMF-Ton wählen. Probleme treten bei den Start- und Zielorten auf, da nicht alle Straßen sequentiell vorgelesen werden können. Abhilfe schafft hier eine alphabetische Selektion: Der Benutzer drückt `1` für `A`, `2` für `B` etc. Anschließend listet das System die nächste Ebene auf; `1` für `Ac`, `2` für `Ad` etc. Somit kann sich jeder Teilnehmer zum jeweiligen Ort vortasten. Diese Methode ist nicht unbedingt für schnelle, einfache Eingaben geeignet. Richtig kompliziert wird es, wenn ein Name eingegeben werden soll; dieser müßte dann vollständig durch Tasten angegeben werden, bietet allerdings den Vorteil, daß von nahezu jedem Telefon Eingaben getätigt werden können.

- **Texterkennung und -ausgabe per SMS (*Short Message System*):**
  Dieser Dienst der Mobilfunk-Anbieter kann genutzt werden, um Eingaben und Benachrichtigungen im Klartext zu übermitteln. Beispielsweise kann eine Eingabe wie folgt realisiert werden: Das erste Wort ist der Name, das zweite der Vorname, das dritte und vierte Wort gibt den Start- und Endpunkt an, das fünfte Wort ist die Uhrzeit etc. Eine Kombination von Spracheingabe und SMS liefert gute Ergebnisse, da die Anfrage mittels natürlicher Sprache getätigt wird, und die Bestätigung oder sonstige Mitteilungen (bei Ausfall oder Änderung einer Vermittlung) per SMS im Klartext empfangen und gelesen werden kann.

- **Menüführung per Mail-Box der VSM-Zentrale:**
  Mit einem PC und einem Modem kann über eine Ein- und Ausgabemaske der VSM-Mail-Box ein intuitiver und interaktiver Dialog geführt werden, bei dem alle Ein- und Ausgaben im Klartext erfolgen.

- **Menüführung per T-Online über Terminal/PC:**
  Analog zum oben erwähnten Mail-Box-Verfahren, kann der T-Online-Dienst

zur Kommunikation genutzt werden. Von Vorteil ist hierbei die Möglichkeit, sich dem System von einem öffentlichen Terminal (Bahnhöfe, Post etc.) aus mitzuteilen.

- **Menüführung per HTML-Seite und JAVA via Internet:**
  Jede Person, deren Rechner an das Internet angeschlossen ist, kann über eine Hompage der VSM-Zentrale Dokumente abrufen, mittels derer die Kommunikation stattfinden kann. Durch die Programmiersprache JAVA ist es weiterhin möglich, größere Flexibilität bei dieser Interaktion zu erreichen. Benachrichtigungen können per e-mail versandt werden.

- **Manuelle Eingabe per Bedienpersonal:**
  Da man nicht davon ausgehen darf, daß jeder Teilnehmer über eines der oben aufgeführten Medien verfügt oder damit umgehen kann, muß es ebenfalls möglich sein, eine manuelle Eingabe durchzuführen. Zu diesem Zweck kann eine Person zur VSM-Zentrale oder einer Filiale gehen bzw. mittels Telefon dort anrufen und sich und ihre Wünsche vom Bedienpersonal eingeben lassen. Benachrichtigungen können dann per Telefon vom Personal oder per Brief erfolgen. Dieses dauert selbstverständlich länger und ist für spontane Nachrichten oder Änderungen nicht zu empfehlen.

Der Dialog-Modul sollte zusätzlich zur Kommunikation Schnittstellen zu einer Datenbank besitzen, in welcher die jeweiligen Teilnehmer mit ihren Attributen abgelegt sind. Die eingegebenen Daten müssen der Datenbank hinzugefügt und Ergebnisse abgerufen werden, so daß eine manuelle oder vollautomatische Benachrichtigung eingeleitet werden kann.

Die hier beschriebenen Dialogkomponenten sind in der Dissertation nicht realisiert worden; es existiert allerdings ein Eingabedialog, der individuelle Angaben zum Vermittlungswunsch akzeptiert (siehe dazu Kapitel 7).

4.7 Generator

Der Generator soll anstelle des Dialog-Moduls verwendet werden, um dessen Funktionen zu simulieren. Dabei handelt es sich um generierte Daten bezüglich Verteilung von Wohnungs- und Arbeitsstätten sowie der Start- und Zielpunkte der Anbieter und Nachfrager im ÖIV, wobei die digitale, vektorisierte Karte der Stadt die Rahmenwerte festlegt.

4.7.1 Funktionalität

- **Generierung privater Mitnehmer im Öffentlichen Individualverkehr:**
  Mittels eines Parameters kann die Anzahl der privaten Mitnehmer im Öffentlichen Individualverkehr (ÖIV) festgelegt werden. Der Generator verteilt die

Mitnehmer-Fahrzeuge zufällig auf die Wohnungen und legt zusätzlich die Routen fest, die mit dem jeweiligen ÖIV-Fahrzeug zurückgelegt werden. Welche Art von Routen generiert werden, hängt von der prozentualen Verteilung der Gesamtfahrzeuge nach folgenden Kriterien ab:

- Prozentsatz von Personen, die regelmäßige Routen fahren (z.B. Fahrten zur Arbeitsstätte);
- Prozentsatz von Personen, die spontane Routen fahren (z.B. Fahrten zur Einkaufs- oder Freizeitstätte).

Eine regelmäßige Route besteht aus einem Startpunkt und einem Endpunkt. Zusätzlich zu der Routeninformation existieren Abfahrt- und Ankunftzeiten für die Hin- und Rückfahrt. Die Fahrzeiten werden als einfache Regeln spezifiziert, z.B.: „Jeden Tag um 8.00 außer Samstag und Sonntag.“ Aus den Regeln können dann Fahrten für den jeweiligen Tag abgeleitet werden. Eine spontane Route ist eine einmalige Fahrt von einem Startpunkt zu einem Endpunkt zu einer festgelegten Zeit. Darüber hinaus kann eine Verteilung für das Mitfahrangebot angegeben werden, aus der der Generator weitere Attribute für eine Fahrt erzeugt. Zu den Mitfahrattributen zählen:

- Geschlecht,
- Platzbedarf,
- Kindersitz,
- Rauchverhalten,
- besonderer Weg,
- Fahrtunterbrechung,
- Reisezeit-Toleranz und
- Umsteigepunkte.

- **Generierung von kontinuierlichen Mitfahrern:**
  Um diese Art Mitfahrer generieren zu können, kann in etwa auf die gleichen Parameter zurückgegriffen werden, wie bei den semi-professionellen Mitnehmern; es kann die Anzahl der kontinuierlichen Mitfahrer angegeben werden. Der Generator erzeugt dann Routen (Hin- und Rückfahrt) mit den Regeln für die Fahrzeiten. Neben der Route und den Fahrzeiten kann eine prozentuale Verteilung für den Fahrtwunsch angegeben werden. Hierzu zählen:
  - Geschlecht,
  - Platzbedarf,
  - Kindersitz,
  - Rauchverhalten,
  - besonderer Weg,
  - Fahrtunterbrechung,
  - Reisezeit-Toleranz und

  - Umsteigepunkte.

- **Generierung von spontanen Mitfahrern:**
  Bei den spontanen Mitfahrern handelt es sich um Personen, die eine Route nur einmal zu einem bestimmten Zeitpunkt durchführen. Die Anzahl dieser Mitfahrer kann durch einen Parameter gesteuert werden. Der Generator erzeugt die Routen mittels der zusätzlich angegebenen Verteilung der Fahrtwünsche.

Die graphische Darstellung der generierten Routen ist in Kapitel 8 aufgezeigt. Je nach verwendeten Parametern des Generators werden die Routen gleichmäßig über das definierte Gebiet per Zufallszahlengenerator verteilt. Dabei werden die Zahlen des Generators mit der jeweiligen Kanten-ID gleichgesetzt. Zusätzlich besteht die Möglichkeit zur Definition von Clustern[1], um eine erste Annäherung an OD-Matrizen zu erreichen. Mehr zur Definition von Clustern ist in Abschnitt 8.20 zu finden.

4.7.2 Parameter für den Betrieb

- **Vorgabe der MIV-Geschwindigkeit:**
  Dieser Wert gibt an, mit welcher Geschwindigkeit sich ein privater Mitnehmer auf welchen Straßenarten fortbewegt. Die Vorgabe baut auf einem dynamischen Modell auf, in dem einer Klasse von Straßen oder sogar einer einzelnen Straße eine zeitabhängige MIV-Geschwindigkeit zugeordnet wird. Diese ist in CORONA – nach dem derzeitigen Wissen – erstmalig verwendet worden. Die kommerziellen Navigationssysteme von BOSCH oder PHILIPS kennen zwar die erlaubten Geschwindigkeiten, berücksichtigen aber keine zeitabhängigen Geschwindigkeiten, so daß z.B. der Berufsverkehr und die damit verbundene reduzierte Geschwindigkeit unberücksichtigt bleibt.

- **Modifikation der ÖPNV-Fahrpläne:**
  Grundsätzlich ist die Geschwindigkeit (von Haltestelle zu Haltestelle) durch die Fahrpläne festgelegt. Eine Modifikation dieses Wertes erfolgt somit immer durch die Änderung der Fahrpläne. Die Möglichkeit der manuellen Veränderung der Pläne innerhalb des Generators ist sinnvoll, um gewisse Streckenoptimierungen und -änderungen zu simulieren.

- **Modifikation der Fußgänger-Geschwindigkeit:**
  Dieser Wert spielt insofern eine Rolle, da möglicherweise eine Mindestgeschwindigkeit erforderlich ist, um eine Vermittlung zustande zu bringen, zum Beispiel dann, wenn es nötig ist, eine Straßenbahn pünktlich zu bekommen, oder zu einem Treffpunkt zwecks Aufnahme in einen Pkw zu gelangen.

[1] Gebiete, in denen sich die Start- und Zielpunkte der Anbieter und Nachfrager befinden.

- **Modifikation der maximalen Umwegtoleranz:**
  Die maximale Umwegtoleranz, die ein Mitnehmer eingehen will, muß variierbar sein, um Ergebnisse der Auswirkungen bezüglich der Vermittlung, Netzauslastung etc. zu erhalten.

Die ÖPNV-Geschwindigkeit ist durch die entsprechenden Fahrpläne gegeben. In dieser Arbeit wird davon ausgegangen, daß eine ÖPNV-Linie plangemäß fährt. Zusätzlich können Fahrplanänderungen komfortabel vorgenommen werden, so daß z.B. bei Umleitungen oder längeren Ausfällen das System keine undurchführbaren Vermittlungen ausgibt (siehe dazu Kapitel 7).

4.8 Monitor

Der Monitor dient zur Anzeige und Verifikation der gewonnenen Daten aus den Vermittlungen in der Flotte der Miteinander Fahrenden. Er muß sie in verständlicher Form visualisieren, so daß klar und eindeutig zu erkennen ist, welche Auswirkungen durch die Vermittlungen eingetreten sind (Realbetrieb) oder eintreten könnten (Simulation).

- **Anzeige der Generatorparameter / Stadtparameter:**
  Alle Parameter müssen bei einer Simulation und im Realbetrieb des Mitfahrvermittlungssystems (MVS) angezeigt werden können. Dazu zählen die Anzahl der Mitnehmer (IV, ÖIV, ÖPNV) mit ihren jeweiligen Daten, die Anzahl der Mitfahrer mit ihren jeweiligen Daten sowie Verteilungszahlen aller Wohnungen, Arbeitsplätze und Einkaufs- sowie Freizeitzentren.

- **Anzeige der Vermittlungsparameter:**
  Der Monitor muß die für die Vermittlung relevanten Daten zur Verfügung stellen. Hierzu zählen die besonderen Wünsche der Mitnehmer und Mitfahrer.

- **Anzeige der stattfindenden Fahrten:**
  Alle bereits vermittelten Fahrten müssen mit Start- und Zielangabe graphisch auf dem Bildschirm dargestellt werden. Dazu werden sie in die digitale Karte eingezeichnet. Aus den Angaben muß der Startort, der Zielort, die benutzten Verkehrsmittel, die Streckenlänge, die Kosten, die Zeitersparnis, die Kilometerersparnis und die Fahrzeugkilometerreduktion der jeweiligen Fahrt hervorgehen.

- **Berechnung des Vermittlungsgrades aller Fahrzeuge im ÖIV:**
  Der Vermittlungsgrad ist für alle beteiligten Fahrzeuge zu berechnen und textuell sowie graphisch darzustellen.

- **Angabe der Fahrzeugkilometerreduktion im Modell:**
  Die Fahrzeugkilometerreduktion im Gesamtnetz wird berechnet, so daß dieser Wert jederzeit textuell und graphisch ausgegeben werden kann.

- **Berechnung der Gesamtnetzauslastung im Modell:**
  Für die Effizienz des Systems ist es notwendig, die Auslastung des gesamten Straßen- und ÖPNV-Netzes zu berechnen und darzustellen. Dieses muß der Monitor gewährleisten.

- **Überschlagrechnungen zur Gesamtfahrleistungsreduzierung:**
  Um zu Aussagen zu kommen, wie hoch die zu erwartende Fahrleistungsreduktion in der Gesamtflotte der Region ist, soll eine Überschlagsrechnung getätigt werden, damit der VS-Manager eine erste Übersicht bekommt, in wie weit seine Auflagen erfüllt sind.

Zusätzlich zu diesen Daten gibt der Monitor alle relevanten statistischen Daten zur Vermittlung textuell und graphisch aus.

4.9 Zeitfenster

Die Startzeiten der Anbieter und Nachfrager werden in sogenannte *Slots* eingeteilt, so daß die Anzahl aller Miteinander Fahrenden segmentiert wird. Diese Slots sind kleine Zeitfenster, in denen ein definierter Teil von Anbietern und Nachfragern vermittelt wird. Die Slotbreite orientiert sich dabei an der Startzeit-Toleranz. Dieser Wert gibt an, wie lange ein Nachfrager bereit ist, auf den Anbieter zu warten. Beträgt diese Toleranz z.B. 10 Minuten, so wird die teilnehmende Population in 10-Minuten-Slots unterteilt. Dabei werden angebots- und nachfrageabhängige Verteilungsmethoden angewendet, so daß von einer realitätsnahen Slot-Bevölkerung gesprochen werden kann.

Die Abbildung 4.1 zeigt ein Beispiel: Dargestellt wird jeweils einer von sechs Slots pro Stunde bei einer Slotbreite von 10 Minuten. Die Häufungen von Anbietern und Nachfragern sind hier in den Zeiten des Berufsverkehrs zu erkennen.

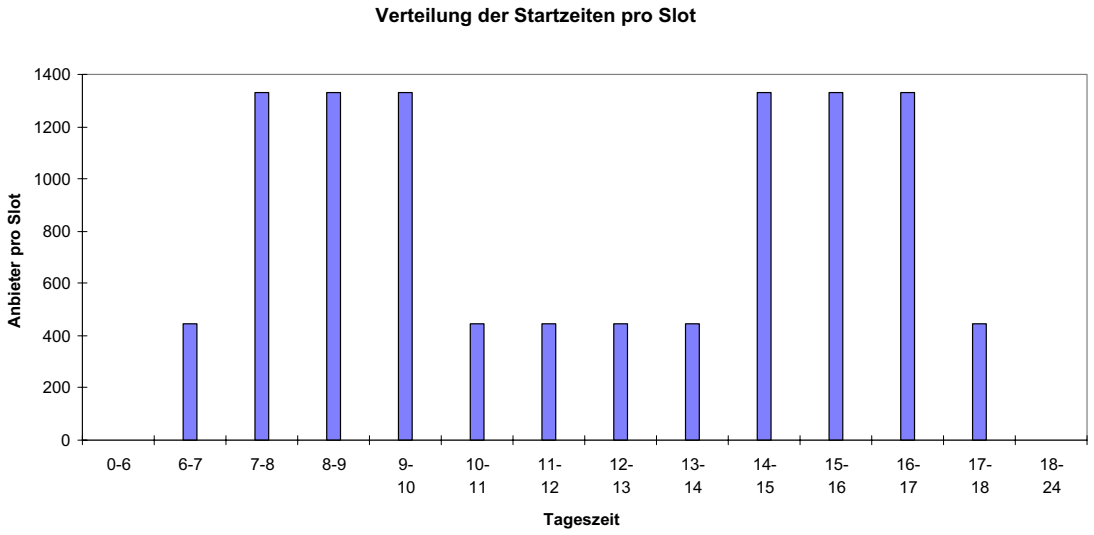


Abbildung 4.1: Slot-Verteilung am Beispiel eines von sechs Slots pro Stunde (Slotbreite: 10 Minuten).

Kann ein Angebot oder eine Nachfrage nicht vermittelt werden, so wird diese in den nächsten Slot übertragen – es sei denn, daß die Startzeit-Toleranz bereits überschritten ist. Alle Berechnungen, die mit CORONA angestellt wurden, sind gemäß dieser Slot-Definition betrachtet worden. Eine genaue Herleitung der Slots und deren Verwendung wird in Kapitel 8 aufgeführt.

5. Struktur und Nutzung vektorisierter Straßenkarten

5.1 Eigenschaften

Um effiziente Vermittlungen in der PersonenLogistik durchführen zu können, wird eine vektorisierte, hoch-auflösende Straßenkarte benötigt. Das Institut für Informatik und Verkehr hat 1996 von der Firma `Tele Atlas` eine entsprechende Karte der Stadt Bremen erworben. Die enthaltenen Informationen werden im folgenden aufgeführt:

- **Straßenart:**
  Die Straßenart wird durch achtstufige Klassifizierungen beschrieben. Sie werden als Autobahnen, Bundesstraßen, Landstraße, Hauptstraßen (regional / überregional), Nebenstraßen und Zufahrtstraßen interpretiert. Zusätzlich sind Informationen darüber enthalten, ob es sich beispielsweise um einen Platz oder eine Fußgängerzone handelt.

- **Verkehrsführungen:**
  Es existieren Informationen darüber, ob eine Straße eine Einbahnstraße ist, ob sie einspurig oder mehrspurig ist, einen Kreisel darstellt oder eine Autobahnabfahrt.

- **Halte- bzw. Parkmöglichkeiten:**
  Die einzig verfügbare Angabe hierzu beschränkt sich auf Parkplatzanbindungen bzw. Parkplatzeinfahrten.

- **Verkehrsflüsse:**
  Es gibt keinerlei Angaben zu Verkehrsflüssen oder Ampelschaltzeiten, die aber nach Konvertierung einbaubar sind.

- **Straßennamen:**
  Die Straßennamen und eventuelle Zusätze (wie z.B. A1 oder B75) sind für jeden Straßenzug verfügbar.

- **Hausnummern:**
  Im Datensatzformat sind momentan Bereiche für Hausnummern reserviert,

aber *nicht* eingetragen. Die reservierten Bereiche sollen Auskunft geben über die Nummerierung einer Straße.

Die digitale Karte bietet darüber hinaus Informationen über Schiffsanbindungen, Straßenzustand (Ausgebaut, Feldweg etc.), Straßengebühren, Klassifizierungen von Wasserwegen, Hotels, Restaurants, Picknick-Plätze sowie über Tankstellen und Flughäfen. Die Visualisierung der für die Vermittlung relevanten Informationen ist in Abbildung 5.1 zu sehen.

Abbildung 5.1: Ausschnitt der Stadt Bremen. Die Visualisierung wurde durch die konvertierte, digitale Straßenkarte erreicht (siehe Kapitel 7).

5.2 Konvertierung

Die digitalen Daten einer Stadt werden in einem Format ausgeliefert, welches sich für die PersonenLogistik nicht gut eignet. Es müßten unnötig viele Listen erstellt und verwaltet werden, so daß Einbußen in der Performanz der internen Strukturerstellung (Abschnitt 6.2) entstehen. Aus diesem Grund habe ich ein Konvertierungsprogramm entwickelt, welche die Datenflut des Originalformates zusammenfaßt. Das Ergebnis der Konvertierung ist eine Kantenliste inklusive aller relevanten Attribute jeder einzelnen Kante[1]. Dieses ist von besonderem Vorteil, da so zu jeder Kante ohne große Umstände Informationen hinzugefügt werden können[2]. Zudem konnte durch die Konvertierung ca. 30% des benötigten Speichers eingespart werden, welches beim Durchsuchen zusätzlich starke Zeitvorteile mit sich bringt.

Für eine Beschreibung des Programms zur Konvertierung sei auf das Kapitel 7 hingewiesen, in welchem das Tool `convertGDF` ebenso wie die resultierende Datenstruktur näher beschrieben wird.

5.3 Ergänzungen

Der entwickelten PersonenLogistik muß die Straßenkarte als Grunddatensatz zur Verfügung stehen, damit sie ihre Aufgaben wahrnehmen kann. Zu diesem Zweck sind die Daten der vektorisierten Straßenkarte nicht nur zu konvertieren sondern auch zu ergänzen.

1. Die fehlenden Informationen müssen beschafft und mit in die digitalen Daten eingebunden werden.
2. Da keine Hausnummerinformationen zur Verfügung stehen, müssen für die Testläufe Approximationen vorgenommen werden. So sollen die Hausnummern aus den Daten der Katasterämter bezüglich des Verhältnisses Straßenbreite zur Breite der Grundstücke herangezogen werden.
3. Da Angaben zur Straßenart (Autobahn, Bundesstraße etc.) und Verkehrsführung (einspurig, mehrspurig etc.) vorhanden sind, können unter Berücksichtigung der durchschnittlichen Grundstücksbreite an diesen Straßen Hausnummern verteilt werden, so daß eine einigermaßen realistische Adressierung gewährleistet werden kann[3].

Zur Zeit liegen diese Daten jedoch nicht in einem verwertbaren Format vor. Da die Daten der Stadt – bzw. des CORONA-Datensatzes einer Stadt – beliebig erweitert werden

[1] Die Knotenliste wird zur Laufzeit von CORONA generiert, siehe Abschnitt 6.2.
[2] Das Hinzufügen von Daten wird bezüglich der Verkehrsflußinformationen durchgeführt.
[3] Dieses entspricht dem geforderten Haus-zu-Haus-Transport.

können, ist es im Prinzip kein Problem, die fehlenden Angaben nachzutragen. Dennoch sollte es bis zu diesem Nachtrag möglich sein, eine realistische Funktionsweise bezüglich der Simulation bzw. des Betriebes zu erhalten. Aus diesem Grund wird zunächst von einer Kante-zu-Kante-Verbindung ausgegangen. Da Straßenzüge als Menge von Kanten repräsentiert sind, ist es möglich, an Straßenabschnitten zu starten oder sie zum Ziel zu erklären; man kann somit auch von einer Art Block-zu-Block-Vermittlung sprechen.

Updates der vektorisierten Straßenkarte werden von der Herstellerfirma (in diesem Fall `Tele Atlas`) angeboten. Diese bringt in regelmäßigen Abständen neues Material heraus; allerdings zu relativ hohen Kosten. Für den Fall, daß geringfügige Änderungen am Straßennetz entstehen, können diese mittels des CORONA-Systems leicht hinzugefügt werden (siehe Kapitel 7), so daß der VS-Manager (bzw. jeder andere Anwender der PersonenLogistik) diese Kosten nicht jedesmal tragen muß.

6. Verwendete Heuristiken

6.1 Matching-Kriterien

Die PersonenLogistik soll über die Möglichkeit verfügen, Vermittlungswünsche zu berücksichtigen. Durch diese Forderung ist es gegeben, daß nur diejenigen Fahrten vermittelt werden, deren Matching-Kriterien erfüllt sind. Jeder Mitnehmer und jeder Mitfahrer gibt seinen jeweiligen Wunsch an. Einige Beispiele von Kriterien sind im Anschluß – unterteilt nach Mitnehmer und Mitfahrer – aufgeführt.

- Abfahrzeit
- δRZ (Reisezeit-Toleranz) in Abhängigkeit von der angebotenen Route
- δSZ (Startzeit-Toleranz)
- Angaben zur Nutzung der gewünschten Verkehrsdiensten
- Mitnehmer ist Raucher / Nichtraucher
- Ist das Fahrzeug straßen- bzw. schienengebunden? Fährt es nach einem bestimmten Fahrplan?
- Geschlecht des Mitnehmers
- Alter des Mitnehmers

Diese Punkte können beliebig erweitert werden; es macht jedoch keinen Sinn, exotische Kriterien aufzunehmen. Durchaus sinnvoll sind dagegen Wünsche wie: „Ich möchte jeden Tag von Montag bis Freitag um 7:30 Uhr abgeholt werden. Mein Ziel ist Bibliothekstraße 1. Ich möchte von Montag bis Freitag um 16:05 Uhr von der Bibliothekstraße 1 nach Hause. Ich möchte nur bei nichtrauchenden Personen unter 25 Jahren mitfahren.“ Dieser Wunsch klingt zwar nicht minder exotisch, macht allerdings Sinn, da so z.B. eine Art *Frauenmitnahme* – ähnlich dem Frauentaxi – realisiert werden kann. Selbstverständlich können ebensolche Kriterien für Mitnehmer festgehalten werden: „Ich fahre ... zur Straße ... und nehme nur Nichtraucher mit. Ich biete zwei Kindersitze und einen großen Kofferraum an... Ich bin bereit, bis zu 10% meiner Reisezeit zusätzlich zu fahren.“

Mittels dieser Matching-Kriterien wird eine Auswahl an Personen getroffen, welche die Zahl der zu berechnenden Wege reduziert, da nur im Falle der Erfüllung aller geforderten Kriterien eine Vermittlung stattfinden kann. Das δRZ allerdings kann erst nach erfolgter Wegesuche überprüft werden, so daß es im Falle einer Auswahl bezüglich der oben aufgeführten Kriterien nicht ins Gewicht fällt.

Im folgenden Abschnitt werden zunächst die Such- und Vermittlungsalgorithmen beschrieben, welche zum Auffinden eines Weges dienen. Im Anschluß werden diverse Methoden betrachtet, mit deren Hilfe unnütze Wege herausgefiltert werden können. Da es eine wesentliche Forderung an das System ist, daß durch Heuristikanwendung keine potentiellen Mitfahrer übersehen werden, wird im Laufe dieses Kapitels eine heuristische Methode entwickelt, die diese Forderung erfüllt und dennoch genügend Wege reduziert.

6.2 Routensuch- und -vermittlungsalgorithmus

6.2.1 Der verwendete Algorithmus

Die Rohdaten der konvertierten, vektorisierten Stadt liegen im ASCII-Code vor. Die jeweiligen Datensätze beinhalten neben geographischen Koordinaten der Kanten auch deren Name und eindeutige Identifikationsnummer innerhalb der Datei sowie Informationen zu Straßenart, Abbiegevorschriften etc. Nun gilt es, diese Daten derart im Speicher eines Rechners abzulegen, daß möglichst schnell auf bestimmte Kanten zugegriffen werden kann, um Knoten zu lokalisieren und Wege zu suchen.

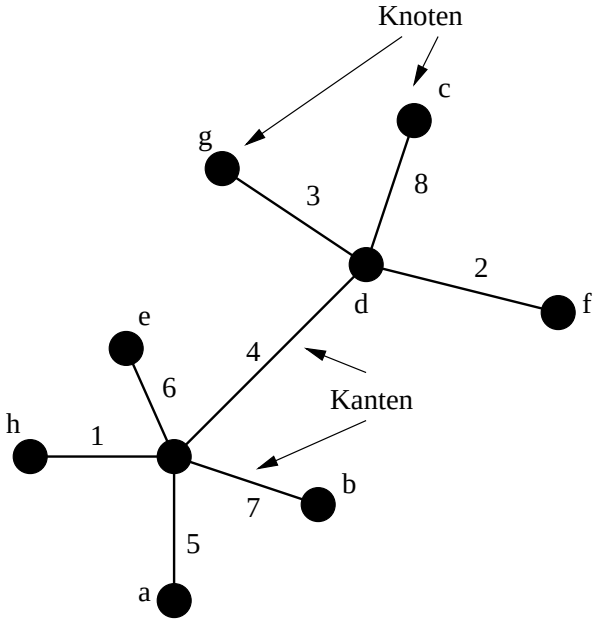


Abbildung 6.1: Beispiel eines Graphen.

Die Abbildung 6.1 stellt einen sehr kleinen Graphen dar, welcher zur Erklärung des Algorithmus jedoch ausreicht. Da die Kartendaten nur Kanten bereitstellen, müssen zunächst die Knoten herausgesucht werden. In den Datensätzen sind Informationen über `from-` und `to-`Punkte der jeweiligen Kante enthalten, welche als Knoten verstanden werden. Somit wird zunächst eine *Knotenliste* erstellt, da diese nicht explizit in den Datensätzen enthalten sind (siehe Kapiten 5). Jede Kante zeigt auf ihren jeweiligen `from-` bzw. `to-`Knoten. Für jeden Knoten wird eine Liste aller hin- und wegführenden Kanten generiert, so daß jeder Knoten auf diese Liste verweist. Jedes Listenelement dieser *Knotenelementliste* zeigt wiederum auf die ursprüngliche *Kantenliste*. Die Graphik in Abbildung 6.2 zeigt ansatzweise, wie diese drei Listen miteinander verbunden sind.

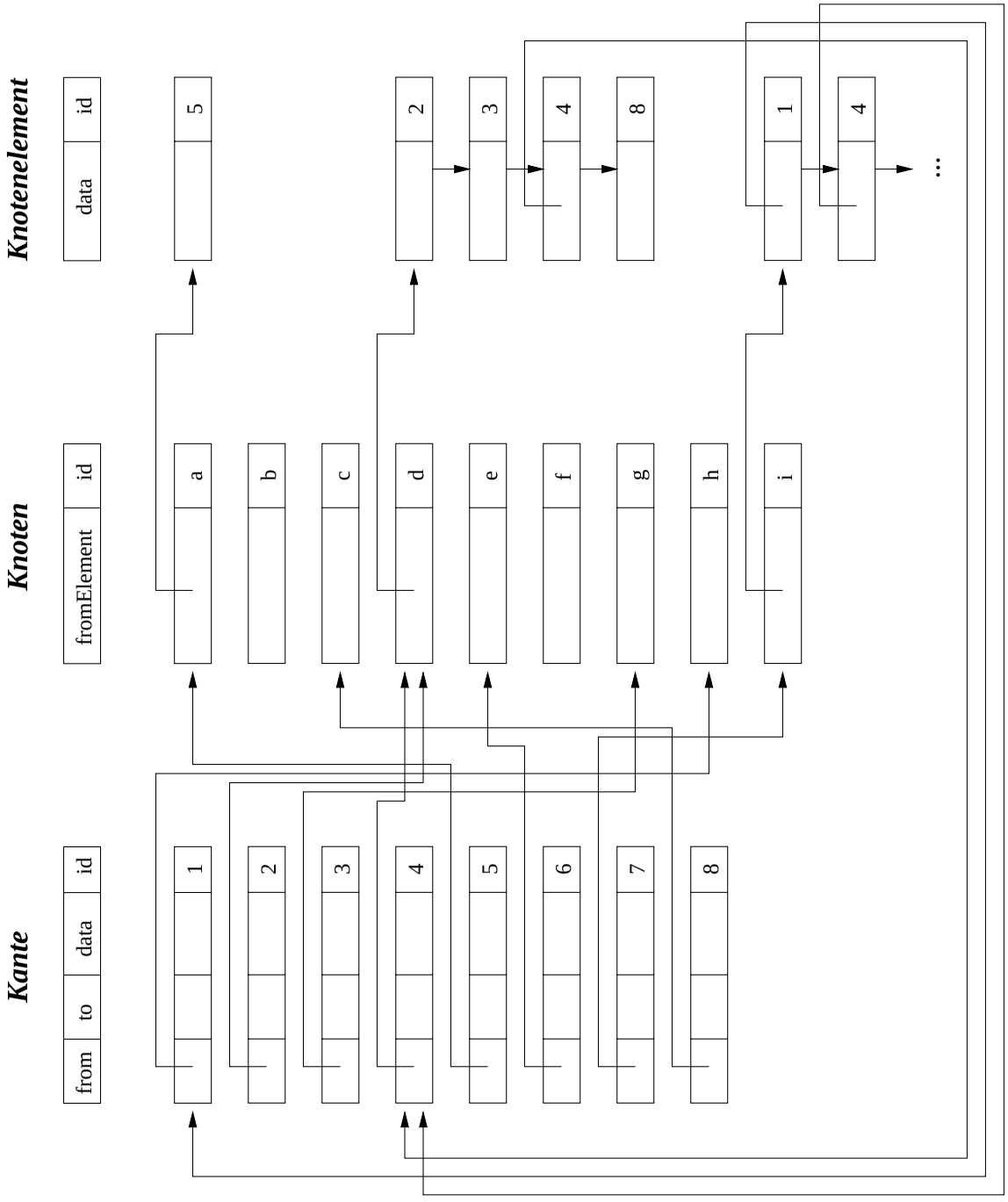


Abbildung 6.2: Darstellung des Zusammenhangs zwischen Kantenliste, Knotenliste und Knotenelementliste im Speicher.

Somit kann von jeder Kante aus direkt auf jeden Knoten zugegriffen werden; ist der Startpunkt erst einmal bekannt, wird der Graph durch Zeigeroperationen durchsucht.

Diese Suche geschieht nach dem Prinzip der *Breitensuche*, da sich diese als optimale Suchstrategie herausgestellt hat, da hier die Kanten weitestgehend die gleiche Länge (Wertigkeit) besitzen. Optimal ist diese Suche deshalb, weil bei herkömmlichem Vorgehen Probleme auftreten, welche die Realität außer acht lassen: Normalerweise sucht ein Algorithmus den kürzesten oder schnellsten Weg; je nach Gewichtung der Kanten ([Tur96] und [Sed92]).

Für die angestrebte Korrektheit der Berechnungen und in bezug zu den Abgrenzungen muß der Suchalgorithmus vernünftige Wegen finden. Berechtigterweise stellt sich hier die Frage, was vernünftig im wissenschaftlichen Sinne bedeutet. Im Fall des Suchalgorithmus bedeutet dieses, daß die gefundenen Wege *im Mittel korrekt* sein müssen; sie dürfen in ihrer Länge nicht stark von kürzesten Wegen abweichen. Der hier vorgestelle Algorithmus sucht somit nicht den kürzesten Weg, sondern *einen kurzen Weg*. Ferner gibt es *erhebliche Zeitprobleme*, die eine Verwendung eines schnelleren Algorithmus erfordern. Eine genauere Beschreibung der Komplexität der diversen gängigen Algorithmen, sowie die entsprechende Begründung für die Verwendung des hier beschriebenen Algorithmus finden sich in Abschnitt 6.2.3 und 6.2.4.

Die kurzen Wege, die der verwendete Suchalgorithmus findet, sind deshalb kurz, weil er den Graphen kantenweise durchsucht; er tastet sich (breitensuchgemäß) durch den Graphen, bis er an dem definierten Zielknoten angelangt ist und terminiert. Somit ist gewährleistet, daß der gefundene Weg keinesfalls der längste Weg ist, sondern zu den geforderten im Mittel korrekten Wegen gehört. Dieses Verfahren wird in Abbildung 6.3 beschrieben.

```
success = false;
do
  if(source_id != dest_id) then
    AddFatherRelation;
    AddNodesToList;
  else
    success = true;
    exit;
  source_id = GetNextNode;
while(ExistEdges);
```

Abbildung 6.3: Algorithmus zum Suchen von *kurzen* Wegen.

Der Algorithmus überprüft solange die aktuelle Knoten-ID (`source_id`) mit der Ziel-ID (`dest_id`), bis diese übereinstimmen oder keine weiteren Knoten existieren. Bis einer dieser Fälle eintritt, wird der zurückgelegte Weg in Form von Vater-Beziehungen in die entsprechenden Datenstrukturen eingetragen. Anschließend werden alle wegführenden Knoten in die Breitensuchliste eingetragen, und die aktuelle ID auf den nächsten Knoten gesetzt. Nach der Terminierung des Algorithmus kann der zurückgelegte Weg anhand der Vater-Beziehungen wiedergewonnen werden.

Die Schritte, die zum Generieren der hierfür notwendigen Graphrepäsentation notwendig sind, werden in zwei Bereiche unterteilt.

1. Die *Knotenrelationen* erstellen (Abbildung 6.4).
2. Die *Kantenrelationen* erstellen (Abbildung 6.5).

Diese Graphenstruktur wird initial erstellt; sie benötigt relativ viel Zeit, bietet aber den Vorteil, daß durch diese Anordnung der Datensätze das Suchen eines Weges extrem beschleunigt wird.

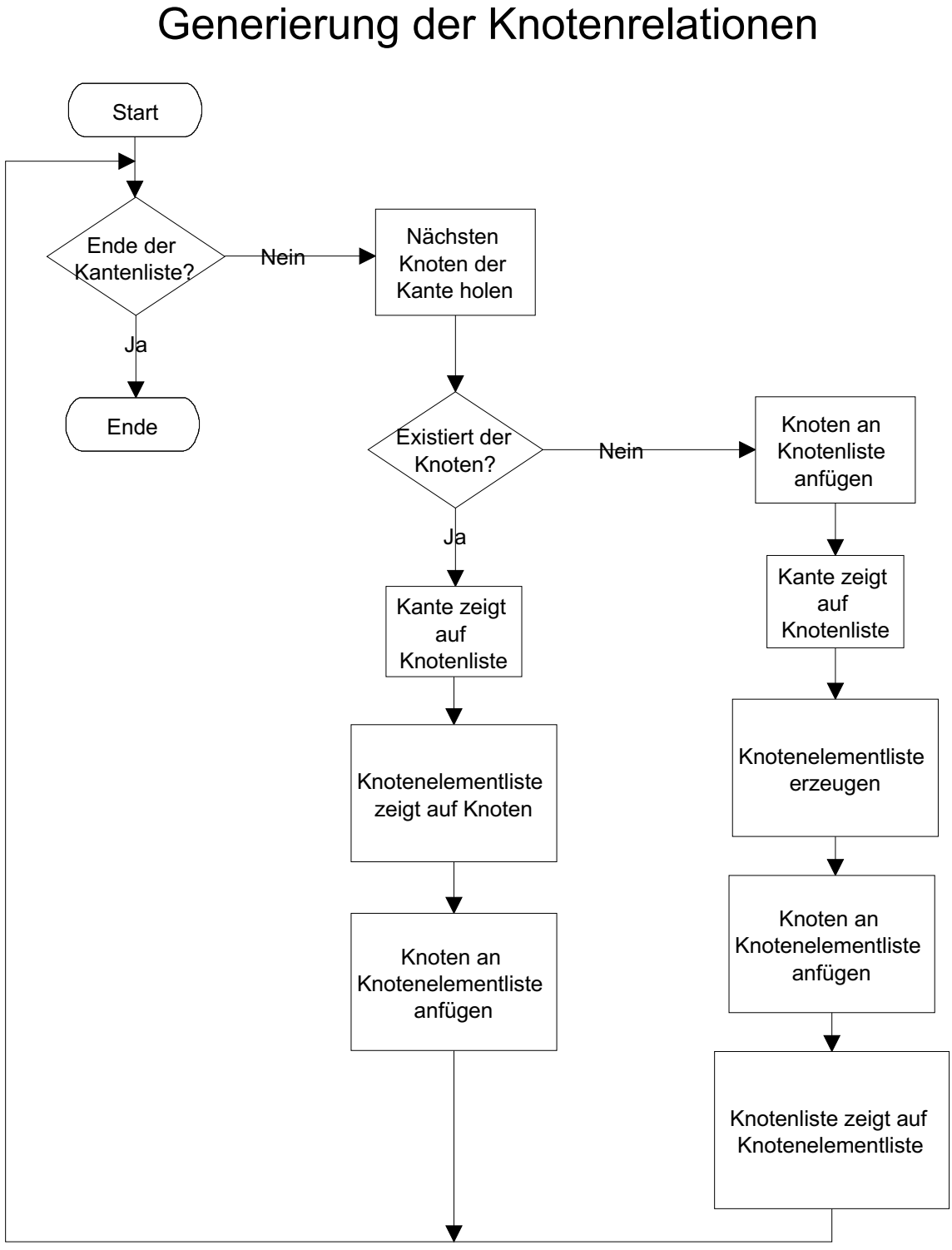


Abbildung 6.4: Notwendige Schritte zum Erzeugen der Knotenrelation.

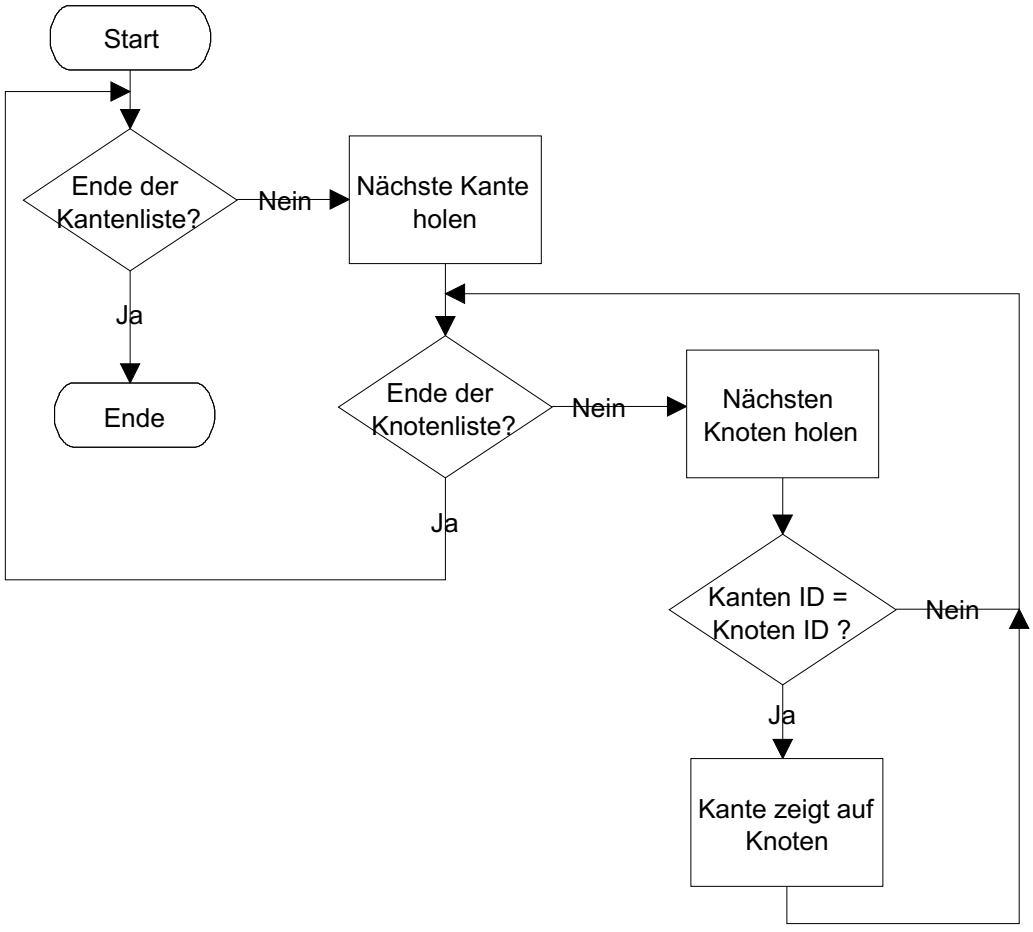


Abbildung 6.5: Notwendige Schritte zum Erzeugen der Kantenrelation.

6.2.2 Optimierungsversuch

Der vorgestellte Algorithmus sucht nach *Brute-Force*-Art alle Wege innerhalb eines Graphen durch, bis er an einem Ziel angekommen ist. Es liegt nahe, den Algorithmus zu optimieren. Die Abbildung 6.6 zeigt das Konzept eines Optimierungsversuches der Wegesuche durch Eingrenzung des Suchgebietes: Ein Weg führe von O nach D. Man nehme von dieser Verbindung die Luftlinie und deren Mittelpunkt als Mittelpunkt M eines Kreises. Zudem definiere man einen Toleranzwert δ, welcher den Kreis im Radius r_0 ein wenig vergrößert.

$$\begin{aligned} M &= \frac{\overline{OD}}{2} \\ r_0 &= \overline{MD} + \delta \end{aligned} \tag{6.1}$$

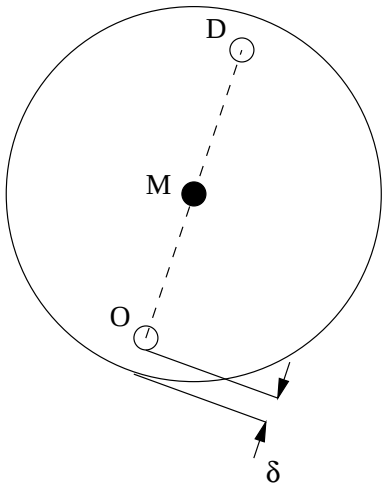


Abbildung 6.6: Eingrenzung der möglichen Wege durch einen Kreis.

Nun soll jeder Knoten überprüft werden, ob er in diesem Kreis liegt; wenn ja, dann sollen die abgehenden Kanten weiterverfolgt werden, ansonsten wird der Knoten nicht weiter betrachtet. Die Überprüfung wird mittels der *Verschiebungsform der Kreisgleichung* vorgenommen:

$$(r - r_M)^2 = \varrho^2 \tag{6.2}$$

Dabei ist $r_M = M$ und $\varrho^2 = r^2$ aus Gleichung 6.1. Durch Einsetzen der Koordinaten des jeweiligen Knotens in r kann die geforderte Überprüfung stattfinden.

Testläufe haben ergeben, daß durch diese Methode bis zu 60% der Wege eingespart werden können; Wege, die ohne diese Optimierung vergeblich durchsucht wurden und

nicht zum Ziel führten. Die Zeitmessung ergab allerdings eine *Verschlechterung*, welche zu erwarten war, da die Überprüfung der Knoten eine höhere Laufzeit beansprucht, als die *Brute-Force*-Methode. Ein Beispiel mit einem δ-Wert von 10% des Radius soll dies verdeutlichen:

> Um von einer Kante zur nächsten zu gelangen, wird mittels der Struktur aus Abbildung 6.2 eine reine Speicher-Zeigeroperation angewendet, welche – per Definition – einen Taktzyklus beanspruchen soll. Die Überprüfung, ob ein Knoten im Kreis liegt, soll zwei Taktzyklen andauern. Dieser Wert bzw. dieses Verhältnis ist realistisch, da innerhalb der Verschiebungsform addiert, subtrahiert, quadriert und verglichen wird. Somit dürfte diese Prozedur auf jeden Fall länger dauern, als eine reine Neusetzung eines Speicherregisters. Bei 60% Ersparnis müssen 40% der Knoten überprüft worden sein, ob diese im Kreis liegen. Somit werden unter der Annahme, daß 100 Knoten existieren, 40 Taktzyklen zur Suche und $2 \cdot 40 = 80$ Taktzyklen zur Prüfung – insgesamt also 120 Takte – benötigt. Ohne diese Kreis-Optimierung würden alle 100 Knoten durchsucht werden; jedoch jeder mit einem Taktzyklus.

Erst bei einer Ersparnis von mehr als 2/3 der Wege würde sich ein Geschwindigkeitszuwachs einstellen. Eine stärkere Eingrenzung des Suchgebietes (indem man den δ-Wert reduziert) ist allerdings nicht sinnvoll, da ansonsten einige Wege nicht gefunden werden, und somit die Vermittlung gefährdet ist. Wie aus diesem Beispiel zu ersehen ist, ist die *Brute-Force*-Methode, bzw. die Breitensuche ohne jegliche Heuristik, effektiver. Bedingt durch die Anordnung der Listenelemente ist das Verwenden dieser Methode – auch wenn sie weniger elegant ist – durchaus gerechtfertigt.

6.2.3 Vergleich zu gängigen Suchalgorithmen

Wie oben erwähnt, wurde in der Implementation der PersonenLogistik ein einfacher Suchalgorithmus verwendet, da dieser deutliche Zeitvorteile mit sich bringt und dennoch im Mittel korrekte Ergebnisse liefert. Das Problem der Zeit kann durch Abschnitt 6.3 verdeutlicht werden. Tatsache ist, daß die PersonenLogistik sehr viele Wege berechnen muß, um zu den geforderten Ergebnissen zu kommen. Ferner ist der Graph, auf dem der Algorithmus die Wege suchen muß, *extrem groß* und somit nicht mit den in der Literatur üblich verwendeten abstrakten, einfachen Graphen zu vergleichen. Die Stadt Bremen, die in dieser Arbeit als Graphstruktur verwendet wird, verfügt über 13.709 Knoten und 42.578 Kanten. Sollen in diesem riesigen Graphen Wege gesucht werden, so sind geringe Zeitunterschiede bei kleinen Graphen hier deutlich zu spüren. Ferner müssen viele Wege in möglichst kurzer Zeit gesucht werden, welches die Zeitproblematik weiter verdeutlicht. In den unten aufgeführten Betrachtungen sei e die Anzahl aller Kanten und v die Anzahl aller Knoten.

Dijkstra-Algorithmus:

Als erstes soll der Dijkstra-Algorithmus daraufhin untersucht werden, ob er die geforderten zeitlichen Bedingungen – und somit eine akzeptable Komplexität – erfüllt ([Jun90], S. 64 ff. und [AHU87], S. 203 ff.): Dieser Algorithmus hat in seiner reinen Form die Komplexität $O(v^2)$. Verwendet man eine Adjazenzliste in Verbindung mit einer sortierten Prioritätenliste, in welcher die Elemente nach Entfernung geordnet sind, so ergibt sich eine weitaus geringere Komplexität mit $O(e \cdot \ln v)$. Da in dem Graphen *Bremen* über 13.000 Knoten existieren, und $\ln 13.000$ ca. 9.4 ergibt, ist die Komplexität somit mehr als neunfach größer als der von mir verwendete Algorithmus mit $O(e)$. Bedenkt man zudem noch die Band- oder Speicherkomplexität [HU92], dann ist die Komplexität des Dijktra-Algorithmus ebenfalls um den Faktor $\ln v$ größer.

Floyd-Warshall-Algorithmus:

Dieser Algorithmus, der auf der Abstandsmatrix von Floyd beruht, berechnet die Abstände zwischen je zwei Punkten – im Gegensatz zu dem oben beschriebenen Dijkstra-Algorithmus, der den Abstand zu *einem* Punkt angibt[1] ([Jun90]). Diese Variante hat die Komplexität $O(v^3)$ und ist somit *noch* aufwendiger als die oben aufgeführte Dijkstra-Variante. Der Floyd-Warshall-Algorithmus und der Dijkstra-Algorithmus, die die Verbindungen bzw. die Abstände zwischen je zwei Punkten berücksichtigen, zählen zu der Gruppe der *APSP*-Probleme[2].

Diese beiden Algorithmen sind die üblichen, um in Graphen nach kürzesten Wegen zu suchen. Die Größe des Graphen *Bremen* und die erwartete Anzahl an zu suchenden Wegen machen deutlich, daß in diesem Fall ein anderer Algorithmus das Testergebnis auf Korrektheit der PersonenLogistik nicht beeinflußt. Weiterhin wird im nächsten Abschnitt der hier verwendete Algorithmus daraufhin überprüft, ob dieser in der Tat im Mittel korrekte Wege liefert.

In [Ert98] ist eine stark modifizierte Art des Dijkstra-Algorithmus aufgeführt, der speziell für das Suchen von Wegen in großen Graphen entwickelt wurde. Bei der vom Autor entwickelten Methode wird der zu durchsuchende Graph präpariert und eine spezielle Datenstruktur geschaffen, so daß das Suchen von Routen um den Faktor 60 gegenüber des herkömmlichen Dijkstra-Verfahrens beschleunigt werden kann. Dabei geht der Author unter anderem von der Annahme aus, daß die kürzeste Verbindung zwischen zwei Punkten eher über eine Schnellstraße bzw. Autobahn verläuft, als über langsame Straßen; besonders in dem Fall, wenn die beiden Punkte weit voneinander entfernt sind ([Ert98], S. 17). Der Nachteil dieser Methode ist, daß die erwähnte Datenstruktur erschaffen werden muß, um die Vorteile dieses Verfahrens nutzen zu können.

[1] Der Dijkstra-Algorithmus kann ebenfalls die Abstände zwischen je zwei Punkten berechnen, müßte aber modifiziert werden, so daß dessen Komplexität auf $O(v^3)$ bzw. $O(v \cdot e \cdot \ln v)$ steigt.

[2] APSP = All-Pair Shortest Path.

Ich möchte ausdrücklich darauf hinweisen, daß das Problem der hier definierten PersonenLogistik die große Anzahl von Wegen ist, die es *sinnvoll zu reduzieren* gilt, und nicht die Qualität der Suchmethoden. Selbst wenn der effizienteste Suchalgorithmus implementiert wäre (siehe [Ert98]), müßte das System nach wie vor mit der Last der vielen Wege zurechtkommen; insbesondere dann, wenn es um Realzeit-Vermittlung geht, wenn also möglichst viele Anbieter und Nachfrager in sehr kurzer Zeit vermittelt werden müssen.

6.2.4 Vergleich zu kommerziellen Navigationssystemen

Um den hier verwendeten Algorithmus zum Suchen von Routen praxisnah zu testen, wurden diverse Start- und Zielpunkte im Stadtgebiet von Bremen bestimmt (Ausschnitte der Punkte sind in Abbildung 6.7 und 6.8 dargestellt). Diese Punkte dienen zum einen CORONA und zum anderen dem TravelPilot-System von Blaupunkt als Eingabeparameter. Am Ende der Testfahrten wurden die Routen von CORONA sowie die Routen vom TravelPilot auf ihre Distanz hin verglichen (siehe Tabelle 6.1). Die Reisezeit wurde nicht aufgeführt, da die Flüsse auf den Straßen vom TravelPilot *nicht beachtet* werden.

Das Testfahrzeug war ein Mercedes-Benz E240 mit Mercedes-Navigationssystem[3]. Die ausgesuchten Start- und Zielpunkte lagen jeweils in Wohngebieten, in entfernten Stadtteilen, auf beiden Seiten der Weser sowie an entgegengesetzten Punkten von Bremen. Es wurden ebenfalls Punkte außerhalb von Bremen angefahren, die hier allerdings nicht im Vergleich zu CORONA betrachtet werden können, da sich die vektorisierte Straßenkarte des CORONA-Systems nur auf das Stadtgebiet von Bremen und nicht auf das naheliegende Umland beschränkt.

[3]Das Mercedes-Navigationssystem ist mit dem TravelPilot von Blaupunkt vergleichbar, da beide Systeme mit der gleichen Datenbasis arbeiten, der vektorisierten Karte von TeleAtlas, die auch in konvertierter Form von CORONA benutzt wird.

Abbildung 6.7: Ausschnitt aus der vektorisierten Straßenkarte von Bremen mit einem Teil der Teststraßen (eingerahmt).

		Alle Angaben in km	TravelPilot	CORONA	Abweichung
1	Source	BIBLIOTHEKSTRASSE			
	Dest	AGNES-HEINEKEN-STRASSE	15,200	12,998	-2,202
2	Source	MARTINISTRASSE			
	Dest	HEINRICH-PLETT-ALLEE	7,200	7,108	-0,092
3	Source	HEINRICH-PLETT-ALLEE			
	Dest	JOHANN-BORNEMACHER-STRASSE	8,900	9,109	0,209
4	Source	JOHANN-BORNEMACHER-STRASSE			
	Dest	BIBLIOTHEKSTRASSE	8,000	8,776	0,776
5	Source	GERHARD-ROHLFS-STRASSE			
	Dest	AUMUNDER FLUR	0,7	1,612	0,912
6	Source	AUMUNDER FLUR			
	Dest	FURTSTRASSE	2,5	3,554	1,054
7	Source	FURTSTRASSE			
	Dest	VULKANSTRASSE	3,8	5,253	1,453
8	Source	VULKANSTRASSE			
	Dest	KOLONIESTRASSE	8,2	9,871	1,671
9	Source	BIBLIOTHEKSTRASSE	5,5	5,104	-0,396
	Dest	KOPERNIKUSSTRASSE			
		SUMME	60,000	63,385	3,385

Tabelle 6.1: Vergleich von Routen (in km); ermittelt vom TravelPilot und von CORONA.

Als Ergebnis läßt sich festhalten, daß die durchschnittliche Abweichung pro Weg ca. 370 Meter beträgt; die prozentuale Abweichung der längeren Wege von CORONA in Bezug zum Navigationssystem beträgt entsprechend ca. 5,6%. Diese Abweichung und die teilweise drastische Abweichung in Einzelfällen können durch die topographische Lage bzw. durch die Funktionalität beider Systeme erklärt werden: Da beide Systeme die gleiche Datenbasis benutzen, sind in beiden Systemen keine Hausnummern vorhanden; dementsprechend suchen beide Systeme einen Weg zur Zielstraße. Das Navigationssystem beendet seine „Führung“, sobald sich das Fahrzeug in der Zielstraße befindet – unabhängig von deren Länge und dem tatsächlich gesuchten Ziel des Fahrers. Somit kann es passieren, daß man den eigentlichen Weg verlängern muß, um zur gewünschten Adresse zu gelangen – bei langen Straßen können das mehrere Kilometer sein. Ähnlich verhält es sich bei CORONA; hier wird als Zielpunkt eine Kante innerhalb einer Straße definiert. Weichen diese Zielpunkte (Navigationssystem - CORONA) voneinander ab[4], so werden unterschiedlich lange Routen gefunden.

Die folgende Abbildung soll diese Problematik näher erläutern.

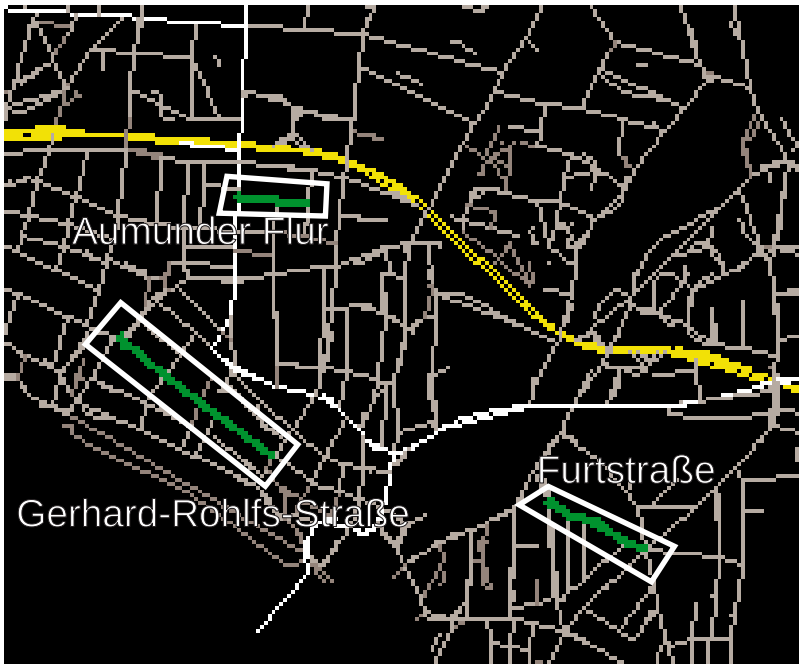


Abbildung 6.8: Ausschnitt aus der vektorisierten Straßenkarte von Bremen (Stadtteil Vegesack) mit einem Teil der Teststraßen (eingerahmt).

Der erste Weg führte von der Gerhard-Rohlfs-Straße zur Aumunder Flur. Das Navigationssystem startet an der linken Seite der Startstraße und erreichte

[4]Da eine Straße in der vektorisierten Karte aus einer Menge von Kanten besteht, kann es leicht vorkommen, daß unterschiedliche Kanten zum Ziel erklärt werden, das Ziel (Straßenname) jedoch korrekt ermittelt wird.

die linke Seite der Aumunder Flur. CORONA hingegen startete an der rechten Seite der Gerhard-Rohlfs-Straße und berechnete dementsprechend einen längeren Weg zum gleichen Zielpunkt[5]. Ähnlich verhält es sich mit der Furtstraße und den anderen Testzielen.

Da das Navigationssystem *nicht* die zu fahrenden Kilometer vor der Fahrt und ebenfalls *nicht* die gefahrenen Kilometer nach der Fahrt angab, mußte die Entfernungsmessung mittels des Tageskilometerzählers erfolgen. Bedenkt man dennoch den oben geschilderten Sachverhalt, kann man behaupten, daß beide Systeme nahezu gleichlange Routen berechnet haben.

Um zu zeigen, daß Abweichungen in der Route auch bei kommerziellen Navigationssystemen normal sind, soll auf die Abbildung 6.9 verwiesen werden.

[5]Die Gerhard-Rohlfs-Straße ist zu einem großen Teil eine Fußgängerzone.

Quelle: ***Auto* Bild**, 23. Januar 1998

Abbildung 6.9: Vergleich von Routen von vier kommerziellen Navigationssystemen.

Wie auf der Abbildung zu erkennen ist, finden die vier verwendeten Systeme vier *unterschiedliche* Wege. Jedes System hat somit andere Gewichtungen bei der Wahl der Route.

Nach diesen Testläufen läßt sich als weiteres Ergebnis festhalten, daß der Algorithmus des verwendeten Navigationssystems autobahnsüchtig bzw schnellstraßensüchtig ist. Wann immer es geht, versucht das System, den Fahrer auf eine entsprechende Straße zu lotsen (hier A28, B74, B75, B6), auch wenn es nicht immer sinnvoll ist. Hierzu der Verlauf eines Testfalls:

> Das Navigationssystem sollte einen Weg von der Bibliothekstraße (Universität) in die Agnes-Heineken-Straße (Arsten) finden[6]. Im Normalfall würde ein Fahrer entweder über die Autobahn (Universität → A28 → A1 → Arsten) fahren oder einen Weg über den Bremer Stadtteil Schwachhausen → Habenhauser Brückenstraße → Arsten wählen; der letztgenannte Weg wurde auch von CORONA berechnet. In diesem Fall aber lotste das Navigationssystem den Wagen über Parkallee → Stern → Hollerallee → Am Dobben → Rembertiring → Hochstraße Breitenweg → B75 → Neuenlander Straße → Autobahnzubringer Arsten → Arsterdamm zum Ziel. Der gefahrene Weg hatte eine Länge von 15,2km — Die von CORONA berechnete, oben beschriebene, Route wurde mit einer Länge von 12,9km angegeben.

6.2.5 Fazit

Der von mir verwendete Suchalgorithmus ist ausreichend schnell und findet in aller Regel kurze Wege. Betrachtet man die Antwortzeit beider Systeme, so läßt sich festhalten, daß CORONA im 1/10 Sekundenbereich die Route berechnet hat, während das Navigationssystem bis zu 15 Sekunden benötigte[7].

Da CORONA ein anderes Suchverfahren als das Navigationssystem verwendet, weichen die Routen von ihrem Verlauf her teilweise voneinander ab (siehe obiges Beispiel und Abbildung 6.9); das Ergebnis – bezogen auf die zurückgelegte Distanz – wird dabei kaum merklich beeinflußt. Da dieser Algorithmus primär dazu dient, Wege von Anbietern und Nachfragern zu Testzwecken des Vermittlungsalgorithmus zu generieren, soll er in dieser Arbeit zunächst verwendet werden.

Zur Zeit arbeite ich an der Implementation eines erweiterten, „intelligenten“ Kurzwegalgorithmus im Sinne des oben beschriebenen Verfahrens, die ich aber aus Zeitgründen nicht vervollständigen konnte. Meine Intention ist es, einen Algorithmus zu entwickeln,

[6] Diese Strecke verläuft von Norden nach Süden quer durch Bremen und über die Weser; siehe Abbildung 6.7 auf Seite 41.

[7] Diese Zeiten können nicht ganz verglichen werden, da CORONA auf einer `SUN Ultra`-Workstation läuft. Daß dennoch Zeitprobleme auftreten und ein schneller Algorithmus nötig ist, um die *Probleme der vielen Wege* in einem PersonenLogistik-System zu lösen, wird im nächsten Abschnitt verdeutlicht.

welcher wirklich sinnvolle Wege findet und nicht nach absoluten Kriterien wie Zeit oder Distanz sucht. Der Ansatz eines solchen Algorithmus ist in CORONA bereits implementiert und kann wahlweise verwendet werden. Dieser Kurzwegalgorithmus ist zur Zeit nach der Methode von D'Esopo ([Sch92]) implementiert und wird im Kapitel 8 getestet und mit dem Algorithmus für *kurze* Wege verglichen.

6.3 Methoden zur Reduktion von Wegen

Die Abbildung 6.10 zeigt die Struktur von möglichen Routen. Jeder Punkt repräsentiert einen Startpunkt (origin) oder Endpunkt (destination) der Teilnehmer. Alle vertikalen Linien beschreiben die individuellen Wege vom Start- zum Endpunkt ohne Vermittlung. Alle anderen Linien visualisieren die Umwege, um einen Nachfrager aufzunehmen. Falls ein Anbieter mehr als einen Nachfrager transportieren kann oder möchte, werden die zusätzlichen Wege durch die gestrichelten Linien dargestellt.

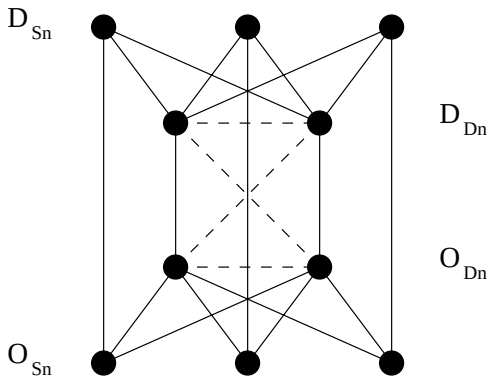


Abbildung 6.10: Struktur aller zu überprüfenden Vermittlungswege bei drei Anbietern und zwei Nachfragern.

In dieser Arbeit sollen jedoch nur die zweier-Fahrten betrachtet werden, d.h. ein Anbieter und ein Nachfrager pro Pkw. In einer typischen Region des Schlanken Verkehrs müssen mehrere hundert bis tausend Vermittlungen pro Tag durchführt werden; unter der Voraussetzung, daß ca. 20% der Pkw-Eigner an dem Öffentlichen Individualverkehr teilnehmen. Um unnötige Berechnungen zu vermeiden und die Routematching-Geschwindigkeit zu erhöhen – ohne Supercomputer anschaffen zu müssen –, müssen Heuristiken entwickelt werden, um passende Paare von Anbietern und Nachfragern zu ermitteln und um die Zeitproblematik zu eliminieren. Die folgenden beiden Abschnitte beschreiben den Aufwand, der ohne Heuristik entstehen würde sowie die Formulierung der im System verwendeten heuristischen Lösung.

6.3.1 Compute All

Der Compute All-Algorithmus berechnet alle möglichen Wege von den Anbietern zu den Nachfragern. Es werden *pro Slot* (δSZ) alle Angebote und Nachfragen untersucht und diejenigen Fahrten vermittelt, die das optimale Ergebnis bezüglich des geforderten δRZ und der Matching-Kriterien liefern. Die Anzahl der Wege, die berechnet werden müssen, sind in Gleichung 6.3 aufgeführt. S ist dabei die Anzahl der Anbieter (Supplier) und D die Anzahl der Nachfrager (Demander).

$$W_{CA} = 2 \cdot S \cdot D + S + D \tag{6.3}$$

Dieser Algorithmus berechnet somit die durchgezogenen Linien aus Abbildung 6.10. In Abbildung 6.11 ist graphisch dargestellt, wieviele Wege es zu berechnen gilt, um eine optimale Vermittlung zu gewährleisten.

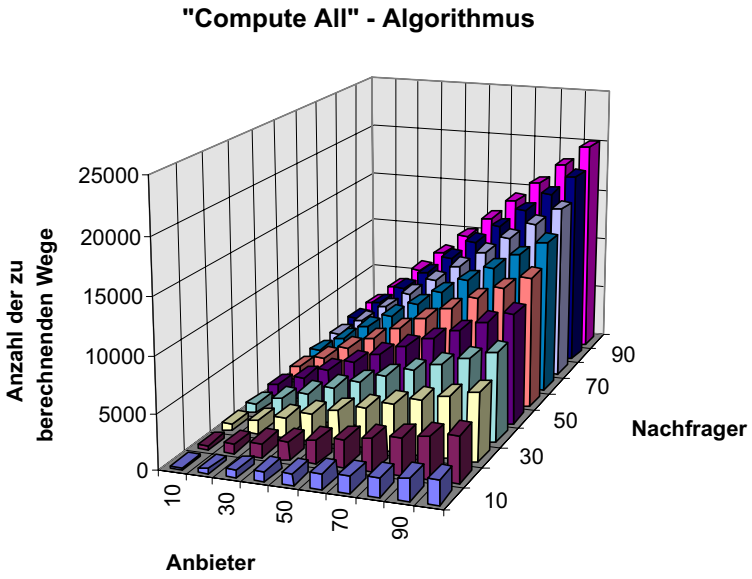


Abbildung 6.11: Aufwand des Compute All-Algorithmus.

Diese Graphik soll lediglich den Aufwand in der Tendenz verdeutlichen. Wenn wir uns vorstellen, daß nur 2.000 Personen (1.000 Anbieter und 1.000 Nachfrager) im Startzeit-Intervall teilnehmen, erhalten wir eine zu berechnende Anzahl von 2.002.000 Wegen. Der Algorithmus liefert zwar das optimale Ergebnis, da er alle nur erdenklichen Kombinationsmöglichkeiten kennt und somit aus dieser Menge die optimalen Verbindungen heraussuchen kann, ist aber vom Zeitaufwand *für die geforderten Verhältnisse völlig inakzeptabel* [8].

[8] Als Beispiel sei hier eine Wegberechnungszeit von 1/10 Sekunde pro Weg aufgeführt. Die Berechnung würde dementsprechend nach ca. 55 Stunden terminieren.

6.3.2 Match and Forget

Um die Zahl der zu berechnenden Wege zu reduzieren, ist die hier beschriebene Heuristik geeignet: Es wird davon ausgegangen, daß von jedem Anbieter aus ein Nachfrager gesucht wird, dessen Matching-Kriterien sich erfüllen lassen. Sobald sich eine passende Vermittlung ergibt, wird sie gesichert und Anbieter sowie Nachfrager werden aus der Liste aller Teilnehmenden herausgenommen. Vermittelte Fahrten werden quasi vergessen[9]. Für den Fall, daß Nachfrager nicht vermittelbar sind, werden diese ebenfalls aus der Liste gestrichen; die Zahl der Anbieter bleibt jedoch erhalten. Unter der Annahme, daß alle Nachfrager vermittelt werden können, beträgt die Zahl der zu berechnenden Wege nunmehr:

$$W_{MaF} = 2 \cdot S \cdot D + S + D - 2 \sum_{i=1}^{n-1} i \text{ für } n = \min\{S, D\} \text{ und } n \geq 2 \qquad (6.4)$$

Dabei ist S bzw. D die Anzahl der Anbieter bzw. Nachfrager. Durch Einsetzen der Summenformel in Gleichung 6.4 erhält man

$$W_{MaF} = 2 \cdot S \cdot D + S + D - 2 \cdot \frac{(\min\{S, D\} - 1)^2 + (\min\{S, D\} - 1)}{2}$$

bzw. durch Kürzen folgende Gleichung:

$$W_{MaF} = 2 \cdot S \cdot D + S + D - ((\min\{S, D\} - 1)^2 + (\min\{S, D\} - 1)) \qquad (6.5)$$

Analog zu Abbildung 6.11 wird der Aufwand an zu berechnenden Wegen dargestellt (Abbildung 6.12). Die Gleichung 6.5 berechnet zunächst alle Wege wie Gleichung 6.3 und subtrahiert anschließend die eingesparten Wege.

$$W_{MaF} = \underbrace{2 \cdot S \cdot D + S + D}_{\text{alle Wege}} - \underbrace{((\min\{S, D\} - 1)^2 + (\min\{S, D\} - 1))}_{\text{Reduktion}}$$

Somit ist $(\min\{S, D\} - 1)^2 + (\min\{S, D\} - 1)$ die maximal zu erwartende Reduktion, die erreicht wird, wenn kein Ausfall bei Vermittlungen entsteht.

Die hier beschriebene Methode kann nun weiter optimiert werden: Wenn man vom Startpunkt des Anbieters aus den Weg zum Nachfrager sucht und anschließend den Weg vom Ziel dieses Nachfragers zum Ziel des Anbieters berechnet und dann die Match and Forget-*Strategie* verfolgt, kann die Anzahl der Wege nochmals reduziert werden. Die Formel zur Ermittlung der minimalen Anzahl zu berechnender Wege ist in Gleichung 6.6, der Aufwand graphisch in Abbildung 6.13 dargestellt.

$$W_{MaF\_opt} = 2 \cdot \min\{S, D\} + S + D \qquad (6.6)$$

[9] Daher auch der Name „Match and Forget“.

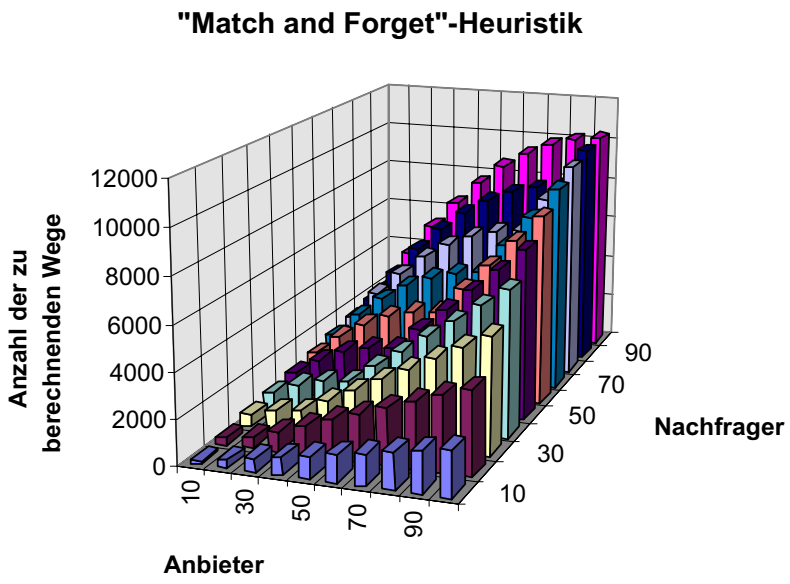


Abbildung 6.12: Rechenaufwand (in Wegen) der nicht-optimierten Match and Forget-Heuristik.

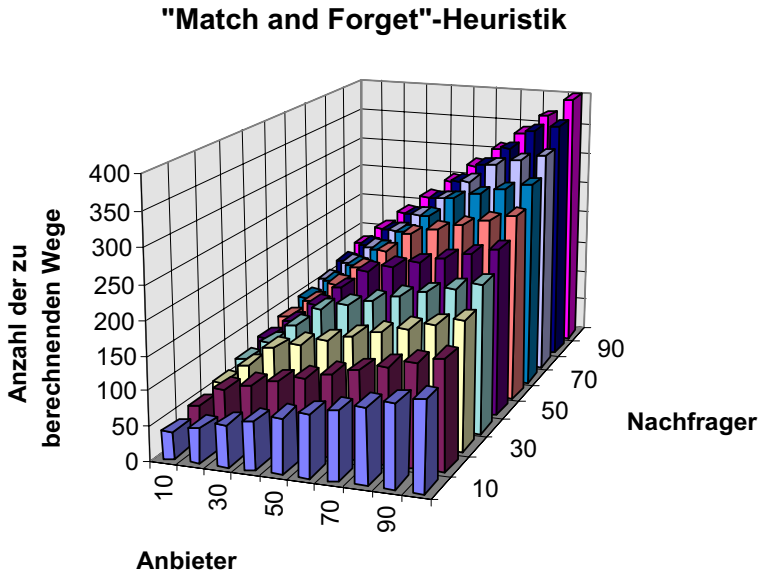


Abbildung 6.13: Rechenaufwand (in Wegen) der optimierten und verwendeten Match and Forget-Heuristik.

Diese optimierte Heuristik wird in dem implementierten System verwendet (Kapitel 7). Die Begründung liegt darin, daß die Wünsche bezüglich der Reisezeit-Toleranz der Anbieter erfüllt sein müssen, um eine Vermittlung möglich zu machen. Es kann somit zwar sein, daß eine andere Vermittlung einen geringeren Umweg ergeben würde, aber aus Zeitgründen diese Berechnungen nicht durchführbar wären. Zudem sei an dieser Stelle auf die Testläufe in Kapitel 8 hingewiesen, aus denen hervorgeht, daß optimale Vermittlungen das Ergebnis – wenn überhaupt – nur minimal verändern und die Verwendung der eben beschriebenen Heuristik das Ergebnis nicht, oder nur kaum merklich, verfälscht.

6.4 Segmentierung der Stadt

Dieses Kapitel beschreibt eine Methode, wie durch Segmentierung einer Stadt der Zeitaufwand zur Wegfindung reduziert werden kann[10]. Eine weitere Reduktion ist notwendig, weil durch alleinige Verwendung der Match and Forget-Heuristik der Berechnungsaufwand zur Wegesuche zwar verringert wird, aber immer noch *unnötige* Wege berechnet werden. Dazu zählen Wege zu Personen, die z.B. am anderen Ende der Stadt wohnen oder dort ihre Ziele haben, so daß eine Überprüfung auf eine mögliche Vermittlung in diesen Fällen wenig Sinn macht.

Im Folgenden wird eine Kreisstadt als Modell herangezogen, um einen einfachen Einblick in die Wirkungsweise dieser Segementierung zu erlangen. Anschließend wird darauf eingegangen, wie die Segmente dieses Modells auf eine reale Stadt übertragen werden können.

6.4.1 Segementierung einer Modellstadt

Es existiert eine kreisförmige Stadt mit einem Durchmesser vom 20 km. Im Zentrum der Stadt liegen die Arbeitsstätten, am Rande liegen die Wohngegenden (Abbildung 6.14). Die Personen sind über das Gebiet gleichmäßig verteilt und fahren jeweils von ihrer Wohnung zur Arbeit im Zentrum (Abbildung 6.14).

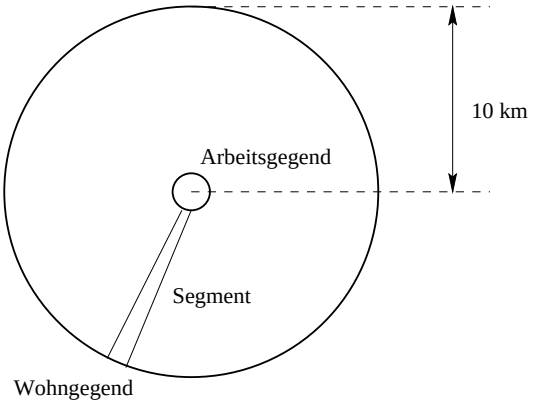


Abbildung 6.14: Darstellung der Kreisstadt. Innen liegen die Arbeitsstätten, außen die Wohngegenden. Die Bevölkerung ist gleichmäßig über die Wohngegend verteilt.

Die unten abgebildete Tabelle zeigt in Abhängigkeit der Reisezeit-Toleranz δRZ die Anzahl der Segmente, die daraus reduzierten Wege sowie die zu erwartenden Rechenzeiten.

Hieraus ist zu erkennen, daß mit zunehmender Reisezeit-Toleranz der Rechenaufwand

[10] Die hier angesprochene Einteilung ist *nicht* äquivalent zur Einteilung, die in Kapitel 6.5.1 beschrieben wird.

Reisezeit-Toleranz in %	**5**	**10**	**15**	**20**	**30**	**50**
Segmentanzahl	126	63	42	31	21	13
Wege pro Segment	1720	6640	14760	26080	57838	159598

Tabelle 6.2: Segmente und die daraus resultierenden Wege in Anhängigkeit der Reisezeit-Toleranz δRZ.

steigt, da ein größeres Gebiet als Einzugsgebiet der zu vermittelnden Fahrten in Frage kommt.

6.4.2 Segmentierung einer realen Stadt

Die im vorherigen Kapitel gezeigte Segmentierung einer Stadt aufgrund der Reisezeit-Toleranz der zu vermittelnden Personen bezog sich auf die dort eingeführte Kreisstadt. Da eine reale Stadt selten kreisrund ist, die Wohngegenden nicht auf der Außenseite des Kreises und die Arbeitsstätten nicht im Zentrum liegen, die Personen nicht gleichmäßig auf die Stadt verteilt sind, und nicht alle die gleiche Reisezeit-Toleranz akzeptieren, liegt der Schluß nahe, daß die aufgeführten Berechnungen für reale Verhältnisse nicht aussagekräftig sind.

Dieser Schluß jedoch ist falsch: Die Kreisstadt dient zum Veranschaulichen der Segmentierungmethodik. Die Berechnungen bezüglich des Gewinns an Rechengeschwindigkeit beziehen sich nicht auf die Form der Stadt oder auf die Richtung der Segmente sondern lediglich auf deren Anzahl[11]. Nun dürfen in einer realen Stadt die Segmente nicht wie in der Kreisstadt gerichtet sein (Abbildung 6.15), da nicht alle Personen in Nord-Süd-Richtung fahren.

[11] Diese ist zwar von δRZ abhängig, jedoch bezieht sich δRZ ebenfalls nicht auf die Stadtform.

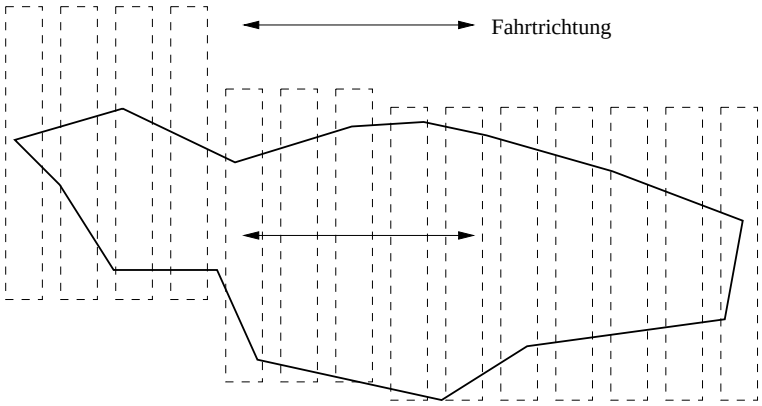


Abbildung 6.15: Hier werden die Segmente (gestrichelte Kästen) in gleicher Richtung plaziert. Da die allgemeine Fahrtrichtung nicht der Richtung der Segmente entspricht, fallen viele Personen aus der Vermittlung heraus.

Die Richtung der Segmente darf auch nicht zur Stadtdehnung parallel angeordnet sein, da ebenfalls nicht davon ausgegangen werden kann, daß alle Personen diese Fahrtrichtung anstreben (Abbildung 6.16).

Sollten die Segmente dennoch so angeordnet sein, dürften wenig Mitfahrten zustande kommen, da viele Personen nicht vermittelt werden können.

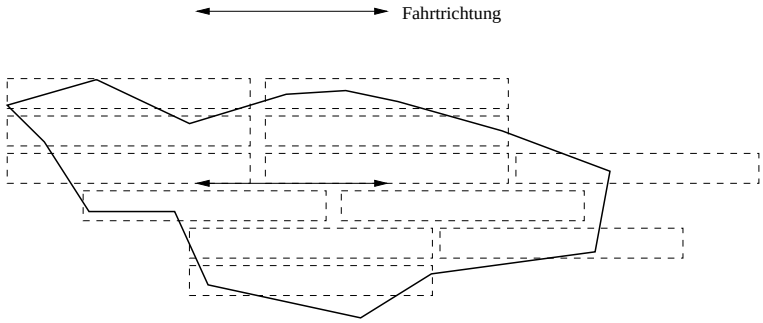


Abbildung 6.16: Hier werden die Segmente (gestrichelte Kästen) ebenfalls in gleicher Richtung sogar zur allgemeinen Fahrtrichtung plaziert, jedoch ist es nach wie vor so, daß Personen nicht vermittelt werden können, falls sie segmentübergreifend fahren wollen.

Um dieses Problem zu lösen, müssen die Segmente analog zu den Hauptverkehrsachsen bzw. Hauptverkehrswegen angeordnet werden, damit die größtmögliche Zahl von Personen eine Chance erhält, vermittelt zu werden, ohne daß die Rechenleistung extrem ansteigt. Diese Segmentierung ist in Abbildung 6.17 ersichtlich.

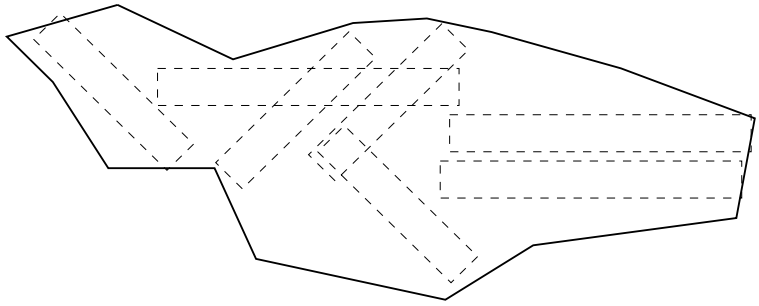


Abbildung 6.17: In dieser Abbildung sind die Segmente (getrichelte Kästen) parallel zu den Hauptverkehrsachsen (-wegen) angeordnet.

Es muß jedoch darauf geachtet werden, daß die Anzahl der Segmente nicht zu gering ist, da sonst kein annehmbarer Zeitgewinn zu erwarten ist. Aufgrund der Tabelle 6.2 wäre eine Zahl von 60 Segmenten empfehlenswert.

Die hier beschriebenen Einteilungen stellen keine zufriedenstellende Lösung dar, da sie von Stadt zu Stadt variieren und somit kein einheitliches System darstellen bzw. sich nicht in einheitliche Algorithmen oder Heuristiken fassen lassen. Der nächste Abschnitt beschäftigt sich mit der Lösung dieses Problems, so daß von einer allgemeingültigen Struktur gesprochen werden kann, die für jede beliebige Stadt gilt.

6.5 Methoden zur Vermittlung gemeinsamer Wege

6.5.1 Einteilung in Gebiete

Eine Lösung zur Einschränkung der zu berechnenden Wege stellt die Einteilung der Stadt in Gebiete dar[12]. Hierbei wird die Stadt sinnvoll unterteilt, und Vermittlungen werden nur innerhalb von Gebieten überprüft (Abbildung 6.18). Personen, die in unterschiedlichen Gebieten wohnen bzw. starten wollen, werden also niemals vermittelt.

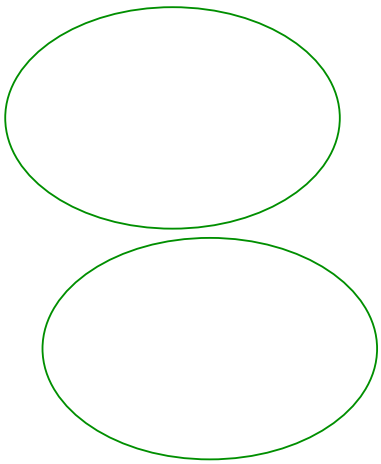


Abbildung 6.18: Einteilung der Stadt in Gebiete. Vermittlungen kommen nur innerhalb von Gebieten zustande.

Nun ist es aber denkbar, daß Personen dicht am Rand dieses Gebiets starten wollen. Insbesondere in Wohngegenden kann es vorkommen, daß benachbarte Straßen unterschiedlichen Gebieten zugeordnet sind, obwohl Personen das gleiche oder ein naheliegendes Ziel haben (Abbildung 6.19).

[12] Die hier beschriebenen Gebiete sind *nicht* identisch mit den in Abschnitt 6.4 behandelten Segmenten. Dieses wird im Verlauf der Beschreibung deutlich werden.

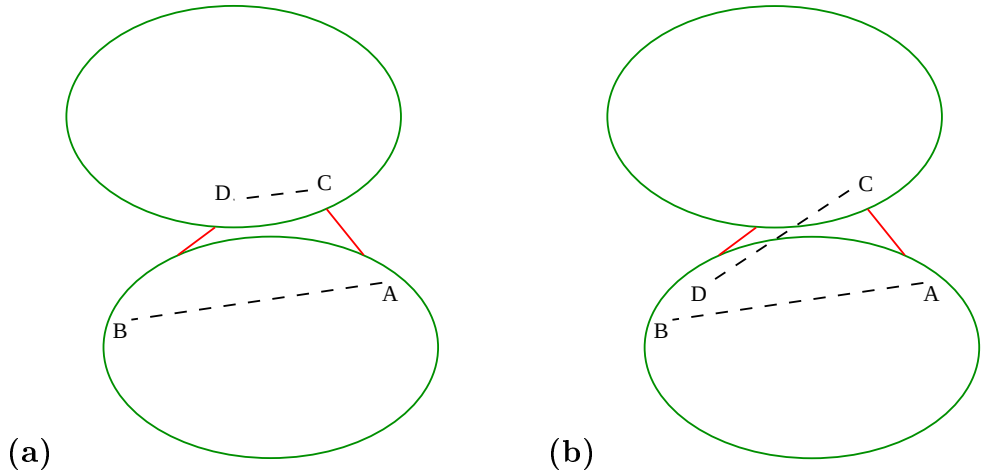


Abbildung 6.19: Personen, die dicht am Gebietsrand starten wollen, könnten ohne weiteres „sinnvoll“ vermittelt werden.

Somit wäre es denkbar, die Gebietsgrenzen zum Rand hin unschärfer erscheinen zu lassen, damit dieser Fall berücksichtigt werden kann. Die in Abbildung 6.19 dargestellte Einteilung besitzt Zugänge zum angrenzenden Gebiet, so daß eine Vermittlung erfolgen kann. Personen, die am Rand des Nachbargebiets starten und ihr Ziel haben (a), oder ein Ziel im Gebiet des Anbieters anvisieren (b), würden verlorengehen, wenn keine Unschärfe existiert.

Allerdings kann es sein, daß nur einer oder ein entfernter Zugang zum Nachbargebiet existiert (Abbildung 6.20). In diesem Fall wäre ein Vermittlungsversuch unsinnig, da der zurückzulegende Weg zu lang ist.

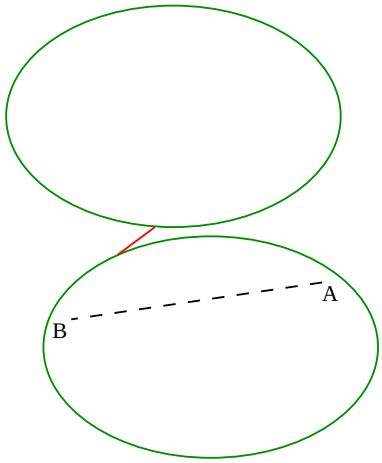


Abbildung 6.20: Bei Vermittlungen über ein Gebiet hinaus müssen die Zugänge berücksichtigt werden.

Durch diese Beispiele wird deutlich, daß eine einfache Einteilung einer Stadt in Gebie-

te zwar ausreicht, um die Anzahl der Vermittlungsmöglichkeiten zu reduzieren, aber sinnvolle Vermittlungen unterdrückt werden. Selbst eine Abschwächung an den Grenzregionen bringt ohne genaue Kenntnis der Zugangsstraßen keinen Vorteil. Und auch wenn diese Zugänge bekannt sind, so existiert doch für jeden Nachfrager und Anbieter ein individuelles δRZ, welches an den unscharfen Grenzen zu überprüfen ist und diese modifiziert.

6.5.2 Individuelles Gebiet

Eine weitere Möglichkeit besteht darin, für jeden Anbieter ein bestimmtes Umfeld zuzuweisen, welches von seiner Reisezeit-Toleranz abhängig ist. Befindet sich der Startpunkt eines Nachfragers innerhalb dieses Gebietes, kann eine Vermittlung erfolgen (Abbildung 6.21).

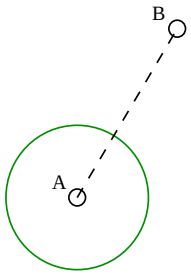


Abbildung 6.21: Vermittlungen können nur stattfinden, wenn der Startpunkt des Nachfragers im Gebiet des Anbieters liegt.

Dieses ist jedoch eine unbefriedigende Lösung, da sie zwar die Menge der zu überprüfenden Personen einschränkt, aber dennoch einige Fälle nicht abdeckt. Existiert z.B. ein Nachfrager innerhalb des Gebietes, wird der Vermittlungsalgorithmus aktiv, jedoch ohne Wissen, ob der Nachfrager in das Zielgebiet des Anbieters will. Ist dieses nicht der Fall, so wurde diese Wegefindung unnützerweise berechnet.

Eine Erweiterung dieser Methode wäre, das Ziel des Anbieters ebenfalls mit einem Umfeld zu umgegeben (Abbildung 6.22). Somit soll nur dann auf eine Vermittlung überprüft werden, wenn der Start- bzw. der Zielpunkt des Mitfahres in einem dieser Gebiete liegt.

Durch diese Methode werden allerdings Nachfrager, die zwischen den Gebieten liegen, nicht berücksichtigt.

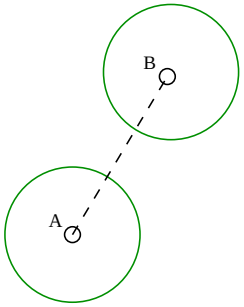


Abbildung 6.22: Erweiterung um ein Gebiet im Umfeld des Ziels des Anbieters.

6.5.3 Matching-in-a-box

Beschreibung

Das *Matching-in-a-box*-Prinzip versucht, alle vorher beschriebenen Probleme zu beseitigen. Bei diesem Prinzip wird eine Box um den Start- und den Zielpunkt des Anbieters gelegt, welche in ihren Ausmaßen der individuellen Reisezeit-Toleranz δRZ entspricht (Abbildung 6.23). Die Vorgehensweise zum Erstellen dieser Box wird in Abschnitt 6.5.3 aufgezeigt.

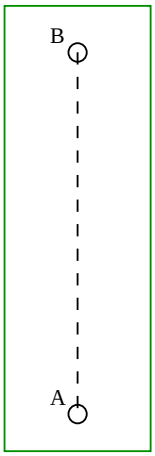


Abbildung 6.23: Das *Matching-in-a-box*-Prinzip. Vermittlungen sind nur dann sinnvoll, wenn der Start- und Endpunkt des Mitfahres innerhalb der Box liegt. Punkt A ist der Startpunkt, Punkt B der Zielpunkt des Anbieters.

Vermittlungen können nur dann zustandekommen, wenn der Start-/Zielpunkt des Nachfragers innerhalb dieser Box liegt.

Nun bleiben noch zwei Problemfälle übrig:

- **Die Richtung des Nachfragers.** Liegen Start- und Zielpunkt des Mitfahres derart, daß der Anbieter quasi zurückfahren muß, um seinen Fahrgast abzusetzen, so ist dieses nicht die gewünschte Lösung (Abbildung 6.24 (a)).
- **Die Topographie innerhalb der Box.** Liegen die Punkte des Nachfragers in der Box, so kann es dennoch sein, daß ein Fluß, eine Bahnlinie etc. die Möglichkeiten, diese Punkte zu erreichen, einschränken (Abbildung 6.24 (b)). In diesem Fall wäre ein Vermittlungsversuch nicht sinnvoll.

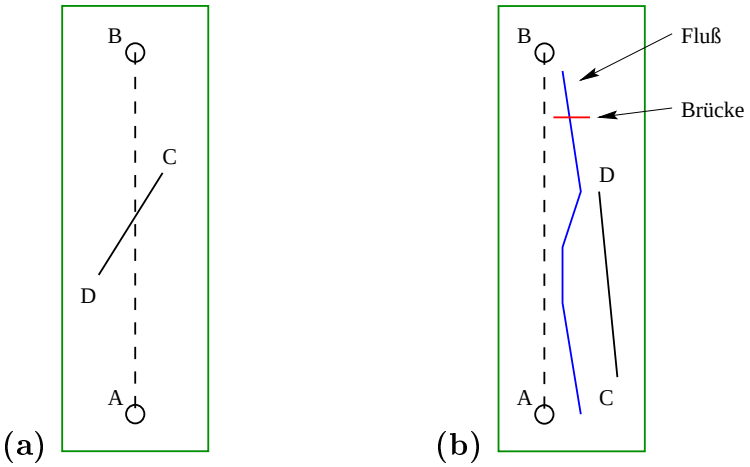


Abbildung 6.24: Zwei Problemfälle der *Matching-in-a-box* Verfahrensweise. In (a) möchte der Nachfrager in die entgegengesetzte Richtung des Anbieters fahren, in (b) existiert nur ein Zugang zum Start-/Zielpunkt des Nachfragers.

Die folgende Beispielrechnung soll aufzeigen, daß diese Methode eine weitere Effizienzsteigerung mit sich bringen kann, als die in Kapitel 6.4 vorgestellte Segmentierung.

Es existieren 5.000 Anbieter und 5.000 Nachfrager. Um jeden Anbieter wird eine *Match-Box* gelegt, innerhalb welcher Personenvermittlungen durchgeführt werden können. Für jeden Anbieter werden die möglichen Wege berechnet:

$$W_{box} = 3 \cdot D + 1 \tag{6.7}$$

Dabei ist W_{box} die Anzahl der zu berechnenden Wege, wenn eine optimale Vermittlung herausgesucht werden soll. Soll nach einer erfüllbaren Vermittlung abgebrochen werden, so verringert sich diese Zahl entsprechend.

Unter der Annahme, es gibt für jede Match-Box (also für jeden Anbieter innerhalb seines Weges) zehn Nachfrager, so ergibt sich für $W_{box} = 31$ Wege. Für alle Anbieter ist dieser Wert nach folgender Formel berechenbar:

$$\begin{aligned} W_{complete} &= S \cdot W_{box} \\ &= S \cdot (3 \cdot D + 1) \end{aligned} \tag{6.8}$$

Die Anzahl aller (maximal) zu berechnenden Wege $W_{complete}$ beträgt somit 155.000. Dieses entspricht einer Rechenzeit von (maximal) 2,6 Stunden bei einer mittleren Wegesuchzeit[13] von 0,06 Sekunden/Weg.

Eine weitere Ersparnis ist erreichbar, wenn zwischen Konstantwegen und Vermittlungswegen unterschieden wird. Unter einem Konstantweg verstehe ich einen Weg, welcher ursprünglich gefahren werden würde, also ohne jegliche Mitnahme bzw. Mitfahrt. Die Bezeichnung Konstantweg bezieht sich darauf, daß ein solcher Weg nur einmal berechnet werden muß, da er lediglich zur Bestimmung von Umwegen und der Fahrzeugkilometerreduktion dient. Entsprechend dazu ist mit dem Begriff Vermittlungsweg ein Weg gemeint, der von der ursprünglichen Route abweicht, so daß ein Anbieter aufgenommen werden kann.

$$\begin{aligned} W &= W_{const} + (S \cdot W_{V\text{-}box}) \\ &= W_{const} + (S \cdot (2 \cdot D)) \end{aligned} \tag{6.9}$$

Durch Verwendung dieser Formel ergeben sich jetzt 110.000 Wege, die nach obigen, ersten Rechenzeiten in etwa 1,8 Stunden abgearbeitet sind.

Diese Werte stellen – wie erwähnt – eine erste Näherung an das Problem dar. Bevor das Programmsystem implementiert wurde, habe ich die hier aufgeführten Berechnungen angestellt, um eine erste Abschätzung der Leistungsfähigkeit zu erhalten. Nachdem ich diese Ergebnisse kritisch betrachtet und für durchführbar befunden habe, wurden diese Heuristiken und Suchalgorithmen oder vielmehr deren Konzepte implementiert. Die folgenden Seiten geben den Algorithmus zum Erstellen der *Match-Box* wieder, wie er von mir konzeptionell entwickelt und implementiert wurde.

[13] Der hier angegebene Wert dient als Beispiel und hat sich im Realbetrieb verändert, siehe Kapitel 8 auf Seite 91.

Algorithmus

Um eine Match-Box zu generieren, wird zunächst die Luftlinie zwischen Start- und Endpunkt des Anbieters bestimmt. Dieses ist der kürzeste Weg, der zwischen `A` und `B` gefahren werden kann (Abbildung 6.25).

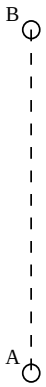


Abbildung 6.25: Erzeugung der Luftlinie zwischen Start- und Endpunkt.

Da normalerweise der Weg, den der Anbieter zurücklegt, von dieser Luftlinie abweicht (Abbildung 6.26), muß die Match-Box diesem Weg angepaßt werden, da ansonsten potentielle Nachfrager von vornherein von einer Vermittlung ausgeschlossen sind.

Abbildung 6.26: Erzeugung des real gefahrenen Weges.

Diese Anpassung geschieht während der Wegebestimmung. Da der Straßenverlauf in der digitalen Straßenkarte durch Knoten und Kanten realisiert ist, wird von jedem Knoten des Weges von `A` nach `B` das Lot auf die Luftlinie gefällt (Abbildung 6.27). Jedes Lot hat dabei eine bestimmte Länge l_n.

Nun wird der größte positive und der kleinste negative Wert von l_n bestimmt. Dabei bedeutet *positiv* die Längenabweichung zur Luftlinie in eine Richtung und *negativ* die

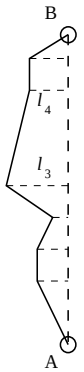


Abbildung 6.27: Berechnung des maximalen Abstandes zwischen Luftlinie und Weg.

Längenabweichung zur Luftlinie in die andere Richtung[14]. Diese beiden Werte bilden die Ausdehnung der Vorstufe zur Match-Box, der *Bounding-Box* (Abbildung 6.28).

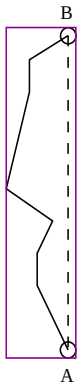


Abbildung 6.28: Erzeugung der *Bounding-Box*.

Da auch vom entferntesten Punkt aus (hier der Knoten an l_3) die Möglichkeit einer Mitnahme gegeben sein muß, wird an dieser Stelle die Reisezeit-Toleranz δRZ zur Breite der Box hinzugenommen. Ebenfalls wird die Box um diesen Toleranzwert in ihrer Höhe verändert, da in diesem Bereich ebenfalls Nachfrager existieren können (Abbildung 6.29 links).

[14]Hier im Beispiel die Abweichung nach *links* und *rechts*.

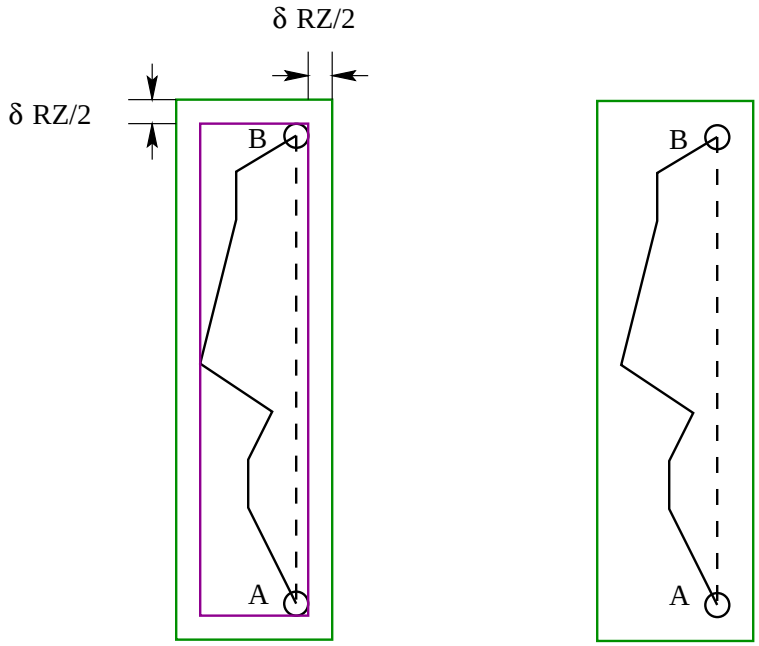


Abbildung 6.29: Erzeugung der *Match-Box* in Abhängigkeit von δRZ.

Die so entstandene Box ist die entscheidende Match-Box (Abbildung 6.29 rechts). Durch dieses Verfahren der Erzeugung einer Match-Box ist gewährleistet, daß kein potentieller Nachfrager übersehen wird.

6.5.4 Der verhaltensorientierte Ansatz — ein Vergleich zur Match-Box

Um zu zeigen, daß die Verwendung von Match-Boxen eine sinnvolle – wenn nicht gar die sinnvollste – Möglichkeit darstellt, um Wege optimal zu reduzieren, ohne potentielle Nachfrager zu verlieren, soll ein menschlicher Weg der Entscheidungsfindung herangezogen werden.

Ein menschlicher Weg der Entscheidung, ob eine Person mitgenommen werden soll, kann durch eine Art *Korridor* beschrieben werden. Die Person „fährt im Geiste den Weg ab“ und überprüft durch Abschätzung, ob eine abzuholende Person in der Nähe wohnt und mitgenommen werden kann (Abbildung 6.30). Das *in der Nähe liegen* ist dabei nichts anderes als die Approximation der Reisezeit-Toleranz des Anbieters.

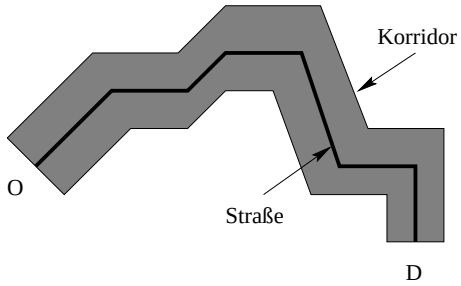


Abbildung 6.30: Darstellung eines *Korridors* von einem Startpunkt (`O`) zu einem Zielpunkt (`D`). Der graue Korridor ist der Bereich, in dem Personen abgeholt und abgesetzt werden können.

Diese Art der Überprüfung wäre zwar am ehesten geeignet, implementiert zu werden, da sie die Realität am genauesten nachbildet, ist jedoch aufgrund der Komplexität der Berechnung zurückzuweisen. Da der Korridor einen Polygonzug darstellt, müßte jeder Start- und Zielpunkt der potentiellen Nachfrager daraufhin überprüft werden, ob dieser innerhalb des Polygons liegt. Da die gesamte Vermittlung sehr zeitaufwendig ist (siehe Abschnitt 8), wird auf diesen Ansatz verzichtet. Bei der Verwendung der *Matching-in-a-box*-Methode ist sichergestellt, daß kein potentieller Nachfrager verlorengeht — es werden lediglich einige wenige Wege vergebens berechnet, da diese nicht in Betracht kommen. Diese scheinbar unnütze Berechnung ist allerdings weit weniger zeitintensiv als eine Polygonüberprüfung. Die Unterschiede sind in Abbildung 6.31 graphisch verdeutlicht.

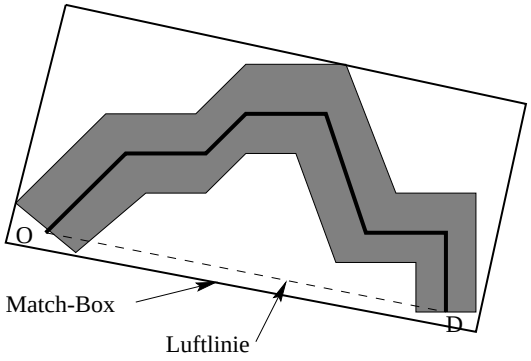


Abbildung 6.31: Darstellung des Korridors und der approximativen Match-Box.

Wie an dieser Darstellung zu erkennen ist, befindet sich der Korridor innerhalb der heuristisch ermittelten Match-Box. Somit werden wiederum keine potentiellen Personen übersehen, da der Korridor eine Teilmenge der Match-Box darstellt und in diesem Korridor bereits alle sinnvoll vermittelbaren Nachfrager enthalten sind. Die Match-Box-Methode ist somit für das Problem der Wegereduzierung eine durchaus sinnvolle Lösung; sie wird im Verlauf dieser Arbeit, zusammen mit der Match and Forget-Heuristik, Verwendung finden.

6.5.5 Durchführung von Vermittlungen

Vermittlungen sollen nun auf drei unterschiedliche Arten durchgeführt werden können:

- Vermittlung: $\Rightarrow$ Pkw (reine Mitnahmen/Mitfahrten)
- Vermittlung: $\Rightarrow$ ÖPNV (reine ÖPNV-Beförderung)
- Vermittlung: $\Rightarrow$ Pkw $\Rightarrow$ ÖPNV (gebrochener Verkehr)

Reine Pkw-Vermittlung:
Die reine Pkw-Vermittlung geschieht nach den vormals beschriebenen Methoden (Match-Box und Match and Forget). Für jeden Mitnehmer werden potentielle Mitfahrer aus der Gesamtpopulation herausgesucht, die Wege von dem aktuellen Mitnehmer zu der Teilmenge der Mitfahrer werden berechnet (Suchalgorithmus) und bezüglich der Matching-Kriterien überprüft (insbesondere die Reisezeit-Toleranz δRZ).

Reine ÖPNV-Vermittlung:
Die ÖPNV-Vermittlung richtet sich nach einem festgelegten, individuellen Radius, in welchem die in Frage kommenden Haltestellen liegen dürfen, und nach der maximalen Wartezeit beim Einsteigen bzw. Umsteigen. Ebenfalls berücksichtigt diese Vermittlungsart den Weg von der Zielhaltestelle zum Ziel des Nachfragers.

Vermittlung im gebrochenen Verkehr:

In der Vermittlungsart gebrochener Verkehr findet eine Art Mischung aus den beiden obigen Vermittlungstaktiken statt. Dabei wird ein potentieller Pkw-Anbieter gesucht (nach den entsprechenden Kriterien) und überprüft, ob dieser den Nachfrager an einer Haltestelle absetzen kann, die von einer Linie bedient wird, die den Nachfrager an dessen Ziel bringt. Als Beispiel sei auf Abbildung 6.32 verwiesen: Es existiert ein Anbieter `Os12`, der zu seinem Ziel `Ds12` fahren möchte. Ein Nachfrager `Od33` möchte nach `Dd33` transportiert werden. Eine reine Pkw-Fahrt ist in diesem Fall ausgeschlossen, da die Reisezeit-Toleranz überschritten werden würde. Da der Anbieter den Nachfrager allerdings an einer Haltestelle `Kulenkampffallee` absetzen und der Nachfrager mit der Linie `28` in die Nähe seines Zieles `Campingplatz` gelangen kann, wird diese Vermittlung von CORONA herausgefunden und vorgeschlagen.

Abbildung 6.32: Beispiel einer Vermittlung im gebrochenen Verkehr durch CORONA.

Bei Vermittlungen wird die Reisezeit-Toleranz des Anbieters und die Startzeit-Toleranz des Nachfragers bzw. Anbieters beachtet. Dadurch, daß nur zweier-Fahrten existieren können, ändert sich die Reisezeit des Nachfragers nicht, da für ihn keine zusätzlichen Umwege entstehen – die Umwege fallen nur für den Anbieter an. Bei einer ÖPNV-Vermittlung oder Vermittlung im gebrochenen Verkehr ist die Reisezeit entsprechend länger. Die Nachfrager haben aber die Möglichkeit, diese Vermittlungsart auszuschließen. Dementsprechend kann eine Reisezeit-Toleranz für den Nachfrager entfallen.

An dieser Stelle enden die Überlegungen und Definitionen zu den verwendeten Algorithmen und Heuristiken. Im anschließenden Kapitel wird das Ergebnis meiner Forschungen und Untersuchungen vorgestellt: Die im Titel erwähnte Teilimplementation des vollautomatischen Vermittlungssystems.

7. Das CORONA-System

7.1 Allgemeines

CORONA ist ein Akronym für Computer Organized Routematching and Navigation und stellt die Teilimplementation dieser Dissertation dar. Im Verlauf dieses Kapitels soll das CORONA-System bezüglich seiner Bedienung, seiner Visualisierung und seiner Fähigkeiten dargestellt werden. Ferner werden diverse Hilfsprogramme aufgeführt, die der Konvertierung bzw. Modifikation der Basisdaten dienen und mit CORONA eine Einheit bilden.

7.1.1 Systemvorraussetzung

CORONA ist in `ANSI-C` geschrieben und mit dem C-Compiler `gcc 2.7.2` kompiliert worden. Für die Oberfläche wurde die `xview 3.2`-Bibliothek sowie die `Xlib`-Bibliothek unter `Solaris 2.5` verwendet. Das System wurde ebenfalls unter `Linux 1.0`, `Linux 1.2.13` sowie `Linux 2.0.x` kompiliert und getestet. Die minimale Bildschirmauflösung beträgt 1024×768 Pixel. Als Hauptspeicherminimum gelten 20MB (`Linux`) bzw. 64MB (`Solaris`). Der verwendete Rechner im Institut für Informatik und Verkehr ist eine Workstation vom Typ `SUN Ultra 1 Creator 3D` (Modell `170E`) mit 128MB Hauptspeicher.

CORONA ist auf jedem `X11`-System kompilier- und lauffähig, wenn die oben genannten Bibliotheken verwendet werden. Die unten aufgeführten Hilfsprogramme sind ebenfalls in `ANSI-C` geschrieben und können ebenso wie das CORONA-System mittels entsprechender Hard- und Software kompiliert werden.

7.1.2 Installation

Um das CORONA-System installieren zu können, ist das Programm-Archiv mittels `unzip` zu entpacken. Der Ordner, in dem sich das Programm befinden soll, kann beliebig definiert werden. Zur Installation bzw. Kompilierung ist lediglich der Aufruf

```
make install
```

zu tätigen, um alle Komponenten lauffähig zu machen. Im aktuellen Verzeichnis befindet sich das reine CORONA-System; im Verzeichnis `convertGDF/` das Konvertierungstool, im Verzeichnis `flow/` das Straßenklassen-Flußtool und das Fahrplangenerierungstool in `timeTable/`. All diese Systeme werden unten näher erläutert.

7.1.3 Konvertierung der vektorisierten Straßenkarte

Für die vektorisierten Straßenkarten gilt im allgemeinen der Formatstandard `GDF`[1]. Da CORONA – wie in Kapitel 5 beschrieben wurde – ein eigenständiges Format benötigt, kann das `GDF`-Format in das `UMF`-Format umgewandelt werden. Dieses geschieht in der ersten Stufe mit dem Programm *convert*, dessen Parameteraufruf wie folgt aussieht:

```
Usage: convert <name> {-7 | -8}
```

Mit `name` ist eine `GDF`-Datei gemeint, die es zu konvertieren gilt. Der Parameter `-7` bzw. `-8` gibt die Anzahl der verwendeten Bits an, die für die Namenskonvertierung verwendet werden sollen: 7-Bit-ASCII (ohne Umlaute) oder 8-Bit-ASCII (mit Umlauten). Das Programm liest anschließend die `GDF`-Daten ein und zeigt die Ergebnisse graphisch an (Abbildung 7.1), wobei die Oberfläche sehr einfach gehalten wurde. Während der Phase des Einlesens werden aktuelle Meldungen ausgegeben. Dieses Protokoll wird auf dem Monitor ausgegeben und sieht wie folgt aus:

```
-> convert bremen
Loading data...
Borders: 8.482598,53.012518 / 8.991670,53.230477
Generating edge relations...88.71 sec.
Generating node relations...95.34 sec.
Generating line attribute relations...318.09 sec.
Generating color relations...464.60 sec.
Generating name relations...14.09 sec.
Updating edge list... 1927 deleted.
Undefined: 157
```

Diese Werte zeigen den Vorgang für die Konvertierung der `GDF`-Daten der Stadt Bremen. Es werden zunächst die kompletten Daten in den Speicher gelesen und die `bounding box` der Daten wird bestimmt. Anschließend werden die Kantenrelationen, die Knotenrelationen, die Straßenklassifikationen, die zur Darstellung benötigten Farbrelationen und die Straßennamenrelationen aus den `GDF`-Daten generiert. Zum Schluß werden nicht benötigte Kanten gelöscht und es wird die Zahl der nicht definierten Kanten bzw. Knoten angegeben. Zu den nicht definierten Objekten gehören beispielsweise Objekte, deren

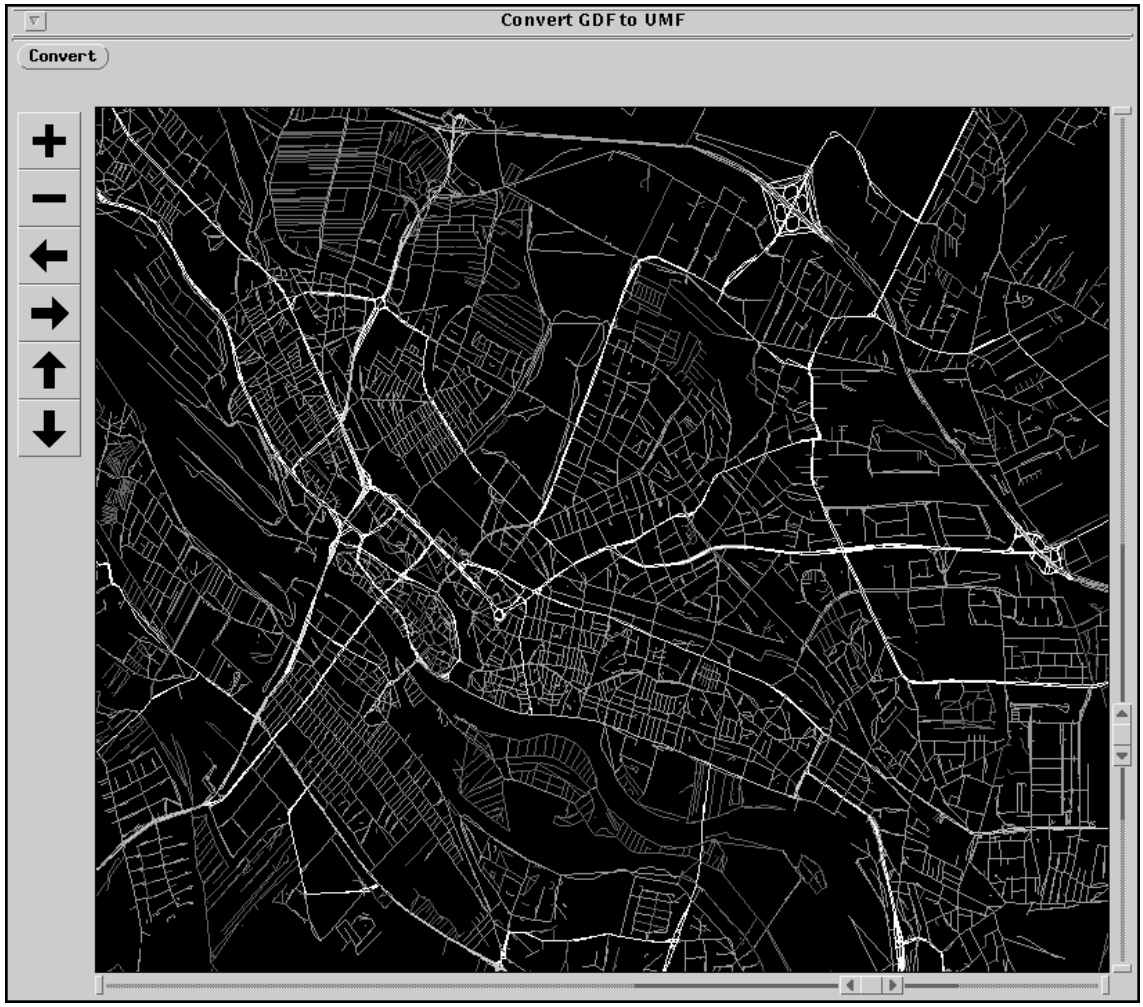


Abbildung 7.1: Die Oberfläche des Konvertierungstools `convert`.

Klassifikation nicht in der `GDF`-Formatbeschreibung enthalten ist. Diese Objekte werden aus der Datenstruktur entfernt.

Anschließend werden die Daten gemäß Abbildung 7.1 dargestellt. Das Programm bietet die Möglichkeit, den dargestellten Ausschnitt mittels Scrollbalken zu variieren, sowie in die Karte hinaus bzw. aus der Karte heraus zu zoomen. Weiterhin kann mittels der Richtungspfeile der Ausschnitt beliebig verändert werden. Am oberen Rand befindet sich ein einzelner Button, welcher die Konvertierungsphase einleitet. Die Daten werden im Verzeichnis `pre-umf/` im `pre-umf`-Format abgespeichert und stehen für den zweiten Schritt der `umf`-Erzeugung bereit. Dieses Vor-Format kann noch nicht für CORONA verwendet werden, da noch keinerlei Flußinformationen existieren. Da diese Informationen nachträglich variierbar sein sollen, existiert ein weiteres Programm, mit dessen Hilfe diese Informationen erstellt werden können.

[1] GDF = Geographical Data File

7.1.4 Einfügen bzw. Modifizieren der Verkehrsflüsse

Um Informationen bezüglich der durchschnittlichen Geschwindigkeit auf den einzelnen Straßenkategorien in Abhängigkeit von der Tageszeit zu erhalten und somit das `umf`-Format zu komplettieren, wurde das Programm *flow* geschrieben, welches folgenden Aufruf benötigt:

```
Usage: flow <name>
```

Der Parameter `name` gibt den Namen der `pre-umf`-Datei an, in welche die Flüsse bzw. Durchschnittsgeschwindigkeiten eingetragen werden sollen. Die Informationen hierzu erhält das `flow`-Programm aus einer speziellen Datei, in welcher die einzelnen Flüsse für die jeweiligen Kategorien definiert sind. Diese Datei ist im Unterverzeichnis `data` enthalten und trägt den festen Namen `flow.dat`. Sie hat eine feste Struktur inne, mittels derer in den jeweiligen Kategorien einer Straße die Durchschnittsgeschwindigkeit pro Stunde angegeben wird. Diese Struktur wird im folgenden aufgeführt — allerdings aus Platzgründen nur für die Kategorie `0`, die Autobahn.

```
0                   Strassenkategorie
120                 Richtung 1 - Geschwindigkeit - 1 Uhr
120                 Richtung 1 - Geschwindigkeit - 2 Uhr
120
120
120
100
100
80
80
80
100
100
100
80
80
80
60
60
80
80
100
100
120                 Richtung 1 - Geschwindigkeit - 23 Uhr
120                 Richtung 1 - Geschwindigkeit - 24 Uhr
-                   Trennung der Richtungen
```

```
120                 Richtung 2 - Geschwindigkeit - 1 Uhr
120                 Richtung 2 - Geschwindigkeit - 2 Uhr
120
120
120
100
100
80
80
80
100
100
100
80
80
80
60
60
80
80
100
100
120                 Richtung 2 - Geschwindigkeit - 23 Uhr
120                 Richtung 2 - Geschwindigkeit - 24 Uhr
---                 Ende Kategorie
```

Nach Durchlauf dieses Programms wird in dem Verzeichnis, in welchem sich die aufgerufene `pre-umf`-Datei befindet unter dem gleichen Namen die `umf`-Datei erzeugt. Die Dateiendung lautet nun ebenfalls `.umf`. Diese Datei kann nun von CORONA verarbeitet werden. Die Vor-Format-Datei wird nicht beeinträchtigt, so daß ohne großen Aufwand ein neuer Fluß in diese Datei eingefügt werden kann. Gibt man anschließend den fertigen `umf`-Dateien unterschiedliche Namen, so ist es möglich, Berechnungen aufgrund verschiedener Geschwindigkeiten durchzuführen.

Die so erstellten Dateien müssen anschließend in das `CORONA`-Verzeichnis[2] `umf` verschoben oder kopiert werden, damit CORONA diese Daten findet.

7.1.5 Die ÖPNV-Struktur und deren Modifikation

Da das CORONA-System auch den ÖPNV bzw. dessen Struktur beinhaltet, muß das ÖPNV-Netz in digitaler Form in das System eingebunden werden. Zu diesem Zweck hat

[2]Das `CORONA`-Verzeichnis ist das Verzeichnis, in dem der Installationsaufruf `make install` erfolgte bzw. in welchem das CORONA-System entpackt wurde. Es ist sinnvoll, diesem Verzeichnis den Namen `Corona` zu geben.

uns die *Bremer Straßenbahn AG* (BSAG) freundlicherweise die Koordinaten aller Haltestellen ihres Unternehmens zur Verfügung gestellt. Es gab allerdings keine Möglichkeit, die Daten des aktuellen Fahrplans aus dem Fahrplansystem der BSAG zu exportieren, so daß das CORONA-System zunächst ohne diese Informationen auskommen mußte. Um dennoch zu einem vernünftigen Ergebnis zu gelangen, habe ich einen Fahrplangenerator entwickelt, mit dessen Hilfe sich aus den analogen Daten des BSAG-Fahrplans eine digitale Repräsentation erstellen läßt. Zunächst soll allerdings erst das ÖPNV-Netz erläutert werden.

Digitales ÖPNV-Netz

Die Daten, die die BSAG mir zur Verfügung gestellt hat, beinhalten die Koordinaten aller Haltestellen im *Gauss-Krüger*-Format. Zusätzlich sind Informationen über die Namen der Haltestellen sowie die Linien, welche diese anfahren, in den Datensätzen enthalten.

Zunächst mußte ich die Koordinaten in geographische Koordinaten umwandeln, da die vektorisierte Straßenkarte (siehe Abschnitt 5) ebenfalls diese Daten verwendet. Da die Gauss-Krüger-Koordinaten einen Bezugsmeridian benötigen, um den *Rechtswert* genau bestimmen zu können und genau diese Angaben fehlte, mußte ich einen Bezugspunkt suchen, an dem eine geographische- und eine Gauss-Krüger-Koordinate zusammenfielen[3]. Anschließend konnten die Koordinaten konvertiert und in das CORONA-System eingebettet werden. Die Darstellung ist in Abbildung (7.2) zu sehen.

[3]Ich habe versuchsweise den 0-Meridian als Bezug verwendet; allerdings ergaben die resultierenden Werte unkorrekte Daten, so daß ein anderer Wert als Bezug eingesetzt werden muß. Dieser war mir allerdings – wie erwähnt – nicht bekannt.

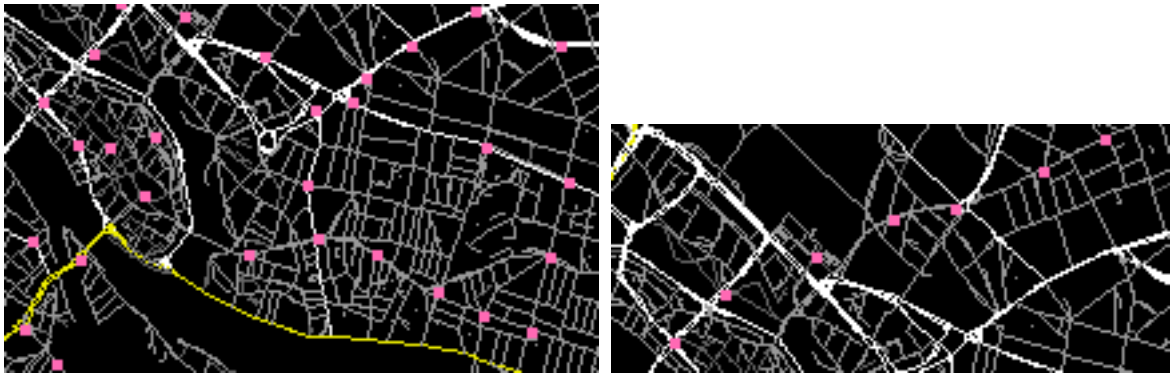


Abbildung 7.2: Ausschnitt aus der Darstellung aller Haltestellen (Kreise) links, sowie einer Linie (Straßenbahn 6) rechts.

Die Datenstruktur der Haltestellen-Datei soll nun kurz erläutert werden:

```
0.0012 0.0018       Deltawerte (x/y)
.
.
.
Universitaet/NW1    Haltestellenname
21 22 23 28 30S     Linien an dieser Haltestelle
8849498 5310508     geographische Koordinaten
```

Die Deltawerte dienen zur Anpassung der Haltestellen an die vektorisierte Straßenkarte, da bei der Konvertierung Rundungsfehler auftreten können. Mittels dieser Werte können die Positionen (optisch) kalibriert werden. Anschließend folgt eine kontinuierliche Liste, die als ersten Wert den Haltestellennamen und als zweiten Wert die geographischen Koordinaten trägt. Die Datei muß im `CORONA`-Unterverzeichnis `StopData/` abgelegt sein. Der Dateiname muß mit dem Namen der Straßenkarte (Abschnitt 7.2) bis auf die Dateiendung identisch sein. Die Endung lautet hierbei `.stop`.

Ein Vorteil dieser separaten Struktur ist der, daß nun weitere Haltestellen ohne große Probleme eingefügt werden können, so daß man in Zusammenhang mit dem generierbaren Fahrplan (nächster Abschnitt) neue Verbindungen im Sinne des Schlanken Verkehrs testen und analysieren kann.

Generieren eines Fahrplans

Nachdem ich erfahren hatte, daß es keinerlei Möglichkeit gibt, den elektronischen Fahrplan als textuelle Datei zu erhalten, stand ich vor diversen Möglichkeiten:

1. Lizenzierung der angebotenen Schnittstellen, so daß CORONA über diese an die Daten gelangen kann (ähnlich wie die *Elektronische Fahrplan-Auskunft* (EFA)).

2. Erstellen des digitalen Fahrplans aus den analogen Daten (abtippen).
3. Erstellen eines Programms, welches die benötigten Daten im geforderten Format erzeugt.

Die ersten beiden Möglichkeiten würden erhebliche Kosten mit sich bringen, zumal zur Lizenzierung der Schnittstellen eine Konvertierung der Daten in ein von mir gefordertes Format hinzukommen würde. Ein „stumpfes Abtippen“ des Fahrplans hätte einer studentischen Hilfskraft übertragen werden können, jedoch wäre diese doch recht langweilige Arbeit spätestens mit dem Erscheinen eines neuen Fahrplans vergebene Mühe. Da in Bremen üblicherweise drei Fahrpläne pro Jahr gelten (Winterfahrplan, Sommerfahrplan, Ferienfahrplan), müssen die so erstellten Fahrpläne auf den aktuellen Stand gebracht werden, da sonst unkorrekte Ergebnisse zu erwarten sind.

Aus diesen Gründen habe ich mich dazu entschieden, einen Fahrplangenerator zu entwickeln, der aus dem jeweils aktuellen Fahrplan die digitale Repräsentation erzeugt. Das Programm `timetable` ist dabei kein Generator im üblichen Sinn, da es keinen optimalen Plan o.ä. erzeugt, sondern dem für jede Linie der *Start-Plan* mit diversen Parametern übergeben wird, und der so alle Pläne für die jeweiligen Haltestellen erzeugt. Der erzeugte Fahrplan ist somit die für die PersonenLogistik benötigte Datenbasis und stellt den jeweils aktuellen, realen Fahrplan dar. Ein Beispiel soll dieses Vorgehen verdeutlichen.

Dem Programm `timetable` liegt eine Parameterdatei zugrunde, welche als Argument dem Programm übergeben wird:

```
Usage: timetable <NAME>
```

`NAME` ist dabei die Datei, welche die Informationen über den jeweiligen Fahrplan beinhaltet. Diese Datei besitzt folgende Struktur:

```
10 2          Anzahl der Haltestellen / Zeitdifferenz zw. Haltestellen
5 3 7 0 15    Start h / Start m / Ende h / Ende m / Zeitdifferenz
7 0 9 0 5
9 0 15 0 10
15 0 17 0 5
17 0 19 0 10
19 0 23 0 15
```

Aus diesen Daten generiert das Programm dann die entsprechende Anzahl an Haltestellen mit dem jeweiligen zeitlichen Versatz. Die oben gezeigte Struktur beschreibt quasi den Fahrplan an der Endstation und erzeugt alle nachfolgenden Zeiten für die einzelnen Haltestellen. Als Beispiel sei ein Ausschnitt der erzeugten Datei aufgeführt:

```
05 03 18 33 48
06 03 18 33 48
```

```
07 00 05 10 15 20 25 30 35 40 45 50 55
08 00 05 10 15 20 25 30 35 40 45 50 55
09 00 10 20 30 40 50
10 00 10 20 30 40 50
11 00 10 20 30 40 50
12 00 10 20 30 40 50
13 00 10 20 30 40 50
14 00 10 20 30 40 50
15 00 05 10 15 20 25 30 35 40 45 50 55
16 00 05 10 15 20 25 30 35 40 45 50 55
17 00 10 20 30 40 50
18 00 10 20 30 40 50
19 00 15 30 45
20 00 15 30 45
21 00 15 30 45
22 00 15 30 45

NEXT_STOP

05 05 20 35 50
06 05 20 35 50
07 02 07 12 17 22 27 32 37 42 47 52 57
08 02 07 12 17 22 27 32 37 42 47 52 57
09 02 12 22 32 42 52
10 02 12 22 32 42 52
11 02 12 22 32 42 52
12 02 12 22 32 42 52
13 02 12 22 32 42 52
14 02 12 22 32 42 52
15 02 07 12 17 22 27 32 37 42 47 52 57
16 02 07 12 17 22 27 32 37 42 47 52 57
17 02 12 22 32 42 52
18 02 12 22 32 42 52
19 02 17 32 47
20 02 17 32 47
21 02 17 32 47
22 02 17 32 47
```

Diese Daten stellen die *ersten beiden* Haltestellenfahrpläne dar. Für den Begriff `NEXT_STOP` ist der jeweilige Name der Haltestelle einzutragen. Mittels dieses Generators ist es möglich, den Fahrplan der BSAG schnell zu generieren oder diesen abzuändern. Somit ist es möglich, im Voraus Berechnungen bezüglich eines modifizierten Fahrplans anzustellen.

Die oben beschriebene Datei muß – analog zu den Haltestellenkoordinaten – im `CORONA`-Unterverzeichnis `StopData/` abgelegt werden. Der Dateiname muß ebenfalls identisch

mit dem Kartennamen (nächster Abschnitt) sein. Die Dateiendung lautet hierbei `.line`.

7.2 Aufruf von CORONA

In den folgenden Abschnitten wird der Begriff Hintergrund-Berechnung eingeführt. Damit ist eine Berechnung gemeint, welche keinerlei graphische Ausgabe erzeugt. Der Sinn liegt darin, daß somit von einem *beliebigen* (VT100-) Terminal aus Berechnungen angestellt werden können. Es ist folglich möglich, von einem gewöhnlichen Windows-PC mittels einer `telnet`-Verbindung eine Berechnung im CORONA-System zu starten. Weiterhin kann so unter Verwendung des `Unix`-Befehls `nohup` (no hang-up) das CORONA-System in einen Hintergrundprozeß versetzt werden, welcher auch *nach dem ausloggen* aus dem `Unix`-System weiterhin aktiv bleibt. Somit können z.B. Berechnungen über Nacht getätigt werden, ohne daß die Workstation durch den Benutzer belegt sein muß.

Das CORONA-System wird mit dem Befehl `corona` gestartet, wobei diverse Parameter notwendig sind. Die Möglichkeiten der Parameterübergabe sind im folgenden aufgeführt:

```
corona <MAP> [ -maponly ]                                                    |
       <MAP> <SUP> <DEM> <ALG> -computeonly                                  |
       <MAP> <SUP> <DEM> <ALG> <P-Car> <P-IM> <P-PT> <T-TABLE> -computeonly  |
       <MAP> <GEN-ParameterFile> -generateonly                               |
       -viewstat [ <STAT> ]
```

Der einfache Befehl zum Starten des Systems ist in der ersten Zeile aufgeführt: `corona <MAP>` startet CORONA und lädt die entsprechende vektorisierte Straßenkarte mit dem Namen `MAP`, die sich im `umf`-Format befinden muß (siehe Abschnitt 7.1.3). Der zusätzliche optionale Parameter `-maponly` dient zur reinen Anzeige der Kartendaten. Hierbei können keine Simulationsläufe getätigt werden. Diese Darstellung eignet sich jedoch besonders zum schnellen Erzeugen von *Clustern* (siehe Abschnitt 7.2.1) oder zur Darstellung des ÖPNV-Netzes.

Der Parameter `-computeonly` dient zum reinen Berechnen im Hintergrund (Abschnitt 7.2.2) und benötigt neben dem Kartennamen die Generatordaten für die Anbieter (`SUP`) und die Nachfrager (`DEM`). Der zu verwendende Algorithmus zum Suchen von Routen kann durch den Parameter `<ALG>` festgelegt werden; 0 steht dabei für die *kürzesten Wege*, 1 für *kurze* Wege (siehe dazu Abschnitt 6.2). Ferner können noch die Prioritäten bei der Vermittlung angegeben werden. Diese Werte liegen für die Mitfahrten (`P-Car`), den gebrochenen Verkehr (`P-IM`) und die reine ÖPNV-Vermittlung (`P-PT`) im ganzzahligen Bereich $[1 \ldots 3]$. Zusätzlich kann ein Flag für die Fahrplanbenutzung angegeben werden; bei 0 wird kein Fahrplan benutzt und ein *Sekundentakt* angenommen, bei 1 wird der Fahrplan – sofern vorhanden – zur Vermittlung herangezogen.

Um das Generieren von Populationen durch CORONA im Hintergrund zu gewährleisten, wurde der Parameteraufruf `-generateonly` entwickelt. Dieser benötigt neben der

Kartendatei eine Steuerdatei, welche dem Generator die entsprechenden Anweisungen mitteilt (Abschnitt 7.2.1).

Um die statistischen Daten, die durch CORONA erzeugt wurden, auch dann visualisieren zu können, wenn nicht das ganze CORONA-System gestartet werden soll, wurde der Parameter `-viewstat` implementiert. Ohne die optionale Angabe `STAT` interpretiert CORONA die aktuelle Statistikdatei `statistics.log`. Wird hingegen ein Name für eine Statistikdatei angegeben, so wird diese zur Interpretation herangezogen. Eine nähere Beschreibung findet sich in Abschnitt 7.3.6 wieder.

7.2.1 Generieren im Hintergrund

Um eine bestimmte Population zu erzeugen und diese simulieren zu können, muß laut Anforderungsdefinition (Kapitel 4) ein Generator implementiert werden. Dieser soll hier nicht näher betrachtet werden, da sich diese Beschreibung mit den Parameteraufrufen beschäftigt. Der Generator ist zudem ausführlicher in Abschnitt 7.3.3 beschrieben.

Um eine Population im Hintergrund, also ohne graphische Ausgabe zu erzeugen, muß der Parameter `-generateonly` verwendet werden, welcher CORONA startet, einer Parameterdatei gemäß die Population generiert, diese abspeichert und das Programm beendet. Die zur Generierung notwendige Datei besitzt folgende Struktur:

```
21 444 444 10 0 0 1 70 GenData/Bremen/ClusterData/bremen4a3w 444_sup 444_dem
21 1333 1333 10 0 0 1 70 GenData/Bremen/ClusterData/bremen4a3w 1333_sup 1333_dem
```

Jeder Parameter einer Zeile ist durch ein Leerzeichen getrennt. Die jeweiligen Werte beschreiben dabei

1. den Initialwert des Pseudozufallszahlengenerators,
2. die Anzahl der Anbieter,
3. die Anzahl der Nachfrager,
4. die Reisezeit-Toleranz δRZ,
5. den prozentualen Anteil an Autolosen,
6. den prozentualen Anteil an ÖPNV-geneigten Mitfahrern,
7. den zu verwendenden Algorithmus (kurzer oder kürzester Weg),
8. den prozentualen Anteil am *Cluster-Hit* (siehe dazu Abschnitt 7.3.2),
9. das Verzeichnis mit den Daten der Cluster (siehe ebenfalls Abschnitt 7.3.2),
10. das Verzeichnis, in welchem die Anbieter-Population abgelegt werden soll und

11. das Verzeichnis, in welchem die Nachfrager-Population abgelegt werden soll.

Die Startzeit-Toleranz δSZ ist zur Zeit konstant auf 10 Minuten gesetzt worden und dementsprechend an dieser Stelle nicht parametrisierbar. Nachdem CORONA gestartet wurde, wird die vektorisierte Straßenkarte eingelesen und die Population generiert; die Ergebnisse werden in den zuletzt spezifizierten Orten abgelegt.

Die Datei kann beliebig viele Steuerzeilen enthalten — somit ist es möglich, mehrere Populationen zu generieren; beispielsweise Populationen mit 10% δRZ und jeweils 100, 200, 300, und 400 Personen.

7.2.2 Berechnen im Hintergrund

Analog zum Generieren von Populationen kann auch die Vermittlung ohne graphische Ausgabe – und somit ebenfalls im Hintergrund – durchgeführt werden. Mittels des Parameters `-computeonly` werden die Kartendaten sowie die (zuvor) generierten Populationsdaten eingelesen. Optional können noch Prioritäten für die Vermittlung, sowie der Fahrplan des ÖPNV angegeben werden. Hierzu sei auf den Abschnitt 7.3.5 hingewiesen, in welchem u.a. die Priorisierung beschrieben wird.

7.3 Visualisierung

Das CORONA-System wurde unter `Unix-X11` unter der Zuhilfenahme der Bibliotheken `Xlib` und `xview` entwickelt. An dieser Stelle sollen jeweils die Bedienelemente und somit ein Teil der Funktionalität erläutert werden.

7.3.1 Die Oberflächen

In Abbildung 7.3 ist die Oberfläche des CORONA-Systems aufgezeigt. Oben befindet sich die Menüleiste sowie Schalter zum Wählen des Maus-Modus, der Darstellung der Straßen, der Darstellung der Match-Boxen und der Zoom-/Richtungs-Tasten. An der linken Seite befinden sich Listen, in welcher die Routen, die Cluster und die ÖPNV-Linien (sofern vorhanden) eingeblendet und ausgewählt werden können. Am rechten und unteren Rand sind zwei Scrollbalken zum Verschieben des Kartenausschnitts angebracht. Der Ausschnitt wird je nach Größe des darstellbaren Gebietes ausgewählt.

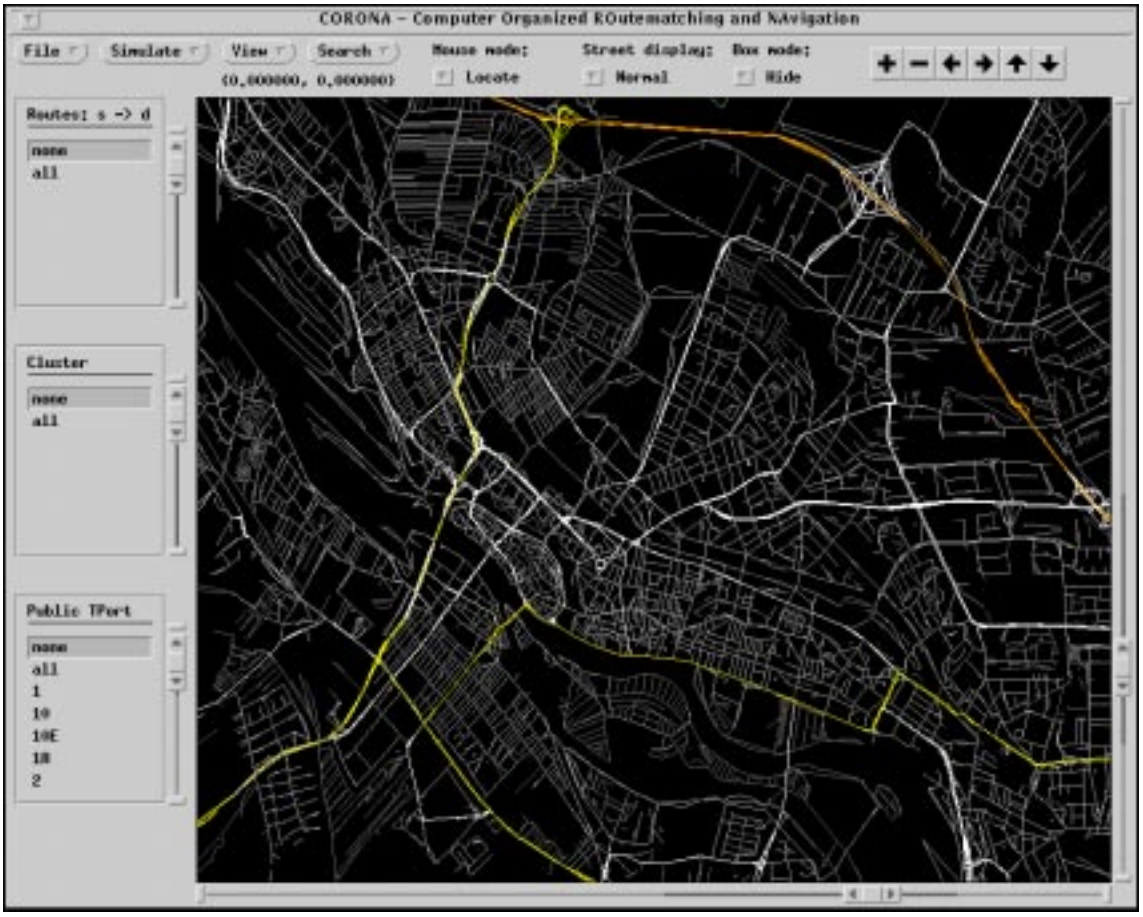


Abbildung 7.3: Die CORONA-Oberfläche mit der Darstellung eines Ausschnitts der Stadt *Bremen*.

Mouse mode: Der Maus-Modus kann drei Zustände annehmen. *Locate*, *Cluster* und *Edge*. Dabei bedeutet

- *Locate* eine Anzeige der aktuellen Mausposition auf dem Bildschirm in geographischen Koordinaten sowie der Anzeige des Straßennamens bei Anwählen einer Kante,
- *Cluster* das Erstellen von Clustern mittels der Maus (Abschnitt 7.3.2) und
- *Edge* die Erzeugung zusätzlicher Kanten zur Erweiterung der Stammdaten (Abschnitt 7.3.4).

Standardmäßig ist dieser Modus auf *Locate* geschaltet, so daß keine unbeabsichtigten Aktionen getätigt werden können.

Street display: Unter diesem Schalter kann die Darstellung der Straßenkategorien variiert werden. Die Option *Normal* (Standard) zeichnet die Straßen mit gleicher

Dicke, während *Dependant* die Haupt- und Bundesstraßen sowie die Autobahnen stärker hervorhebt (Abbildung 7.4).

Box mode: Dieser Schalter kann zwei Zustände annehmen: *Show* und *Hide*. Entsprechend wird bei einer Auswahl aus der Liste *Routes* die korrespondierende Match-Box angezeigt oder nicht. Die Standardeinstellung ist hierbei *Hide*.

Abbildung 7.4: Darstellung der Ergebnisse mit aktiviertem Schalter *Street display: Dependant* (Ausschnitt).

Im ersten Menü *File* können Aktionen zum Laden und Speichern der generierten Populationen und Cluster getätigt werden. In Abbildung 7.5 wurden 200 generierte Anbieter geladen und dargestellt. Die Wege der Anbieter werden farblich anders gekennzeichnet (orange-braun), als die Wege der Nachfrager (blau). Zudem existieren eindeutige textuelle Zuordnungsmerkmale.

Jeder Anbieter und Nachfrager erhält eine eindeutige Identitätsnummer, sowie Kennzeichnungen über den jeweiligen Start- und Zielpunkt. Die Bezeichnung `Os42` bedeutet dabei: `O`rigin `s`upplier No. `42`. Die Teilkennzeichnung `D` steht entsprechend für `D`estination und `d` für `d`emander.

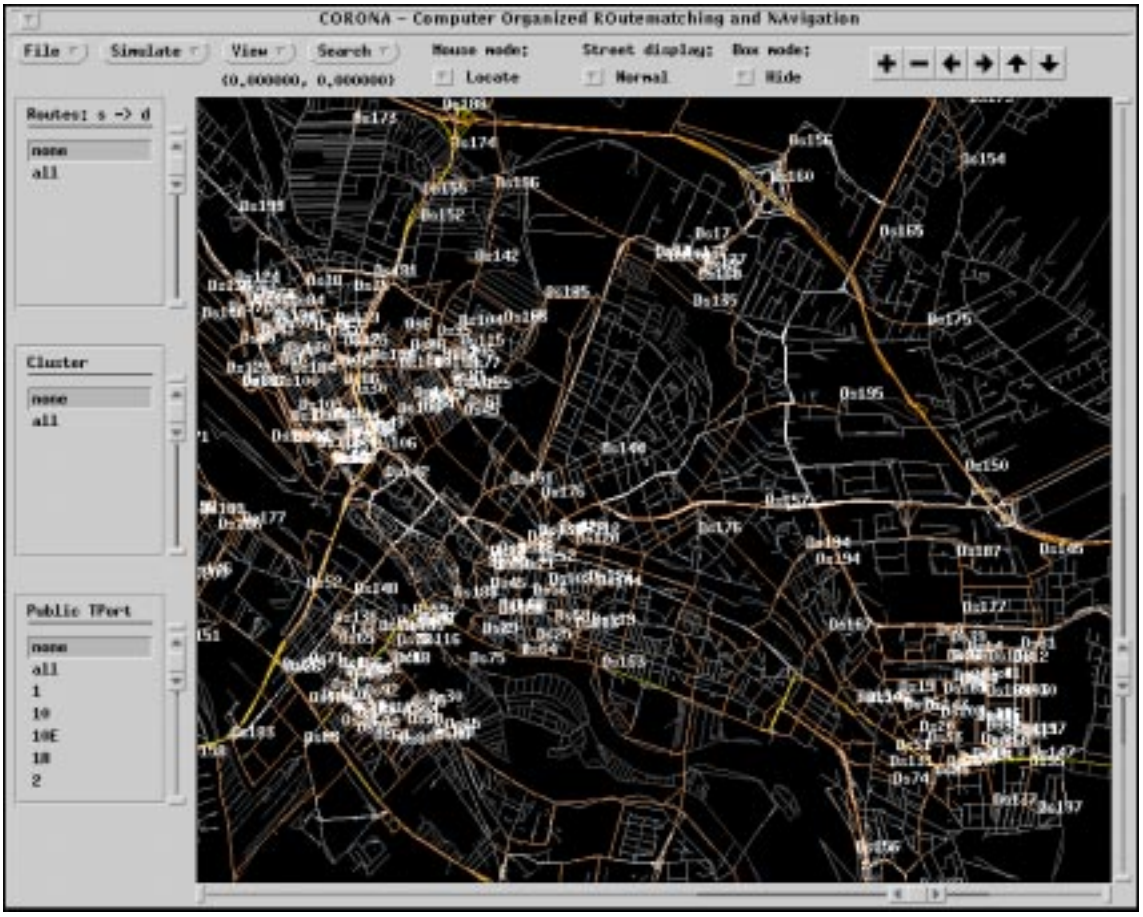


Abbildung 7.5: Darstellung von 200 Anbietern.

7.3.2 Cluster-Generierung

Wie bereits oben beschrieben, kann mittels des Schalters *Mouse mode* die Eingabeoption des Mauszeigers auf *Cluster* gesetzt werden. Somit ist es möglich, graphisch und interaktiv Start- und Ziel-Cluster *beliebig auf der Karte* zu setzen. Weiterhin ist es möglich, die erzeugten Cluster-Daten abzuspeichern und einzulesen.

Das Erzeugen von Clustern vollzieht sich wie folgt: Zunächst muß – wie erwähnt – der *Mouse mode*-Schalter auf *Cluster* gesetzt werden. Anschließend wird mittels der linken Maustaste der Startpunkt gesetzt. Ein weiter Mausklick mit der linken Taste bewirkt das Setzen des Endpunktes[4]. Unmittelbar nach der Definition des Endpunktes wird eine Art „Gummiband-Kasten“ um diese Punkte gelegt und alle darin befindlichen Straßenabschnitte werden rot unterlegt. In der *Cluster*-Liste am linken Rand wird dieser so erzeugte Bereich durchnummeriert angezeigt und mit dem Attribut `end` ergänzt. Dieses `end` bedeutet, daß dieser Cluster ein Ziel-Cluster ist. Jeder Cluster ist bei der Erzeugung

[4] Cluster sind im CORONA-System immer rechteckig.

ein Ziel-Cluster; Start-Cluster können nachträglich aus Ziel-Clustern definiert werden, indem die erwähnte Liste interaktiv modifiziert wird. In der Liste muß im Fall einer Umdefinierung der Cluster das Attribut `end` in `start` umgewandelt werden und der Gummiband-Kasten wechselt die Farbe von rot nach grün (Abbildung 7.6).

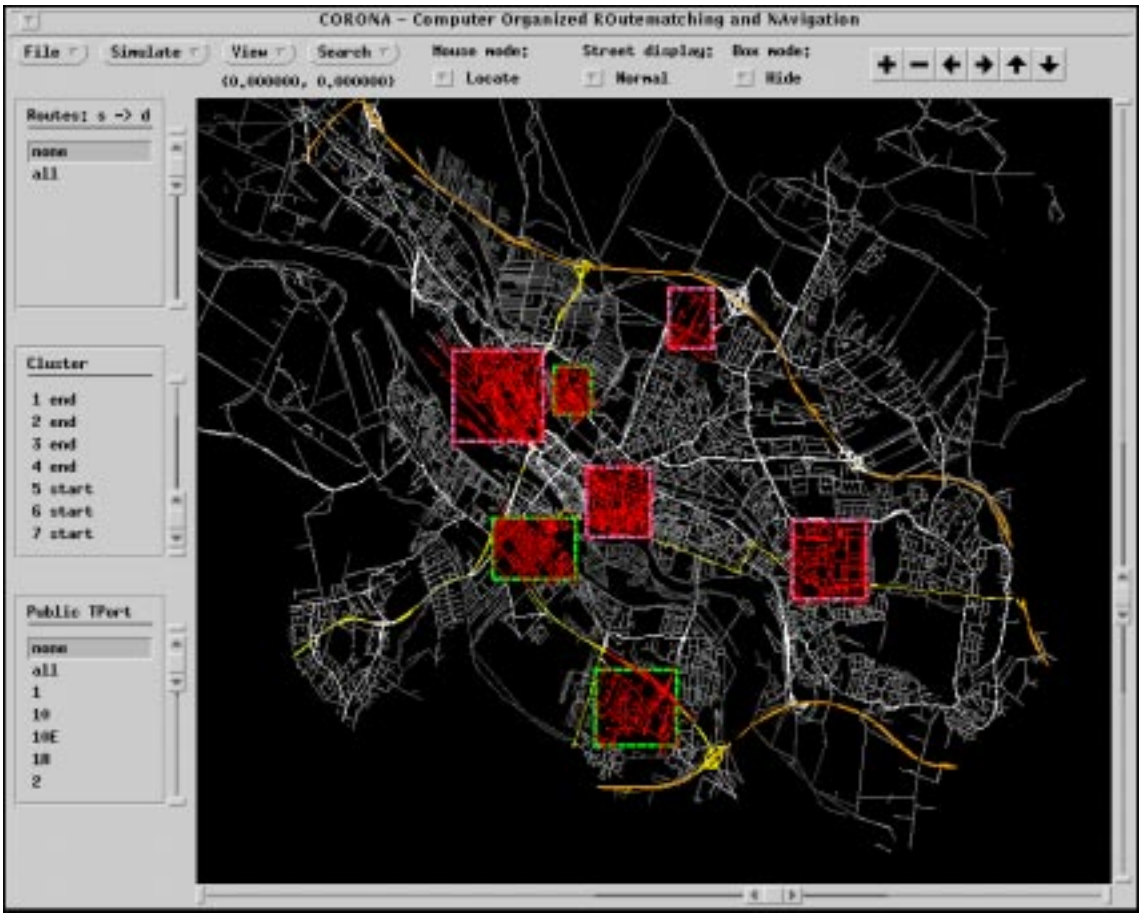


Abbildung 7.6: Darstellung von Start- und Ziel-Clustern (gestrichelte Kästen).

Sollte ein anderes Attribut als die aufgeführten Worte eingegeben werden, so ist die Farbe des Kastens weiß und somit undefiniert.

Das Abspeichern und Einladen von Clustern wird über das *File*-Menü gesteuert, in welchem die entsprechenden Menüpunkte zu finden sind.

7.3.3 Der Generator

Der Generator kann nur dann aufgerufen werden, wenn das CORONA-System ohne zusätzliche Parameter (`corona <MAP>`) aktiviert wurde[5]. Im Menü *Simulate* ist der Punkt *Generate* zu finden, welcher den Generator startet (Abbildung 7.7).

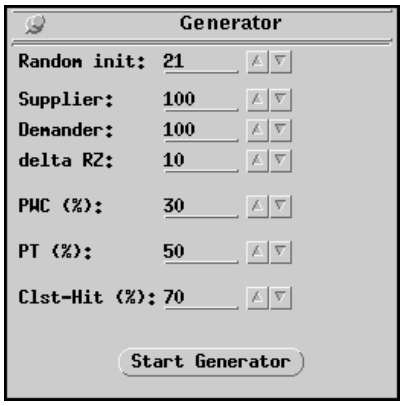


Abbildung 7.7: Parametereingabe des Generators.

In diesem Fenster können nun folgende Einstellungen vorgenommen werden:

Random init: Startwert für den Pseudozufallszahlengenerator.

Supplier: Anzahl der zu generierenden Anbieter.

Demander: Anzahl der zu generierenden Nachfrager.

deltaRZ: Reisezeit-Toleranz δRZ.

PWC: People Without Cars — Autolose; hier kann der prozentuale Anteil der Autolosen bezüglich der Nachfrager angegben werden.

PT: Public Transport — Prozentualer Anteil der Nachfrager, die bereit sind, den ÖPNV mitzubenutzen. Diese Option ist nur dann sichtbar, wenn ein ÖPNV-Netz im CORONA-System existiert.

Clst.-Hit: Cluster-Hit — Prozentualer Anteil der Anbieter und Nachfrager, die innerhalb der Cluster starten und innerhalb der Cluster ihren Zielpunkt haben *müssen*. Der Rest der Population wird gleichmäßig über das Stadtgebiet verteilt; dementsprechend können diese ebenfalls in den Cluster zu finden sein.

[5]Selbstverständlich kann der Generator auch per Kommandozeilenparameter aufgerufen werden (Abschnitt 7.2.1), aber hier wird die Oberfläche beschrieben und innerhalb dieser kann nur bei erfolgter Generierung aller relevanten Graph-Relationen der Generator verwendet werden.

7.3.4 Erzeugen zusätzlicher Kantenzüge

Mittels des *Mouse mode*-Schalters *Edge* können zusätzliche Kanten in den Stammdatensatz der `umf`-Datenbasis eingefügt werden. Zu diesem Zweck wird der Startpunkt der hinzuzufügenden Kante mittels der linken Maustaste gesetzt. Das System sucht den nächsten Knoten zu dieser Position und markiert ihn mit einem roten Punkt. Alle weiteren Aktionen der linken Maustaste setzen einen weiteren Knoten und verbinden diese mit einer Kante. Die neuen Kantendaten werden in einer separaten Datei mit dem gleichen Namen und dem Zusatz `.add` abgelegt. Somit ist gewährleistet, daß die ursprünglichen Daten unverändert bleiben. Beim Einlesen der Karte erkennt CORONA automatisch, ob zusätzliche Informationen gespeichert wurden und bindet diese mit ein. Weiterhin wird jede so eingefügte Kante in einer separaten, interaktiven Liste – die bei Auswahl des Menüpunktes *edge* erscheint – die Nummer der Kante eingetragen. Diese Kante kann so später identifiziert und modifiziert werden. Als Beispiel sei Abbildung 7.8 aufgeführt:

Abbildung 7.8: Darstellung einer hinzugefügten Kante. Der Kreis markiert den Startpunkt und die schräg darüber verlaufende Linie ist die neue Kante.

In diesem Beispiel wurde eine neue Brücke über die Weser generiert und den Stammdaten hinzugefügt. So ist es möglich, die Auswirkungen einer weiteren Straße auf den Vermittlungserfolg zu testen oder neu gebaute Straßen einfach einzufügen, ohne einen neuen Datensatz der vektorisierten Karte erwerben zu müssen.

7.3.5 Vermittlungen

Vermittlungen werden über das Menü *Simulate* und dem dort befindlichen Menüpunkt *Proceed mediation* gestartet. Diese Aktivierung kann nur dann von statten gehen, wenn vorher eine Population von Anbietern und Nachfragern generiert oder eingeladen und

das CORONA-System ohne zusätzlichen Parameter geladen wurde. Zudem ist es möglich anstelle der Nachfrager-Population eine individuelle Nachfrage zu generieren, um dann mit diesem Fall einen Vermittlungsversuch zu starten (Abbildung 7.9). Diese Aktion wird über den Menüpunkt *Manual input* eingeleitet.

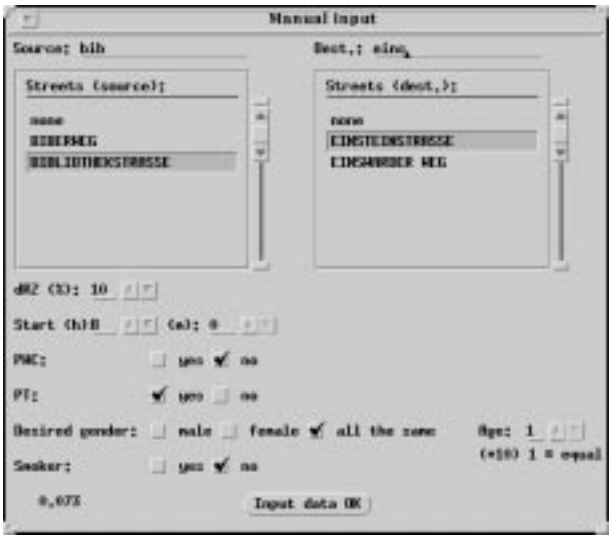


Abbildung 7.9: Die interaktive Eingabemaske.

Analog dazu kann eine reine ÖPNV-Vermittlung stattfinden; sogar in dem Fall, daß das CORONA-System mit dem Parameter `-maponly` gestartet wurde, da für diese Vermittlungsart keine Graphenstruktur der Straßen benötigt werden. Die Eingabemaske ist bis auf einige nicht mehr vorhandene Eingabefelder identisch mit der Maske aus Abbildung 7.9.

Um die Prioritätensteuerung zu variieren, kann mittels des Menüpunktes *Set properties* ein Fenster angezeigt werden, in welchem die aktuelle Priorität der einzelnen Vermittlungsschritte angegeben ist (Abbildung 7.10).

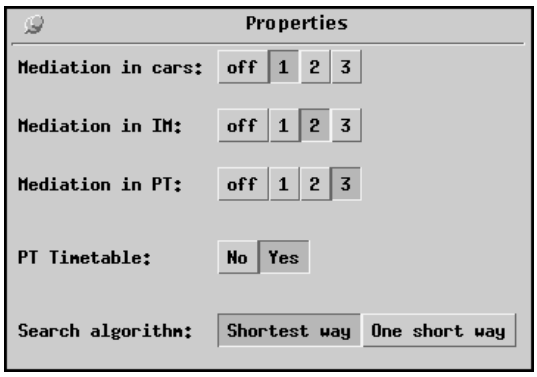


Abbildung 7.10: Eigenschaften der Prioritätensteuerung bei der Vermittlung.

In der `Unix`-Shell werden aktuelle Meldungen bezüglich des Fortschrittes der Vermittlung angezeigt — nach beendigter Vermittlung stehen die statistischen Daten unmittelbar zur Verfügung.

7.3.6 Der Monitor

Um die erhaltenen Daten der Vermittlung visualisieren zu können, kann mittels des Menüs *View* unter anderem der Menüpunkt *Statistics* aufgerufen werden, welcher die Daten textuell und graphisch aufbereitet anzeigt (Abbildung 7.11).

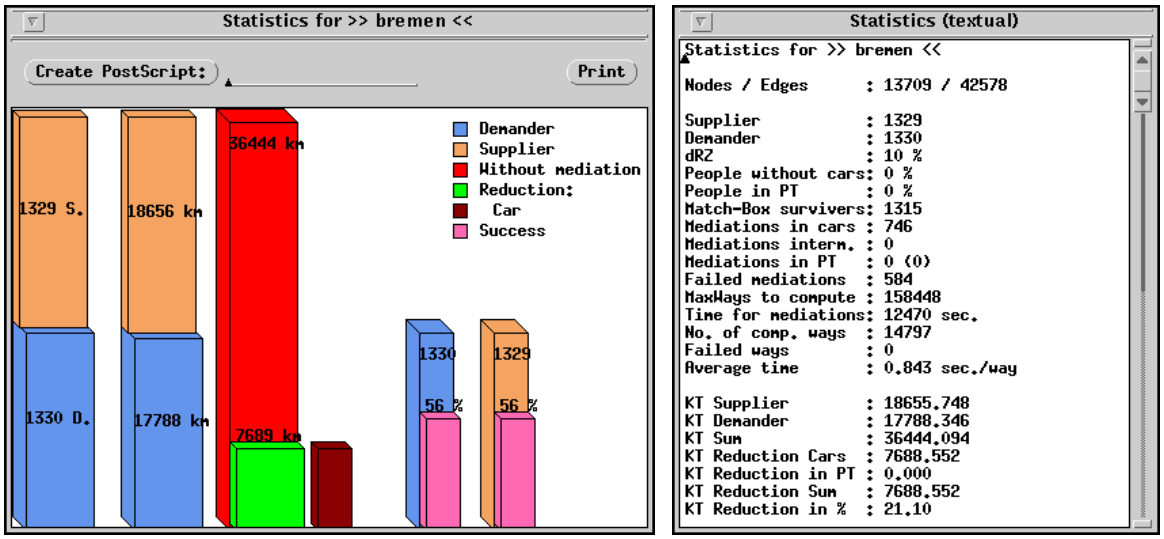


Abbildung 7.11: Statistische Auswertung der gewonnenen Daten. Links die graphische, rechts die textuelle Anzeige.

Die Daten, die im Text-Fenster *für einen Slot* (hier 10 Minuten) angezeigt werden, sind im `CORONA`-Verzeichnis unter dem Namen `statistics.log` abgespeichert. In diesem Zusammenhang sei nochmals darauf hingewiesen, daß diese `log`-Datei in ein beliebiges Verzeichnis kopiert und somit später weiterverwendet werden kann[6]. Diese Verfahrensweise wurde auch für die Darstellung und die Interpretation der Testläufe der Ergebnisse angewendet. Man erhält somit nicht nur die graphische und textuelle Darstellung ohne weitere Berechnungen, sondern auch eine PostScript-Datei mit allen relevanten Informationen.

Ferner beinhaltet die Datei Informationen über die Verteilung der gemeinsam gefahrenen Wege. Dieses gilt nur in dem Fall der Pkw-Beteiligung — auch im gebrochenen Verkehr. Die Werte links neben dem Doppelpunkt geben den Bereich in Kilometern an, rechts wird die Anzahl der Personen für diesen Bereich aufgetragen.

[6]Dieses kann mittels des Kommandozeilenparameters `-viewstat STAT` erreicht werden, wobei `STAT` der Name (inklusive Pfad) der `log`-Datei ist

```
Together
0.0 - 0.5: 1
0.5 - 1.5: 2
1.5 - 2.5: 26
2.5 - 3.5: 51
3.5 - 4.5: 81
4.5 - 5.5: 89
5.5 - 6.5: 58
6.5 - 7.5: 33
7.5 - 8.5: 12
8.5 - 9.5: 14
9.5 - 10.5: 20
10.5 - 11.5: 33
11.5 - 12.5: 69
12.5 - 13.5: 81
13.5 - 14.5: 66
14.5 - 15.5: 55
15.5 - 16.5: 29
16.5 - 17.5: 6
17.5 - 18.5: 11
18.5 - 19.5: 4
19.5 - 20.5: 3
20.5 - 21.5: 0
21.5 - 22.5: 0
22.5 - 23.5: 0
23.5 - 24.5: 1
24.5 - 25.5: 0
25.5 - 26.5: 1
Together : 9.279 km (av)
```

Am Ende dieser Liste wird die durchschnittliche, gemeinsam gefahrene Wegelänge ausgegeben. Nähere Erläuterungen hierzu sind in Kapitel 8 aufgeführt.

7.4 PostScript-Export

Die von CORONA verwendete vektorisierte Straßenkarte und die statistischen Ergebnisse können mittels des implementierten PostScript Exportfilters in Dateien bzw. auf einen PostScript-fähigen Laserdrucker geschrieben werden. Dabei ist zu beachten, daß der PostScript-Interpreter (des Druckers) den `Level 2`-Konventionen entspricht.

7.4.1 Karte

Im Menü *File* findet sich der Menüpunkt *PostScript MAP*. Wird dieser aktiviert, so schreibt das System die Karte in eine Datei mit dem Namen `MAP.PS` in das aktuelle `CORONA`-Verzeichnis. Dabei werden die einzelnen Straßenkategorien durch diverse Grauwerte dargestellt.

7.4.2 Statistische Daten

In Abbildung 7.11 aus Abschnitt 7.3.6 ist in dem linken Fenster (graphische Darstellung) ein Eingabefeld mit einem Button versehen, in welchem ein beliebig definierter Name angegeben werden kann. Unter diesem Namen wird die statistische Darstellung im PostScript-Format abgespeichert und findet sich im Verzeichnis `StatPS` wieder um entsprechend weiterverarbeitet werden zu können. Die detaillierte Erklärung zu diesen Daten wurde bereits in Abschnitt 7.3.6 aufgeführt.

In Kapitel 8 sind diverse statistischen Daten abgebildet, welche durch die hier aufgeführte Funktionalität erzeugt wurden. Der besondere Vorteil dieser Möglichkeit liegt darin, zum einen die Werte direkt auf einem PostScript-Drucker auszudrucken, zum anderen um die Werte in Programmen wie LaTeX oder `FrameMaker` weiterverarbeiten zu können[7].

[7] Wie es auch in Kapitel 8 geschehen ist.

8. Prüfung auf Korrektheit, Robustheit und Leistungsfähigkeit

Im folgenden werden Testdaten für das CORONA-System vorgelegt, um zu zeigen, daß die verwendeten Algorithmen und Heuristiken korrekt arbeiten. Zu diesem Zweck wird zunächst auf die das Ergebnis einstellbaren Variablen eingegangen und anschließend der Testlauf durchgeführt.

8.1 Einstellbare Variablen bei Vermittlungen

Die Berechnungen, die mit CORONA durchgeführt werden können, unterscheiden sich von globalen Macro-Verkehrsmodellen dadurch, daß die hier variierbaren Größen sich stärker auf definierte Szenarien beziehen lassen. Während im LTS-System ([Ehl97]) von J. Ehlich diverse Klassen und Werte modifiziert werden können (Reisezeitbudget, Kilometerleistung, Mitnahmewahrscheinlichkeit, etc.) können in CORONA diese Größen individuell angegeben werden. Weiterhin existieren Möglichkeiten zur geographischen Bestimmung von OD-Matrizen. Es ist möglich, direkt und interaktiv auf dem vektorisierten Stadtplan Start- und Ziel-Cluster zu definieren. Der Generator kann mit einer definierten Wahscheinlichkeit Personen (Anbieter wie Nachfrager) bezüglich der Cluster erzeugen und somit realitätsnahe Bedingungen schaffen. CORONA ermöglicht Aussagen über Auswirkungen eines neuen Wohngebietes, Arbeitsgbietes oder Einkaufszentrums, sowie die Auswirkungen des Schlanken Verkehrs im Allgemeinen.

Da diese Cluster jeweils eine definierte Position und Größe besitzen, gibt es für jede Form unterschiedliche Ergebnisse. Da sich ebenfalls die Wahrscheinlichkeit eines Cluster-Hits ändern kann, modifiziert dieser Wert die Ergebnisse. Wie man hieraus erkennen kann, sind diese Variablen nicht die einzigen Einflußgrößen auf die Simulation und somit den Schlanken Verkehr; der Erfolg ergibt sich aus der Topographie, dem ÖPNV-Netz, der Anzahl von Anbietern und Nachfragern, den indviduellen Start- und Ziel-Clustern, sowie der individuellen Reisezeit- und Startzeit-Toleranz (Tabelle 8.1).

Name	Bedeutung	Herkunft
NUM(A)	Anzahl der Anbieter	Generator
S(A_n)	Startpunkt des Anbieters n	implizit durch HS(C)
Z(A_n)	Zielpunkt des Anbieters n	implizit durch HZ(C)
NUM(N)	Anzahl der Nachfrager	Generator
S(N_m)	Startpunkt des Nachfragers m	implizit durch HS(C)
Z(N_m)	Zielpunkt des Nachfragers m	implizit durch HZ(C)
RZ(A_n)	Reisezeit-Toleranz des Anbieters n (prozentualer Anteil)	Generator
SZ(A_n, N_m)	Startzeit-Toleranz der Anbieter n und Nachfrager m (prozentualer Anteil)	konst. 10 min.
NUM(C)	Anzahl der Cluster	interaktiv / frei wählbar
P(C_i)	Position des Clusters i	interaktiv / frei wählbar
D(C_i)	Dimension des Clusters i	interaktiv / frei wählbar
HS(C)	Cluster-Hit in % für Startpunkte (Anbieter / Nachfrager)	Generator
HZ(C)	Cluster-Hit in % für Zielpunkte (Anbieter / Nachfrager)	Generator
TOP(G)	Topographie eines Graphen G (Stadt) inklusive ÖPNV-Netz und -Fahrpläne	vor dem Start wählbar

Tabelle 8.1: Einstellbare Variablen im CORONA-System.

8.2 Korrektheit

In diesem Abschnitt werden zwei große Testreihen des Gesamtsystems durchgeführt: Ein realitätsnaher und ein stark vereinfachter Test. Der realitätsnahe Test geht dabei von einer fest definierten Verteilung von Start- und Ziel-Clustern in Bremen aus. Die Cluster-Aufteilung wurde wie folgt vorgenommen:

- Start-Cluster:
  1. Neustadt
  2. Findorff
  3. Arsten
- Ziel-Cluster:
  1. Häfen
  2. Centrum
  3. Sebaldsbrück (Mercedes-Benz)

4. Universität (Technologie-Park)

Weiterhin wurden hierbei die Anbieter und Nachfrager mit zwei unterschiedlichen Parametern erzeugt; einmal mit 70% und einmal mit 100% Cluster-Hit[1].

Abbildung 8.1: Verteilung der drei Start- und vier Ziel-Cluster in Bremen.

[1]Die fehlenden 30% im ersten Fall werden gleichmäßig über das Stadtgebiet verteilt.

8.2.1 Test mit realistischen Bedingungen

In der Modell-Stadt Bremen (Stadtgebiet) wird von 270.000 zugelassenen Personenkraftwagen ausgegangen. Als untere Abschätzung wird angenommen, daß insgesamt 40.000 Pkw als Mehrfahrerfahrzeuge (MFF) im Schlanken Verkehr Transportleistung erbringen. Von diesen sollen

- 80% bereit sein, feste Mitfahrgemeinschaften zu bilden, die über einige Wochen halten,
- 20% sollen daran interessiert sein, spontan andere Personen mitzunehmen.

Ferner werden auch hier *nur zweier-Fahrten* betrachtet, da die Bereitschaft, mehr als eine Person mitzunehmen bzw. für diese einen weiteren Umweg zu fahren, äußerst gering ist. Dieses haben auch die amerikanischen Kollegen des *Institute of Transportation Engineers* herausgefunden. Im übrigen werden *keine Autolosen* betrachtet, da diese Personengruppe keine Fahrleistung im MIV erbringt und dementsprechend auch nicht zur einer Reduzierung beiträgt.

Da bekannt ist, daß die Pkw-Fahrten sich in den Morgen- und Nachmittagsstunden häufen, wird in der Simulation zunächst von einer Verteilung ausgegangen, wie sie in Abbildung 8.2 aufgeführt ist. Für feste Mitfahrgemeinschaften soll nur die Zeit von 6.00 bis 18.00 Uhr genutzt werden, d h. also insgesamt 12 Stunden; für spontane Mitfahrten soll eine Zeitspanne von neun Stunden, von 8.00 bis 17.00 Uhr, angenommen werden. Da die festen Mitfahrten sich größtenteils auf Hin- und Rückfahrten beschränken (Berufsverkehr), wird davon ausgegangen, daß in einer Zeitspanne von sechs Stunden alle Teilnehmer vom Start zum Ziel und in der zweiten Tageshälfte wieder zurück fahren. Bei gleichmäßig angenommenen Startzeit-Toleranzen von 10 Minuten ergibt dies also insgesamt 36 Startzeit-Intervalle (Slots) für die festen und 54 Slots für die spontanen Mitfahrten. Im folgenden sollen nur die festen Mitfahrten näher betrachtet werden. Zu diesem Zweck wurde eine sinnvolle Verteilung über die angesprochenen 12 Stunden eines Tages erstellt, die in Tabelle 8.3 zu ersehen ist. Dabei wird eine Symmetrie unterstellt, d.h., daß die Personen, die vormittags fahren, in gleicher Anzahl auch nachmittags den Pkw benutzen wollen. Diese Annahme deckt sich mit der Realität, da die festen Mitfahrten i.A. aus Arbeitnehmern bestehen, welche nach Beendigung ihrer Arbeit nach Hause fahren. Die graphische Darstellung dieser Verteilung findet sich in Abbildung 8.2 wieder, in welcher die Symmetrie deutlich zu ersehen ist.

Tageszeit	6	7	8	9	10	11
Anteile	1	3	3	3	1	1

Tabelle 8.2: Verteilung der Startzeiten im CORONA-System.

Zudem muß untersucht werden, ob der uns zur Verfügung stehende Rechner diese Anzahl von Wünschen bewältigen kann. Mit Hilfe des CORONA-Systems wurden diverse

Tageszeit	6	7	8	9	10	11	12	13	14	15	16	17
Anbieter	444	1333	1333	1333	444	444	444	1333	1333	1333	444	444
Nachfrager	444	1333	1333	1333	444	444	444	1333	1333	1333	444	444

Tabelle 8.3: Umlegung der Verteilung pro Slot (Slotbreite: 10 Minuten).

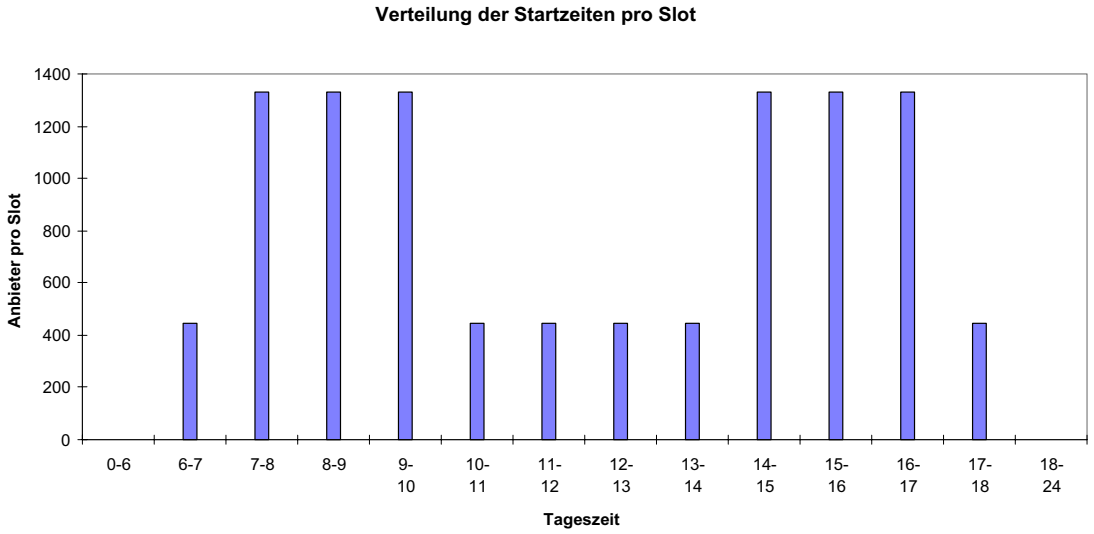


Abbildung 8.2: Zeitliche Verteilung von Anbietern und Nachfragern pro Slot (Slotbreite: 10 Minuten) über den Tag.

Paare von Anbietern und Nachfragern erzeugt und vermittelt; wobei die Berechnungszeit gemessen und in Abbildung 8.3 graphisch aufgetragen wurde.

Die Werte, die in folgenden Graphiken die X-Achse beschreiben, beziehen sich auf gleiche Paare von Anbietern und Nachfragern, d.h., daß der Wert *300* eine Flotte von 300 mitnehmenden Pkw (Anbieter) und 300 Nachfragern darstellt. Der aufgeführte Wert *800* wurde in die Simulation einbezogen, um zu zeigen, ob größere Schwankungen bei stark ansteigenden Werten existieren.

Wie zu erkennen ist, gibt es Zeitprobleme für den realen Einsatz, da innerhalb der geforderten zehn Minuten nur ca. 300 Paare berechnet werden können. Da hier in der Worst-Case-Betrachtung maximal 1.333 Paare berechnet werden müssen, (und zudem feste und spontane Fahrten zusammenfallen können), muß der hierfür relevante Algorithmus verbessert oder eine geeignetere Hardware besorgt werden. Da allerdings vorrangig die Ergebnisse bezüglich Vermittlung und Fahrleistungsreduktion betrachtet werden sollen, kann die Rechenzeit in dieser Simulation bzw. Berechnung vernachlässigt werden.

Ausgehend von obigen Annahmen und Werten habe ich diverse Testläufe mittels CORONA durchgeführt. Die zeitlichen Ergebnisse sind in Abbildung 8.3 dargestellt und sollen

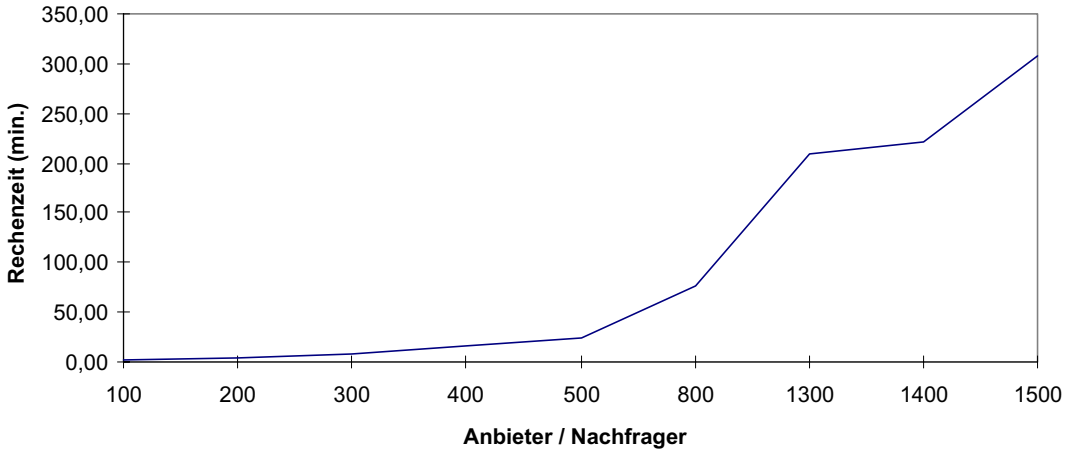


Abbildung 8.3: Graphische Darstellung der realen Rechenzeit von CORONA bei 10% δRZ und einer Slotbreite von 10 Minuten.

hier nicht weiter behandelt werden, da es an dieser Stelle lediglich auf die Ergebnisse der Vermittlung und die daraus resultierenden Fahrleistungsreduktionen ankommt. Weiterhin sollen die hier gewonnenen Daten diskutiert bzw. interpretiert werden, da es darum geht, die Korrektheit des Systems zu überprüfen.

Die Testläufe wurden mit 100, 200, 300, 400, 500 und 800 Anbieter-/Nachfrager-Paaren[2] durchgeführt, wobei die Reisezeit-Toleranz für alle Anbieter δRZ auf 10%, 20% bzw. 30% gesetzt wurde. Die Anbieter und Nachfrager, die hier berechnet wurden, stellen quasi das aktuelle Potential innerhalb eines Slots dar. Wie oben bereits erwähnt, kann die teilnehmende Bevölkerung auf Slot-Größe reduziert werden.

In Tabelle 8.4 sind zunächst die Vermittlungsquoten aufgeführt worden. Es müssen bei höheren Reisezeit-Toleranzen höhere Vermittlungsquoten ermittelt werden, da eine Erhöhung dieser Toleranz einen größeren Umweg erlaubt und somit mehr Personen vermittelt werden müßten.

Anbieter / Nachfrager	**100**	**200**	**300**	**400**	**500**	**800**
δRZ 10%	48	40	45	45	48	54
δRZ 20%	59	53	61	58	63	65
δRZ 30%	63	63	69	65	69	72

Tabelle 8.4: Vermittlungsquote in Prozent (gerundete Werte, Slotbreite: 10 Minuten).

[2] *Paare* bedeutet, daß es jeweils n Anbieter und n Nachfrager gibt.

Wie aus diesen Werten zu ersehen ist, erhalten wir in der Tat mit steigender Reisezeit-Toleranz höhere Quoten; das CORONA-System hat in diesem Fall die zu erwartenden Ergebnisse bestätigt. Die Abbildung 8.4 verdeutlicht die Ergebnisse bildhaft.

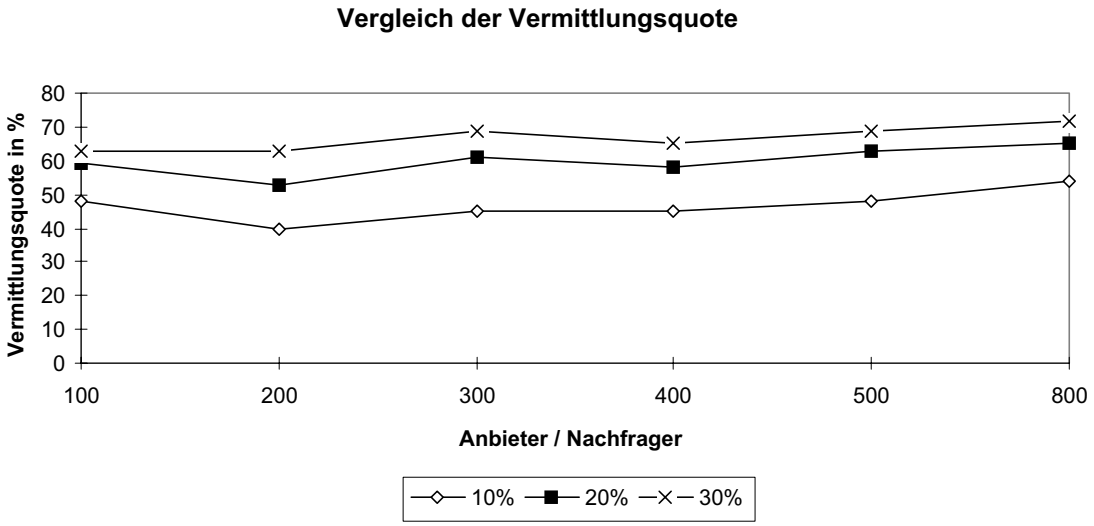


Abbildung 8.4: Vermittlungsquote für 70% Cluster-Hit bei unterschiedlicher Reisezeit-Toleranz (Slotbreite: 10 Minuten).

Wenn man sich den Fahrleistungsreduktionen zuwendet, muß man davon ausgehen, daß die Reduktion mit zunehmender Personenzahl leicht ansteigt.

Anbieter / Nachfrager	**100**	**200**	**300**	**400**	**500**	**800**
δRZ 10%	13	15	16	16	16	20
δRZ 20%	17	16	19	19	19	22
δRZ 30%	16	19	19	21	19	22

Tabelle 8.5: Fahrleistungsreduktion in Prozent (gerundete Werte, Slotbreite: 10 Minuten).

Auch diese Testläufe bestätigen die zu erwartenden Ergebnisse: Die Reduktion steigt stets an bzw. bleibt konstant (Tabelle 8.5 und Abbildung 8.5). Daß in zwei Fällen die Reduktion leicht abnimmt, liegt daran, daß einerseits gerundete Werte verwendet wurden und daß andererseits einige Personen „leicht unglücklich“ wohnen, d.h. sie erbringen zwar eine Fahrleistungsreduktion, die aber weitaus geringer ist, als der zu fahrende Umweg.

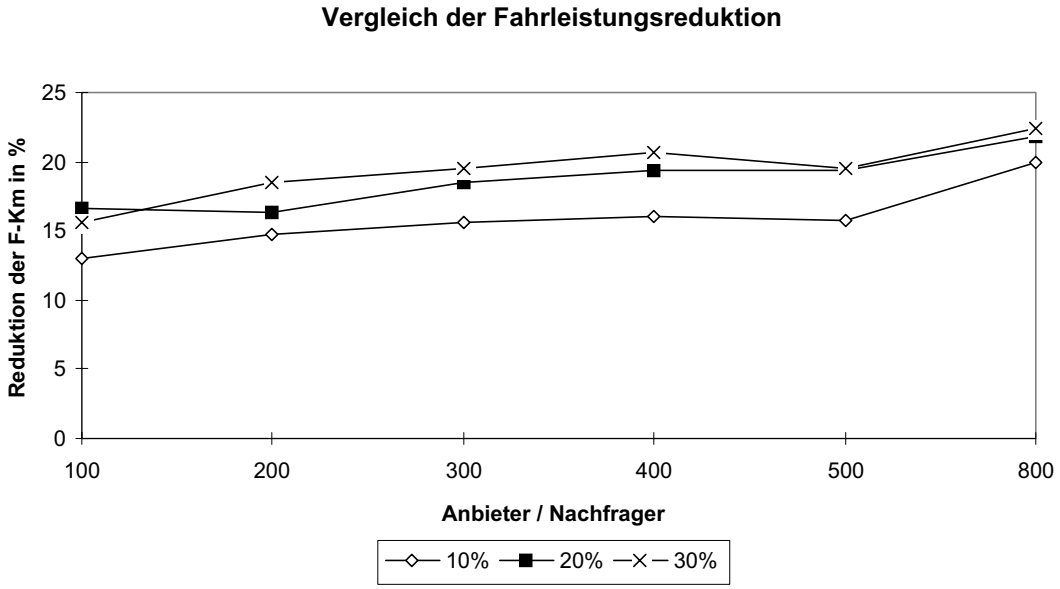


Abbildung 8.5: Fahrleistungsreduktion für 70% Cluster-Hit bei unterschiedlicher Reisezeit-Toleranz (Slotbreite: 10 Minuten).

Nun stellt sich berechtigterweise die Frage, warum diese Werte weit unter dem erhofften Wert liegen. Bildet man gemäß Verteilung aus Abbildung 8.2 die Summe der reduzierten Fahrzeugkilometer eines Tages, so erhalten wir ca. 355.400 km Reduktion in der ÖIV-Flotte. Dieser Wert basiert auf Simulationsläufen, deren Ergebnisse in den Abbildungen 8.6 und 8.7 aufgeführt sind. Geht man ferner davon aus, daß in Bremen jeder Pkw im Schnitt 30 km innerhalb dieser Zeit fährt[3], so erhält man 8.100.000 km. Die Reduktion innerhalb der Gesamtflotte beträgt somit etwas mehr als 4%. Dieser Wert liegt weit unter den gewünschten Zahlen.

Daß die Werte der Fahrzeugkilometerreduktion und der Vermittlungsquote diese unerwarteten Ergebnisse liefern, soll im nächsten Testlauf erklärt werden.

[3]Ich gehe hier von der Annahme aus, daß das konstante Reisezeitbudget 60 Minuten und die durchschnittliche Geschwindigkeit in Bremen 30 km/h beträgt.

CORONA
Statistics

Statistics for >> bremen <<		
Supplier	:	443
Demander	:	444
dRZ	:	10 %
People without cars	:	0 %
People in public transport	:	0 %
Match-Box survivers	:	433
Mediations in cars	:	229
Mediations intermodal	:	0
Mediations in public transport	:	0
Failed mediations	:	215
Maximal ways to compute	:	20450
Time for mediations	:	1487 sec.
Number of computed ways	:	2879
Failed ways	:	0
Average time	:	0.516 sec./way
KT Supplier	:	6499 km
KT Demander	:	6225 km
KT Sum	:	12725 km
KT Reduction Cars	:	2185 km
KT Reduction in PT	:	0 km
KT Reduction Sum	:	2185 km
KT Reduction (fleet)	:	17.18 %

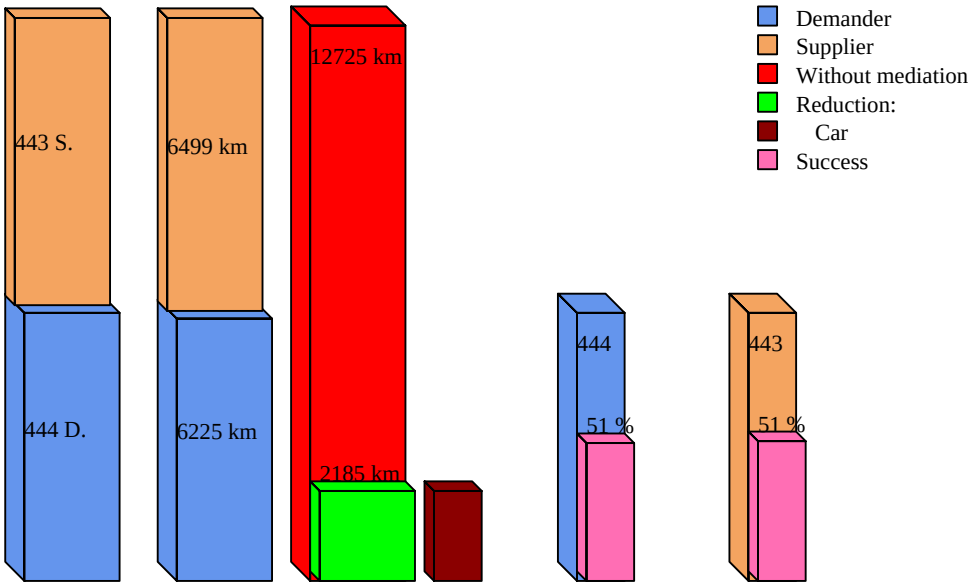
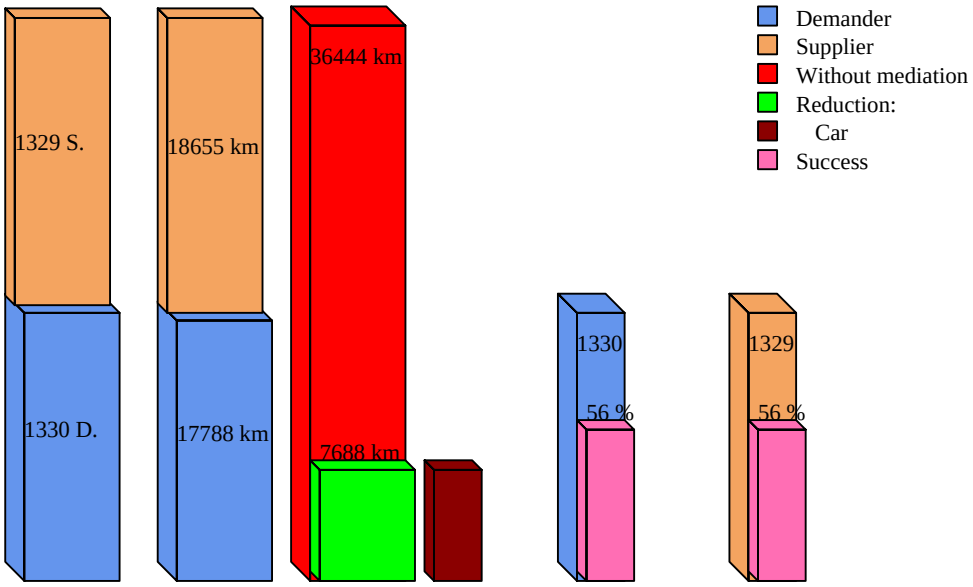


Abbildung 8.6: Durch CORONA ermittelte, statistische Werte für 888 Personen/Slot.

CORONA
Statistics

Statistics for >> bremen <<		
Supplier	:	1329
Demander	:	1330
dRZ	:	10 %
People without cars	:	0 %
People in public transport	:	0 %
Match-Box survivers	:	1315
Mediations in cars	:	746
Mediations intermodal	:	0
Mediations in public transport	:	0
Failed mediations	:	584
Maximal ways to compute	:	158448
Time for mediations	:	12596 sec.
Number of computed ways	:	14797
Failed ways	:	0
Average time	:	0.851 sec./way
KT Supplier	:	18655 km
KT Demander	:	17788 km
KT Sum	:	36444 km
KT Reduction Cars	:	7688 km
KT Reduction in PT	:	0 km
KT Reduction Sum	:	7688 km
KT Reduction (fleet)	:	21.10 %



Institute of Transport and Computer Sciences (IIuV), University of Bremen, Dept. 3, D-28359 BREMEN - Germany

Abbildung 8.7: Durch CORONA ermittelte, statistische Werte für 2.666 Personen/Slot.

8.2.2 Test mit vereinfachten Bedingungen

Analyse des vorherigen Tests

Die von CORONA ermittelten Werte lassen den Schluß zu, daß die Verteilung der durchschnittlichen, gemeinsam zurückgelegten Wegelängen zu kleineren Werten verschoben ist. Dazu wird die Abbildung 8.8 betrachtet. Die gestrichelte Linie zeigt die normale Verteilung von Wegelängen innerhalb einer Stadt an. Die CORONA-Ergebnisse lassen sich dadurch erklären, daß eine Verschiebung dieser Kurve nach links erfolgte (durchgezogene Linie). Wenn nämlich die durchschnittliche Wegelänge, die nach erfolgter Vermittlung gefahren wird, geringer ist, als die ursprüngliche Länge, dann werden weniger Kilometer eingespart, als normalerweise erwartet werden würde.

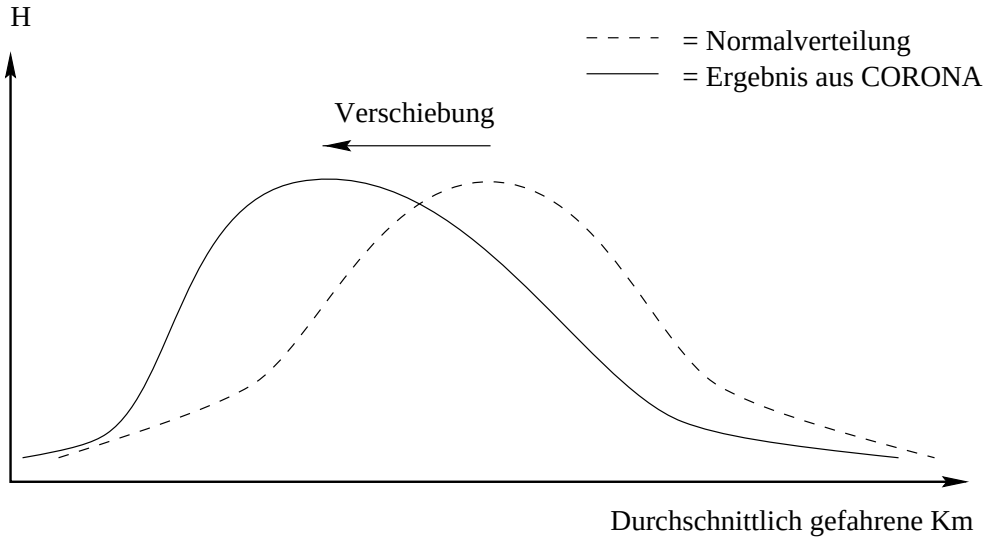


Abbildung 8.8: Verteilungshäufigkeiten der durchschnittlich gefahrenen Kilometer.

Die Abbildung 8.9 zeigt die Umwegeverteilung der vermittelten Anbieter für die Slots – alle mit einer Breite von 10 Minuten – mit jeweils 444 Anbietern und Nachfragern. Bei dieser Konstellation beträgt die durchschnittliche Wegelänge der Anbieter 14,6 km; die durchschnittliche Wegelänge der Nachfrager vor der Vermittlung 14 km [4]. Die zu erkennenden Umwege häufen sich im Meßintervall bis 1,5 km. Dieses Ergebnis bestätigt die Erwartungen, da für die Berechnungen eine Toleranz von 10% angesetzt war und die Umwege somit nicht größer als 1,46 km sein dürfen.

Die Verteilung der anschließend gemeinsam gefahrenen Wege ist in Abbildung 8.10 zu sehen. Diese Graphik bestätigt die Behauptung aus Abbildung 8.8: Das Maximum der Verteilungskurve ist nach links – zum Ursprung hin – verschoben. Die durchschnittliche gemeinsame Wegelänge beträgt im Fall mit 444 Paaren/Slot 9,2 km, die durchschnittliche

[4] Es soll daran erinnert werden, daß in diesem Beispiel keine Autolosen existieren. Dementsprechend haben alle Nachfrager einen Pkw und würden mit diesem auch fahren.

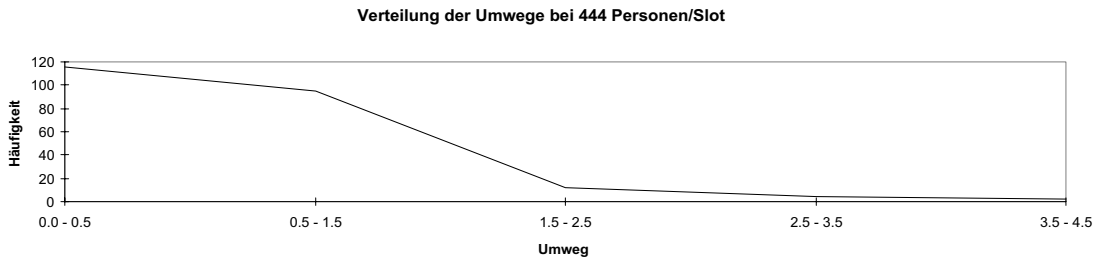


Abbildung 8.9: Verteilung der Häufigkeit der Umwege in km bei 444 Anbietern und 444 Nachfragern.

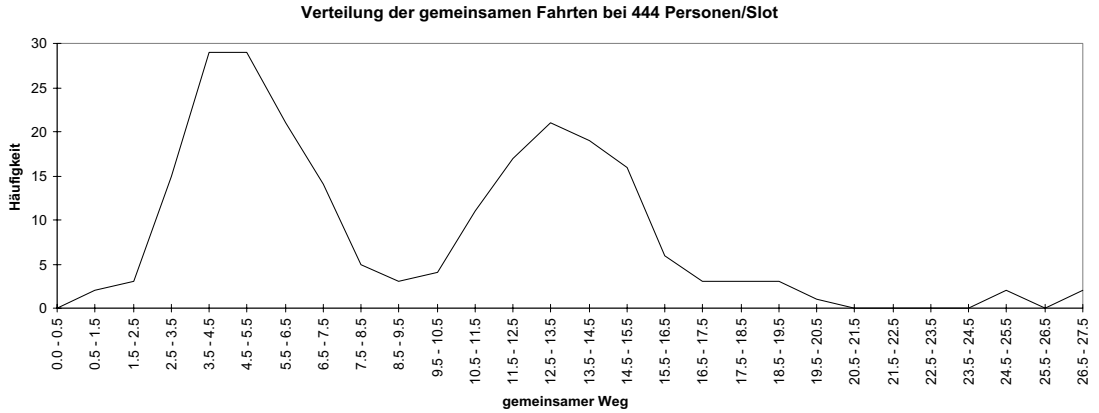


Abbildung 8.10: Verteilung der Häufigkeit der gemeinsamen Wege in km bei 444 Anbietern und 444 Nachfragern.

Fahrleistung aller Anbieter und Nachfrager liegt bei 14,3 km.

Analog dazu soll der Fall mit 1.333 Anbieter- und Nachfragerpaaren pro Slot betrachtet werden. Das eben dargestellte Ergebnis kann auch in diesem Fall bestätigt werden (Abbildung 8.11 für die Umwege, Abbildung 8.12 für die gemeinsam gefahrenen Wege).

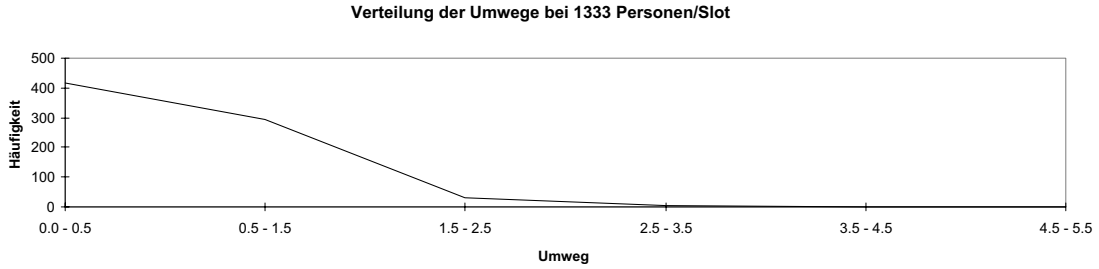


Abbildung 8.11: Verteilung der Häufigkeit der Umwege in km bei 1.333 Anbietern und 1.333 Nachfragern.

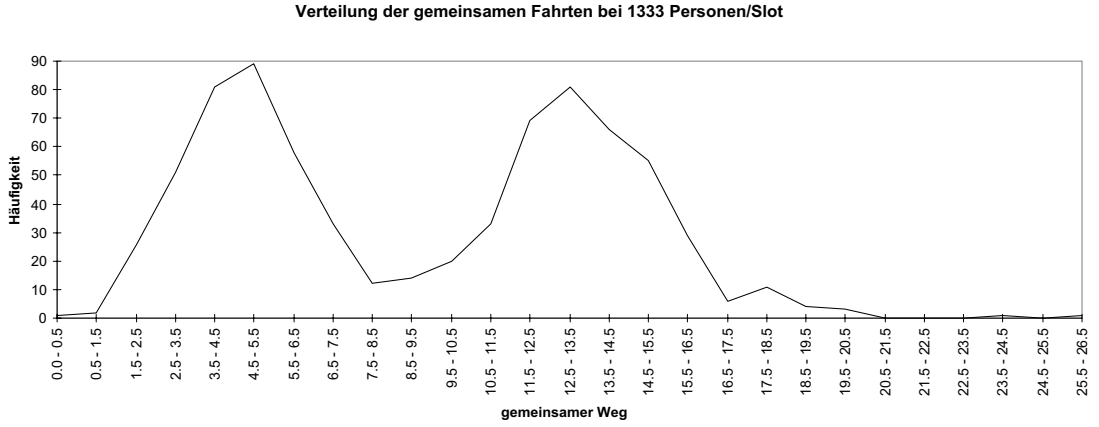


Abbildung 8.12: Verteilung der Häufigkeit der gemeinsamen Wege in km bei 1.333 Anbietern und 1.333 Nachfragern.

Auch hier zeichnet sich eine Verschiebung zum Ursprung hin ab. Die ohne Vermittlung im Durchschnitt gefahrenen Kilometer betragen 13,7 km; nach der Vermittlung werden im Schnitt nur 9,2 km gemeinsam gefahren.

Testlauf

Um die oben aufgeführte Begründung zu beweisen, soll ein weiterer Testlauf durchgeführt werden. Ich habe gesagt, daß bei Vermittlungen aufgrund zeitlicher und topographischer Verhältnisse eine Verschiebung der durchschnittlich gefahrenen Wege in Null-Richtung stattfindet. Dementsprechend muß bei einem speziellen Fall eine Gleichverteilung ohne Verschiebung existieren. Zu diesem Zweck wurde eine bestimmte Population generiert, die in einem kleinem Gebiet startet und die in einem entfernten, ebenfalls kleinen Gebiet ihre Ziele hat. Alle generierten Personen starten innerhalb dieses Gebietes und fahren in das Zielgebiet ein. Die Reisezeit-Toleranz δRZ beträgt dabei standardmäßig 10%. Unter diesen Umständen müßten alle Nachfrager mitgenommen werden, da alle teilnehmenden Personen gehäuft aufeinander wohnen und die Reisezeit-Toleranz entsprechend groß ist, daß die benötigten Umwege gefahren werden können. Somit ist eine Vermittlungsquote von 100% zu erwarten, sowie eine Fahrleistungsreduktion von 50%. Da alle Personen gleiche Start- und Zielpunkte haben, fahren sie die gleiche Strecke bzw. haben eine Nachfrage bezüglich der gleichen Strecke. Die Wege sind im Mittel gleich lang, und da eine Kapazität von zwei Personen pro Pkw – also eine 2er Mitfahrt – gegeben ist, beträgt die maximale Reduktion 50%; wenn alle Nachfrager vermittelt werden können.

CORONA wurde mit diesen Daten gestartet und als Ergebnis erhalten wir folgende Verteilungen (Abbildung 8.13 und 8.14).

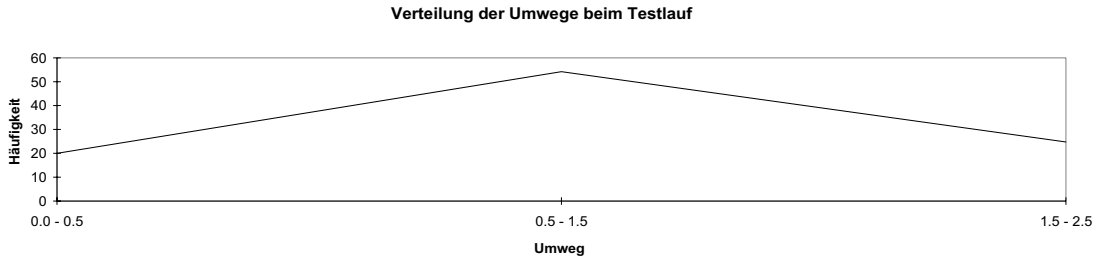


Abbildung 8.13: Verteilung der Häufigkeit der Umwege in km bei einem trivialen – zur Verifikation dienenden – Testlauf.

Wie zu ersehen ist, ist das Ergebnis eine einer Normalverteilung entsprechenden Kurve. Die durchschnittlich gemeinsam gefahrenen Kilometer betragen hierbei 19,2 km; die Verteilung der Umwege ist im Schnitt 1 km. Somit kann davon ausgegangen werden, daß CORONA korrekte Berechnungen anstellt.

Welche quantitativen Aussagen durch CORONA und welche Fahrleistungsreduktion in

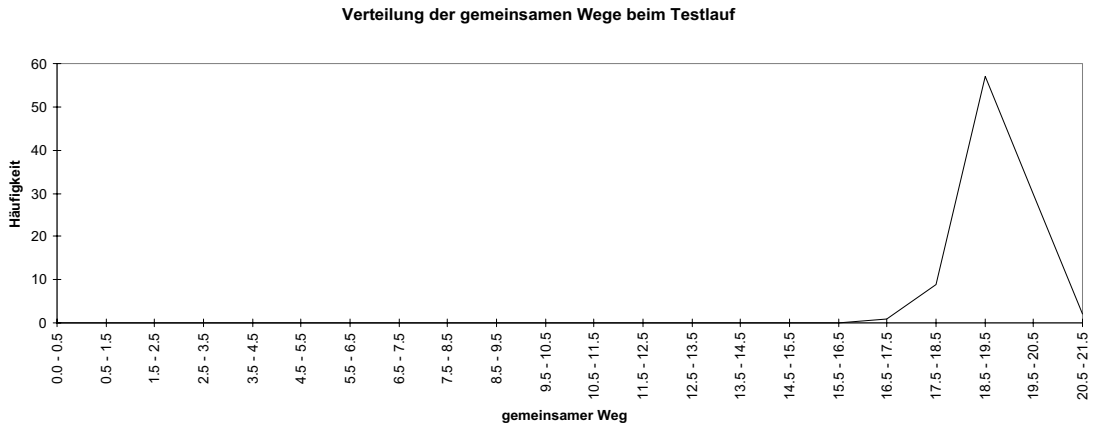


Abbildung 8.14: Verteilung der Häufigkeit der gemeinsamen Wege in km bei einem trivialen – zur Verifikation dienenden – Testlauf.

einer Region möglich erscheinen, kann in [HZ98] nachgelesen werden.

Zur Prüfung auf Korrektheit soll an dieser Stelle zusätzlich auf den Abschnitt 6.2.4 verwiesen werden, in welchem das Suchen von Routen im CORONA-System mit einem kommerziellen Navigationssystem verglichen wird. Es wird gezeigt, daß die von CORONA gefundenen Routen nur unwesentlich von denen des Navigationssystems abweichen und somit davon ausgegangen werden kann, daß das System im Mittel korrekte Routen berechnet.

8.2.3 Test der einzelnen Komponenten

Der im vorherigen Abschnitt beschriebene, ausführliche Testlauf kann nur dann als korrekt angesehen werden, wenn die einzelnen Komponenten *Generator*, *Mitfahrvermittlungssystem* und *Monitor* korrekt arbeiten. Um zu zeigen, daß sie in der Tat korrekt funktionieren, sollen diese drei Einheiten durch einige Testläufe geprüft werden.

Generator

Um zu zeigen, daß der Generator korrekt arbeitet, soll ein einfacher Test vorgenommen werden. Es werden zwei Cluster definiert (ein Start- und ein Ziel-Cluster)[5]. Der Generator soll nun fünf Anbieter und fünf Nachfrager erzeugen, die alle zu 100% ihren Startpunkt sowie ihren Zielpunkt in den jeweiligen Clustern haben (Abbildung 8.15). Das Ergebnis ist in Abbildung 8.16 dargestellt.

[5]Der Start-Cluster befindet sich im Ostertor-Viertel, der Ziel-Cluster in Sebaldsbrück

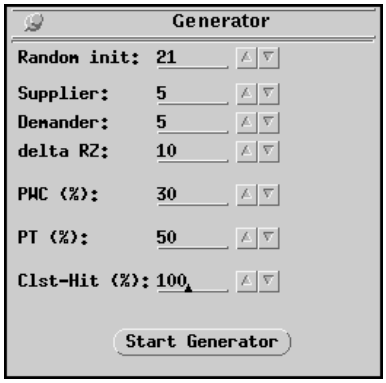


Abbildung 8.15: Generatorparameter mit fünf Anbietern, fünf Nachfragern und 100% Cluster-Hit.

Abbildung 8.16: Ergebnis des Generatortests mit den Parametern aus Abbildung 8.15.

Alle zehn Personen haben ihren Startpunkt im Start-Cluster (links) und ihren Zielpunkt im Ziel-Cluster (rechts)[6].

Als weiter Test sollen die gleichen Cluster verwendet werden, jedoch nur 70% der Anbieter in diesen Clustern ihren Start- und Zielpunkt haben. Die Anzahl der Nachfrager ist in diesem Fall auf 0 gesetzt.

Auch hier ist zu sehen, daß der Generator seine Aufgabe korrekt erfüllt, da zum einen kein

[6]Einige Start- und Zielpunkte ragen scheinbar über den definierten Cluster-Bereich hinaus. Das liegt daran, daß die Beschriftung der Kanten am Ende selbiger erfolgt und diese teils im Cluster liegen, teils sich außerhalb befinden; dennoch sind die Daten korrekt, da die Straße innerhalb des definierten Gebietes verläuft – es ist also ein reines Darstellungsproblem.

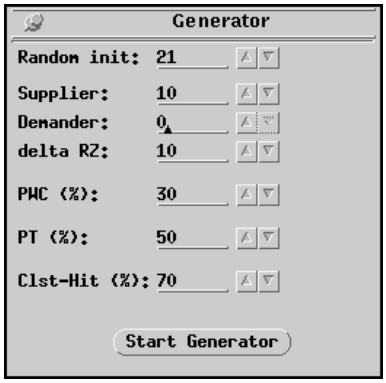


Abbildung 8.17: Generatorparameter mit zehn Anbietern, null Nachfragern und 70% Cluster-Hit.

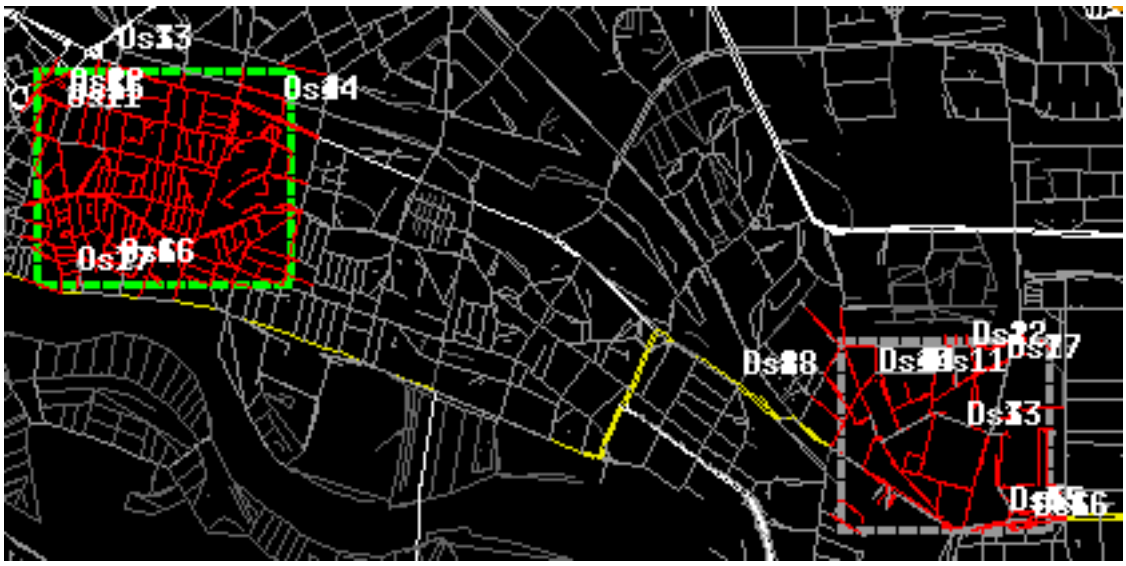


Abbildung 8.18: Ergebnis des Generatortests mit den Parametern aus Abbildung 8.17.

einziger Nachfrager erzeugt wurde und zum anderen, daß sieben Anbieter in den Clustern mti ihren jeweiligen Punkten anzutreffen sind. Die restlichen 30% sind gleichmäßig über das Stadtgebiet verteilt.

Daß diese gleichmäßige Verteilung ebenfalls korrekt ist, soll an folgendem Beispiel gezeigt werden: In Abblidung 8.19 waren die Generatorparameter *100 Anbieter*, *100 Nachfrager* und *keine Cluster*.

Wie in dieser Abbildung zu erkennen ist, finden sich größere Häufungen von Anbietern/Nachfragern im Zentrum. Das ist damit zu erklären, daß im Zentrum viele (kleine) Straßen – und somit viele Kanten – existieren, die alle ihre eigene Kanten-ID besitzen. Da der Generator Anbieter und Nachfrager über die Kanten-ID verteilt, ist dieses Ergebnis als korrekt anzusehen.

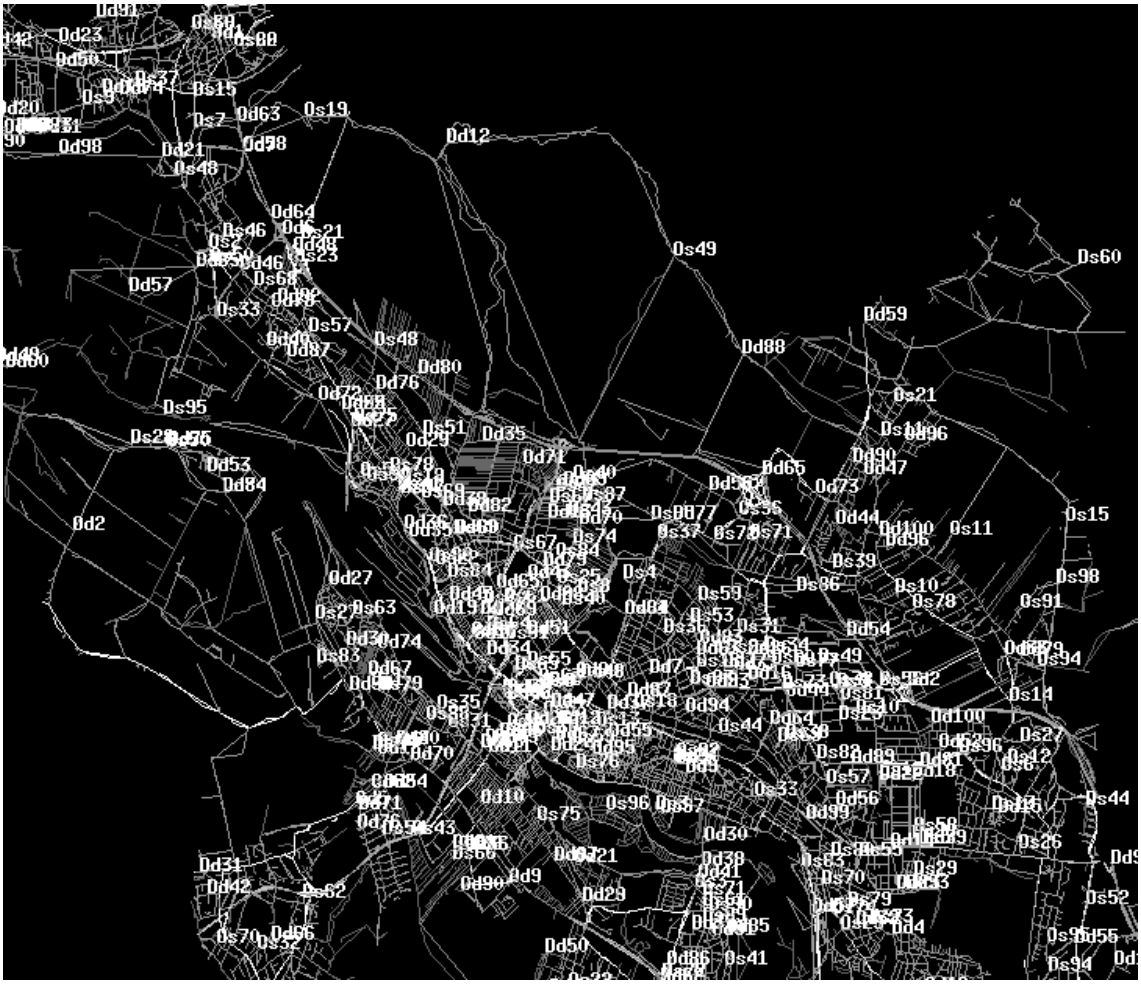


Abbildung 8.19: Gleichmäßige Verteilung von 100 Anbietern und 100 Nachfragern.

Der nächste Testlauf (Abbildung 8.20) soll ebenfalls 100 Anbieter und 100 Nachfrager erzeugen, diesmal jedoch mit der oben aufgeführten Clusterung (Abbildung 8.1) und einem *Cluster-Hit* von 100% – alle Anbieter und Nachfrager müssen dementsprechend innerhalb dieser Cluster ihre Start- und Zielpunkte haben.

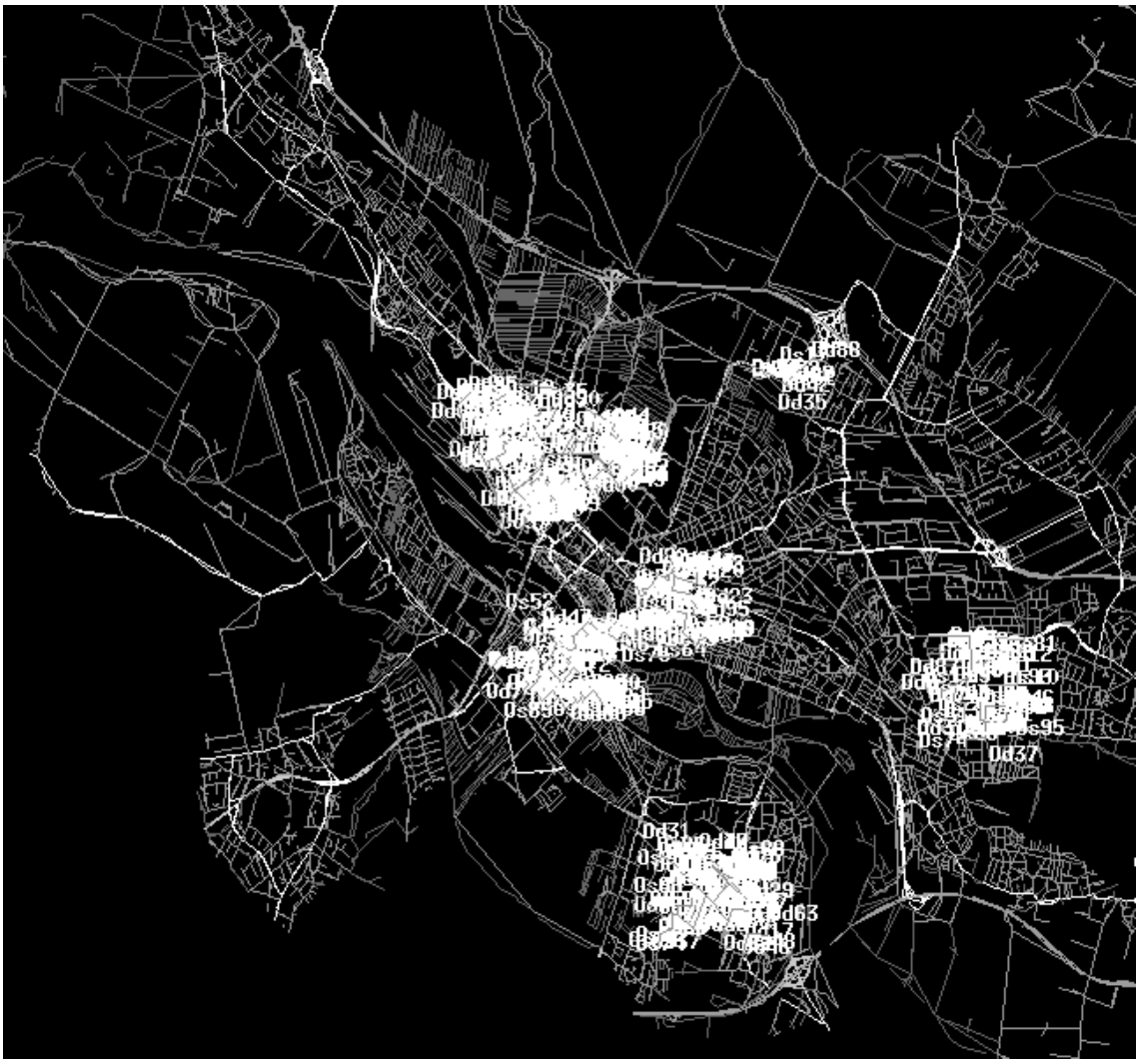


Abbildung 8.20: Verteilung von 100 Anbietern und 100 Nachfragern mit 100% Cluster-Hit.

Wie hier zu sehen ist, sind alle Anbieter und Nachfrager tatsächlich innerhalb der Cluster zu finden. Dementsprechend kann wiederum von einer korrekten Funktionsweise ausgegangen werden.

Vermittlungssystem

In diesem Abschnitt wird das Vermittlungssystem getestet. Dabei werden Vermittlungen für den ÖIV, den ÖPNV sowie für den gebrochenen Verkehr aufgeführt. Die Prioritätensteuerung wurde wie folgt gesetzt: 1. Mitnahme im Pkw, 2. gebrochener Verkehr und 3. reine ÖPNV-Vermittlung.

Um die Vermittlung – unter anderem im gebrochenen Verkehr – zu testen, soll eine

mitnahme- und mitfahrwillige Teilpopulation von 100 Anbietern und 100 Nachfragern pro Slot gleichmäßig über das Stadtgebiet von Bremen verteilt werden[7], wobei alle Nachfrager bereit sind, auch mit öffentlichen Verkehrsmitteln zu fahren. In Abbildung 8.21 sind alle Anbieter und Nachfrager mit ihren jeweiligen Start- und Zielpunkten aufgeführt. Zusätzlich sind alle Haltestellen des ÖPNV (hier die Haltestellen der BSAG) eingetragen, die durch die Vermittlung bedient werden.

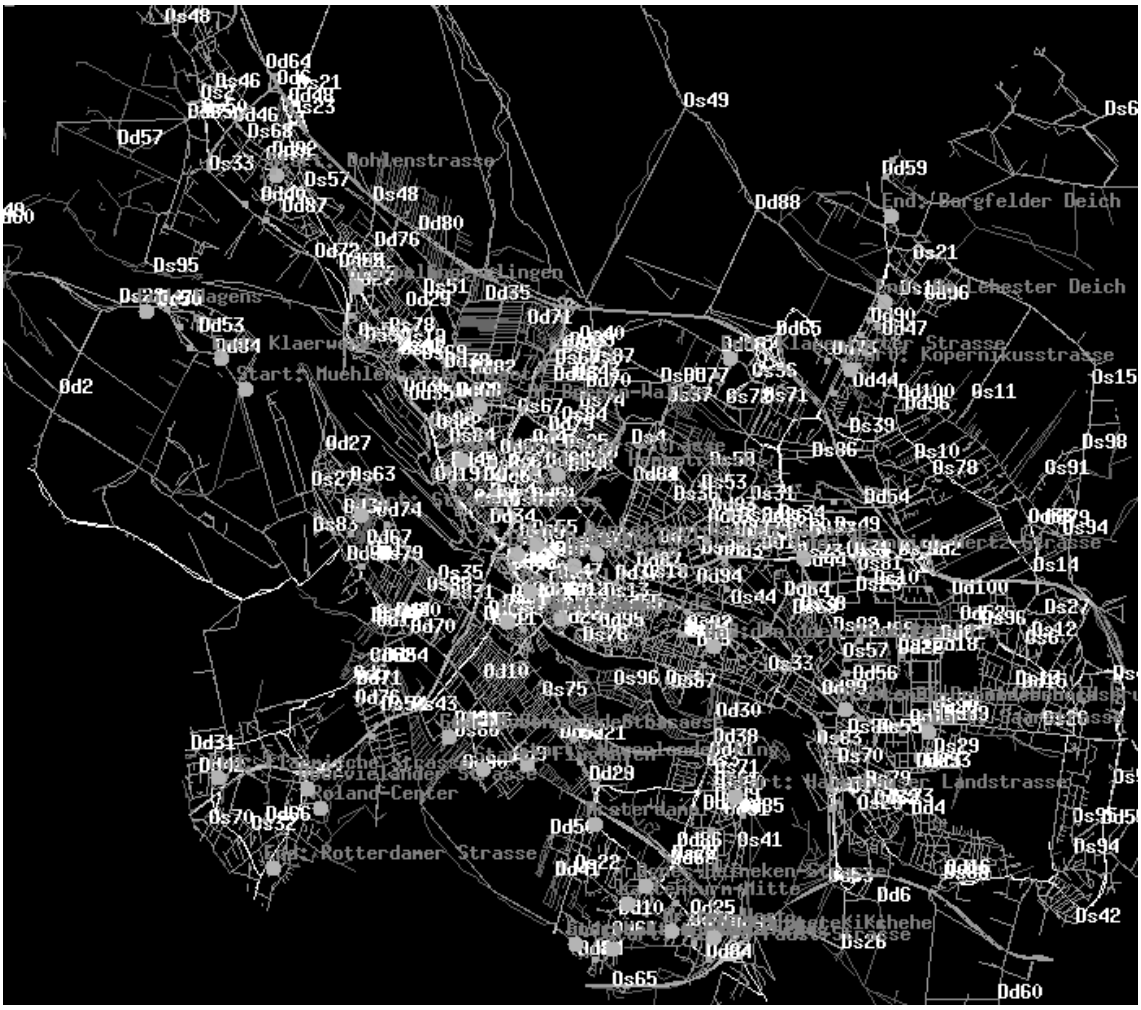


Abbildung 8.21: Ergebnis der Vermittlung von 100 Anbietern und 100 Nachfragern, bei der der gebrochene Verkehr berücksichtigt wird.

Die hier dargestellten Anbieter und Nachfrager sind teilweise im ÖIV, teilweise im ÖPNV und teilweise im gebrochenen Verkehr vermittelt worden. Im folgenden sollen einige Vermittlungen herausgesucht werden, mittels denen gezeigt werden kann, daß die Vermittlungen korrekt durchgeführt wurden.

[7] Die gleichmäßige Verteilung wurde aus dem Grunde gewählt, damit das Ergebnis beeser zu überblicken ist, siehe Abbildung 8.21.

In Abbildung 8.22 hat ein Anbieter (`Os20`) seinen Startpunkt in Bremen-Nord und seinen Zielpunkt (`Ds20`) in Bremen-Walle. Ein Nachfrager hat seinen Startpunkt (`Od20`) in Bremen-Gröpelingen und seinen Zielpunkt (`Dd20`) in Bremen-Arsten.

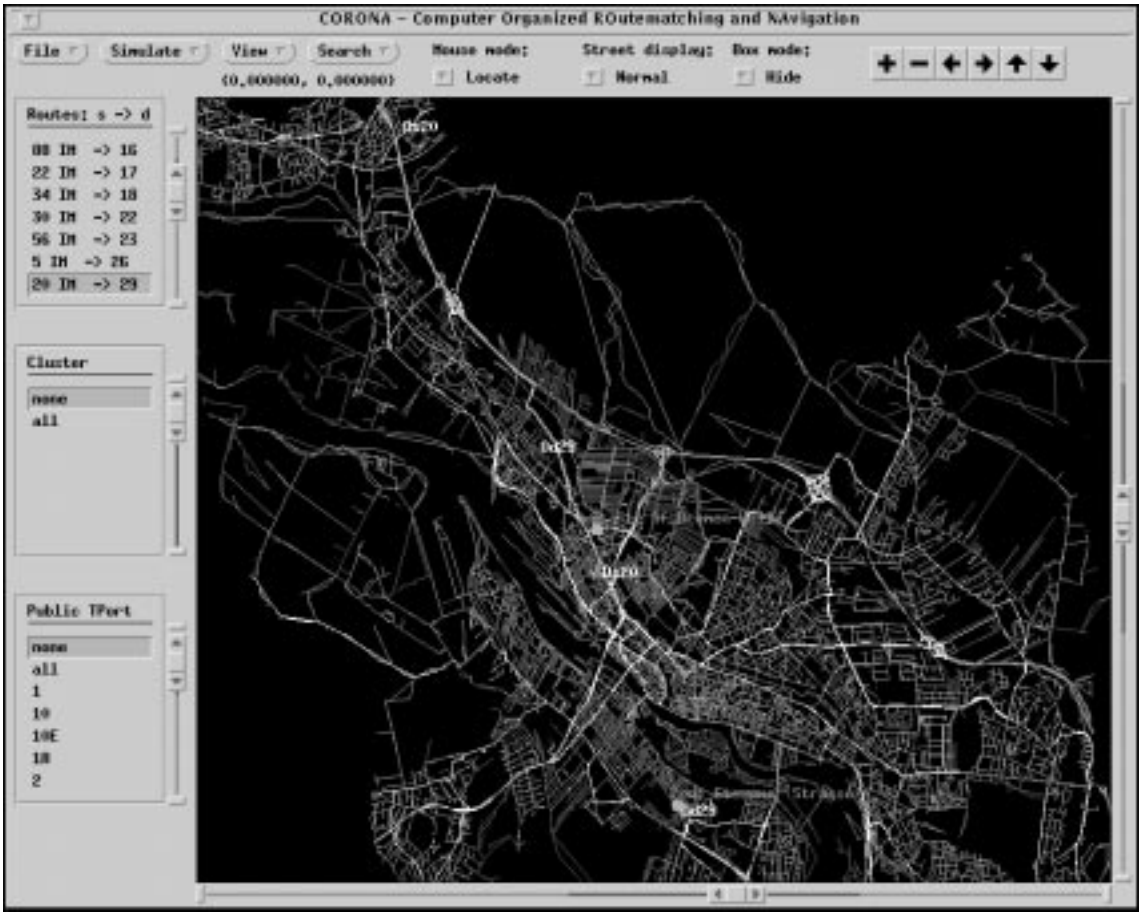


Abbildung 8.22: Beispiel einer Vermittlung im gebrochenen Verkehr mit dem CORONA-System.

Da ein Umweg von Walle nach Arsten (und zurück) außerhalb der Reisezeit-Toleranz des Anbieters liegt, hat CORONA folgende Vermittlung getroffen: Da der Anbieter in der Nähe des Nachfragers vorbeifährt, kann er diesen mitnehmen und ihn an der Haltestelle `Bf Bremen-Walle` absetzen. Diese Haltestelle ist innerhalb der definierten Zeit von Nachfrager zu Fuß zu erreichen. Der Anbieter fährt alleine zu seinem Ziel in Walle weiter, während der Nachfrager mit dem ÖPNV zur Haltestelle `Stenumer Straße` fährt. Diese Haltestelle liegt wiederum innerhalb des vom Nachfrager spezifizierten Radius zu seinem Zielpunkt in Arsten.

In Abbildung 8.23 ist ein weiterer Testfall aufgeführt. Ein Anbieter fährt von `Os30` nach `Ds30`. Ein Nachfrager möchte von `Od22` nach `Dd22`. Da dieser Nachfrager nicht per ÖIV

Abbildung 8.23: Ein weiteres Beispiel aus dem Testlauf mit dem CORONA-System zum gebrochenen Verkehr.

transportiert werden kann[8], wird er von Anbieter Nummer `30` mitgenommen und an der Haltestelle `Martinistraße` abgesetzt, so daß dieser dann mittels des ÖPNV zur Haltestelle `Im Holter Feld` fahren und von dort zu Fuß zu seinem Zielpunkt gelangen kann.

Nun sollen aus dem gleichen Testlauf Beispiele zum reinen ÖPNV-Transport erfolgen. Personen, die nicht die ganze Strecke mit einem Pkw mitgenommen werden können und die auch nicht mittels des gebrochenen Verkehrs an ihr Ziel gelangen konnten, werden daraufhin überprüft, ob sie mit alleiniger Hilfe des ÖPNV an ihr Ziel gelangen können.

[8]Siehe Priorität der Vermittlung.

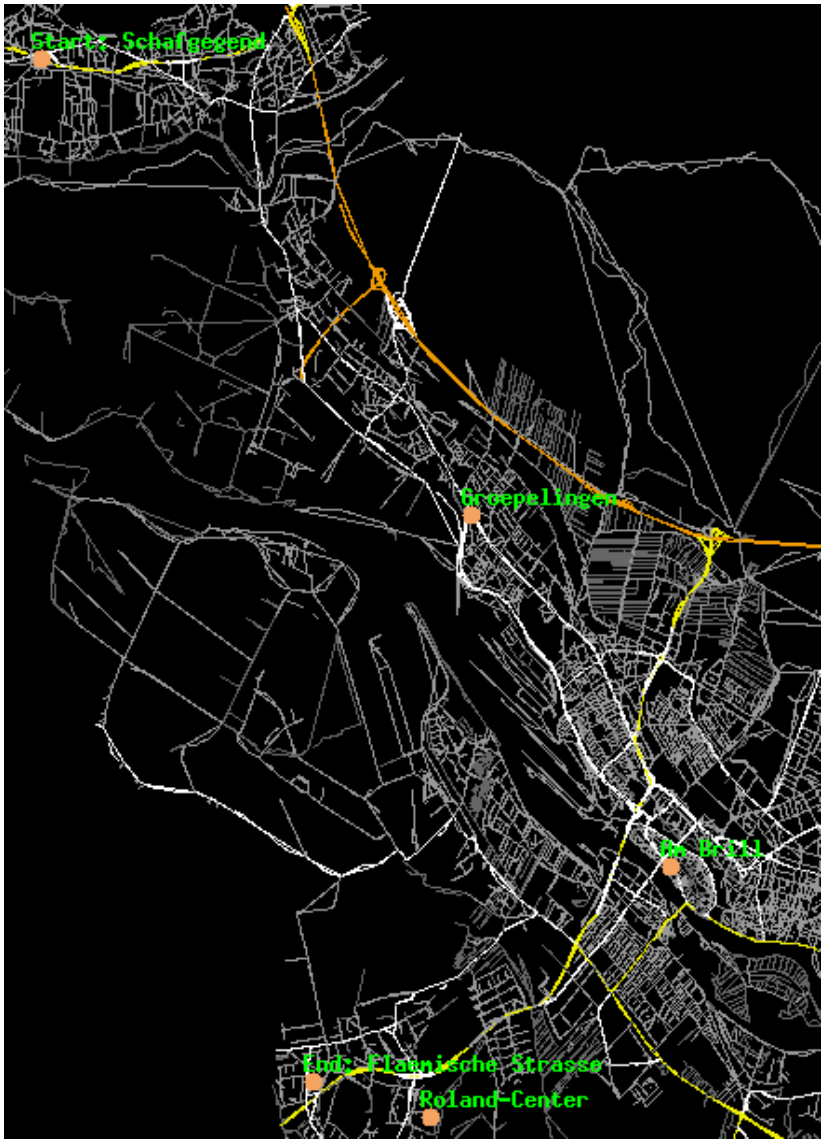


Abbildung 8.24: Vermittlung mit reinem ÖPNV-Transport.

In Abbildung 8.24 ist die Verbindung von Bremen-Vegesack nach Bremen-Huchting als Lösung dargestellt. Der Nachfrager muß bei dieser Route dreimal umsteigen. An der Haltestelle `Gröpelingen`, an der Haltestelle `Am Brill` und an der Haltestelle `Roland-Center`. Von dort aus kommt er definitionsgemäß in die Nähe zeines Zielpunktes.

Die Verbindungsdatei, die CORONA für diese Art von Vermittlungen mitschreibt, sieht dabei wie folgt aus:

```
Stops at source:
----------------
Schafgegend
 70 71

Stops at destination:
---------------------
Flaemische Strasse
 57 58


Connections:
------------
Stop: Schafgegend
(70)
Stop: Groepelingen
(2)
Stop: Am Brill
(6)
Stop: Roland-Center
(57)
Stop: Flaemische Strasse
```

Zunächst werden die Haltestellen in der Nähe des Start- und Zielpunktes gesucht und ausgegeben. Anschließend wird die Verbindung aufgeschrieben (`Connections`). Die Zahlen in Klammern unter den Haltestellennamen geben die Linie an, mit der der jeweilige Nachfrager fahren soll.

Eine weitere Vermittlung führt vom `Flughafen` in die Nähe des `Lehester Deichs` (Abbildung 8.25); der Nachfrager muß zunächst in die Linie `5` einsteigen, an der Blumenthalstraße in den Schnellbus `30S` wechseln und diesen `Am Leherster Deich` verlassen.

Abbildung 8.25: Weiterer Testfall mit reiner ÖPNV-Vermittlung.

Die Verbindungsdatei zu dieser Vermittlung findet am Zielpunkt Haltestellen der Linien `30S` und `31`. Da das Ziel mittels des Schnellbusses `30S` einfacher (und schneller) erreicht werden kann, wird diese Verbindung als Vermittlung eingetragen. Zur Vollständigkeit wird die hierzu erzeugte Verbindungsliste aufgeführt.

```
Stops at source:
----------------
Flughafen
 5 52

Stops at destination:
---------------------
Am Lehester Deich
 30S

Hoegerweg
 30S

Leher Feld
 31

Ostwaldstrasse
 31
```

```
Peter-Henlein-Strasse
 30S


Connections:
------------
Stop: Flughafen
(5)
Stop: Blumenthalstrasse
(30S)
Stop: Am Lehester Deich
```

Auf der folgenden Seite sind die statistischen Daten der Vermittlung – so wie CORONA sie ausgibt – ausgewiesen (Abbildung 8.26). Wie dort zu ersehen ist, sind 100% der Nachfrager bereit, mit dem ÖPNV zu fahren, bzw. in diesen vermittelt zu werden — einschließlich gebrochener Verkehr. Es existieren keine Autolosen, d.h. alle Nachfrager besitzen einen Pkw. Diese Abbildung 8.26 ist das Ergebnis eines Testlaufs mit der oben beschriebenen Slot-Population von 444 Anbietern und 444 Nachfragern sowie den Start- und Ziel-Clustern.

Die reine ÖPNV-Vermittlung besitzt dabei einen Anteil von ca. 6%, während der ÖPNV-Anteil im gebrochenen Verkehr sehr viel höher liegt. Dieses liegt daran, daß beim gebrochenen Verkehr der Nachfrager von seiner Wohnung abgeholt und zu einer Haltestelle der Linie gebracht wird, die ihn *innerhalb seiner Reisezeit-Toleranz* zum Ziel bringt. Bei reiner ÖPNV-Vermittlung muß der betreffende Nachfrager mehrfach umsteigen (siehe obige Beispiele), welches dessen Reisezeit-Toleranz oft überschreitet, da der ÖPNV, gemessen von Haus-zu-Haus, im Mittel halb so schnell ist wie der MIV.

CORONA
Statistics

Statistics for >> bremen <<		
Supplier	:	443
Demander	:	444
dRZ	:	10 %
People without cars	:	0 %
People in public transport	:	100 %
Match-Box survivers	:	442
Mediations in cars	:	197
Mediations intermodal	:	183
Mediations in public transport	:	28
Failed mediations	:	36
Maximal ways to compute	:	30464
Time for mediations	:	4461 sec.
Number of computed ways	:	9032
Failed ways	:	0
Average time	:	0.494 sec./way
KT Supplier	:	5498 km
KT Demander	:	5260 km
KT Sum	:	10759 km
KT Reduction Cars	:	1710 km
KT Reduction in PT	:	2868 km
KT Reduction Sum	:	4579 km
KT Reduction (fleet)	:	42.56 %

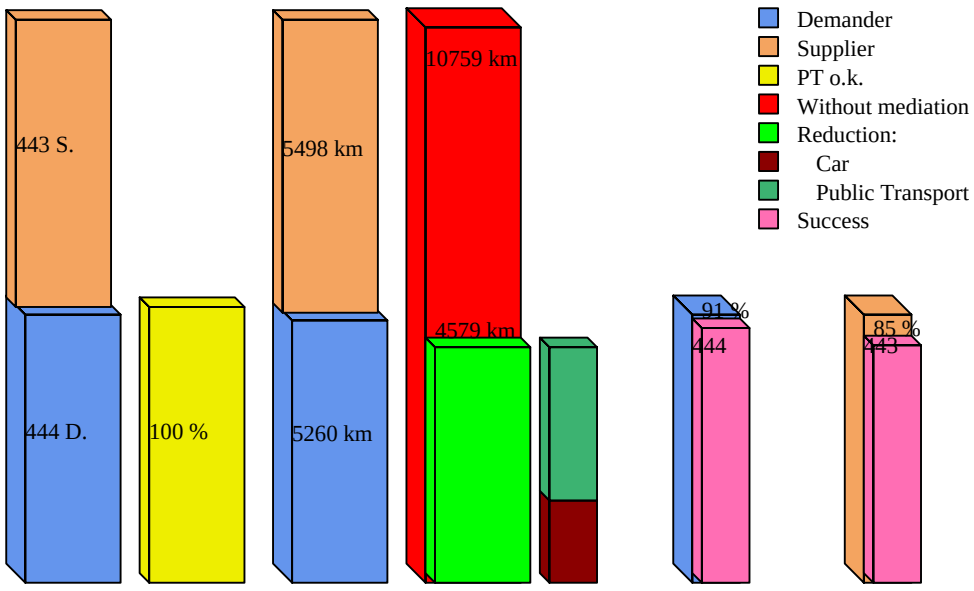


Abbildung 8.26: Darstellung der statistischen Daten mit Vermittlung im gebrochenen Verkehr bei 888 Personen/Slot.

Da im CORONA-System zwei Algorithmen zur Wegesuche implementiert wurden, sollen diese beiden miteinander verglichen werden[9]. Bei dieser Gelegenheit soll auch gezeigt werden, daß die PersonenLogistik CORONA nicht allein auf Bremen beschränkt ist, sondern auch mit Graphen anderer Städte bzw. Regionen umgehen kann. Zu diesem Zweck sind die nun folgenden Beispiele für die Stadt *Hildesheim* aufgeführt. Die vektorisierte Karte von Hildesheim wurde ebenfalls mittels `convert` in ein für CORONA nutzbares Format (`umf`) gebracht. So ist hier gleichzeitig getestet worden, daß CORONA für beliebige Städte – deren Straßennetz im `GDF`-Format vorliegen – eingesetzt werden kann.

Die Abbildung 8.27 zeigt einen Ausschnitt der Stadt Hildesheim. Der Startpunkt des Nachfragers Nummer 3 (`Od3`) und dessen Zielpunkt (`Dd3`) sollen nun näher betrachtet werden. Die weiße Linie zeigt die von CORONA gefundene Route mit dem Algorithmus der *kurzen* Wege.

Abbildung 8.27: Ein *kurzer* Weg von `Od3` zu `Dd3` am Beispiel der Stadt Hildesheim.

Abbildung 8.28 zeigt den gleichen Nachfrager und identischem Start- bzw. Zielpunkt mit dem Unterschied, daß die weiße Linie diesesmal den *kürzesten* Weg beschreibt. Die Unterschiede zwischen diesen beiden Wegen – auf die Länge bezogen – beträgt ca. 350 Meter.

In Abbildung 8.29 ist das Ergebnis in einen eingescannten Stadtplan eingezeichnet worden, um die Unterschiede der beiden Suchverfahren besser zu kennzeichnen, da die von CORONA dargestellte Straßenstruktur in diesem Ausdruck leider nicht deutlich genug zu erkennen ist.

[9]Die hier verwendeten Algorithmen wurden in Abschnitt 6.2 erklärt.

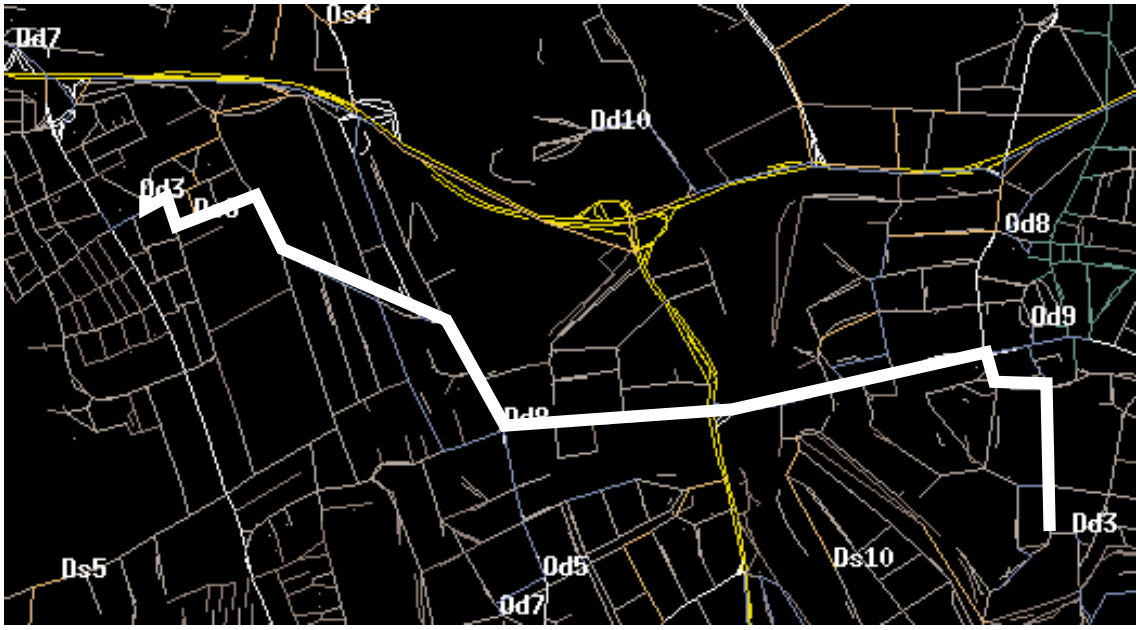


Abbildung 8.28: Der *kürzeste* Wege von `Od3` zu `Dd3` am Beispiel der Stadt Hildesheim.

Abbildung 8.29: Der *kurze* Weg und der *kürzeste* Weg von `Od3` zu `Dd3` der Abbildungen 8.27 und 8.28 auf einem konventionellen Stadtplan.

Wie an diesem Beispiel zu sehen ist, führt der kürzeste Weg durch ein Wohngebiet bzw. durch Nebenstraßen. Der Algorithmus für kurze Wege hingegen findet eine Route, die weitestgehend über Hauptstraßen verläuft. Wie ich bereits in Abschnitt 6.2.4 angedeutet habe, ist für ein solches Routenfindungsproblem ein Algorithmus notwendig, der sinnvolle Wege findet. Es nützt nichts, wenn ein Kurzwegalgorithmus verwendet wird, bei dem die Routen durch Tempo-30-Zonen verlaufen oder Umwege über Autobahnen bzw. Schnellstraßen vorschlagen, wenn dort zu bestimmten Zeiten Stau herrscht oder der Weg im Normalfall real nicht gefahren wird.

Monitor

Die Monitor-Komponente erhält ihre Daten aus einer von CORONA automatisch erzeugten Datei (`statistics.log`). Ist diese Datei nicht existent, so wird die Fehlermeldung

```
Error: No statistic data available!
```

ausgegeben. Dabei ist es egal, ob der Monitor von der Kommandozeile mit dem Befehl `corona -viewstat` bzw. `corona -viewstat <NAME>` oder von der Oberfläche aus mit dem Menüpunkt *View* die statistischen Daten aufgerufen werden; das System überprüft die Steuerdatei, gibt bei Nichtexistenz die entsprechende Meldung aus und läuft stabil weiter.

Wird die Datei allerdings modifiziert und dabei die Datenstruktur nicht eingehalten (siehe Abschnitt 7.3.6), so werden unkorrekte Daten angezeigt. Da die Datei aber automatisch generiert wird und individuell abgespeichert und wieder aufgerufen werden kann, macht es keinen Sinn, diese Datei manuell zu modifizieren. Der Benutzer kann den Monitor benutzen ohne sich um die interne Datstellung der Daten zu kümmern – für ihn ist die Datenstruktur transparent.

Alle in dieser Arbeit aufgeführten statistischen Darstellungen sind mittels dieses Monitors erstellt worden. Ferner sind alle Graphiken zur Darstellung der Straßenkarte von Bremen sowie zur Vermittlung von Personentransport mit CORONA erzeugt und angewählt worden, so daß von einer korrekten Funktionsweise dieses Bausteins gesprochen werden kann.

8.3 Robustheit

Bei allen in dieser Arbeit aufgeführten Tests lief das System einwandfrei — es gab keinerlei Abstürze oder Warnmeldungen. Lediglich bei einer Bevölkerungszahl von 4.000 Personen (2.000 Anbieter und 2.000 Nachfrager) stürzt das System nach langer Rechenzeit ab. Alle Versuche, den verursachenden Fehler zu finden, schlugen fehl, da zum einen die Rechenzeit im Debugger-System drastisch ansteigt (ca. 11 Stunden), und zum anderen der Absturz in einer Systembibliothek erfolgte. Ich konnte zwar den übergeordneten Aufruf im CORONA-System ermitteln, aber erkannte leider keinerlei Hinweise auf etwaige Fehler meinerseits. Aufgrund der extrem langen Wartezeit beim *debugging*, ist es aus Zeitgründen nicht möglich gewesen, den Fehler genau zu bestimmen.

Das CORONA-System wurde mit zwei verschiedenen vektorisierten Straßenkarten getestet: Einem Test-Datensatz der Stadt Hildesheim sowie der bereits erwähnten Karte der Stadt Bremen. Alle Programme (Converter, Flußgenerator, Fahrplangenerator und CORONA) arbeiteten – bis auf den erwähnten Fehler – korrekt. Beide Karten konnten in das `umf`-Format umgewandelt werden und auf beiden Karten konnten diverse Cluster gebildet, Populationen generiert und vermittelt werden.

Das System verhält sich bei unsinnigen Werten ebenfalls stabil. Wird ein Teil der Datenbank – z.B. die der Anbieter – nicht angegeben oder die Anzahl der Elemente auf Null gesetzt, so ist die Oberfläche bezüglich der Vermittlungen gesperrt. Ebenso können keine Vermittlungen durchgeführt werden, wenn nicht vorher eine Datenbank gegenriert oder eine bereits generierte Datenbank eingeladen wurde. Die Parameter des Generators werden zusätzlich darauf überprüft, ob diese in definierten Bereichen liegen; negative Anzahlen von Personen sind z.B. nicht definierbar. Sollten Datensätze von einer anderen Straßenkarte geladen werden, so werden diese nicht akzeptiert, da die geographischen Korrdinaten dementsprechend außerhalb des Definitionsbereiches liegen. Auf der Kommandozeilenebene werden die Parameter ebenfalls überprüft; so wird das CORONA-System nicht gestartet, wenn nicht alle angegebenen Steuerdateien existieren und gültig sind. Für den Fall, daß kein ÖPNV-Fahrplan für eine Linie zu Verfügung steht, geht CORONA davon aus, daß die entsprechende Linie im Sekundentakt fährt. Ist keine ÖPNV-Haltestellendatei zu der gewünschten Straßenkarte vorhanden, wird der ÖPNV als nicht existierend im System geführt — dementsprechend entfallen die ÖPNV-spezifischen Abfragen und Darstellungen.

Ausgehend von diesen Tatsachen kann man festhalten, daß CORONA als eine Pilotimplementation – ursprünglich als Teilimplementation gedacht – weitestgehend stabil arbeitet.

8.4 Leistungsfähigkeit

Wie bereits in Abbildung 8.3 auf Seite 96 dargestellt wurde, benötigt CORONA relativ viel Rechenzeit, um große Anzahlen von Anbietern und Nachfragern pro Slot durchführen zu können. Für den realen Einzatz gilt allerdings die Tatsache, daß die hier aufgeführten Slot-Berechnungen eine Art *worst-case*-Abschätzung darstellen und zudem größtenteils nur einmalig stattfinden werden, da man von vielen konstanten Mitfahrten ausgehen kann. Im realen Einsatz empfiehlt es sich, eine redundante Systemarchitektur zu verwenden, da einerseits beim Ausfall eines Rechners ein anderer ein Sicherheitssystem darstellt und andererseits die Berechnung aufgeteilt werden kann.

Auf jeden Fall ist eine sehr schnelle Workstation ausreichend für eine Region, da durch die große Anzahl von Möglichkeiten zur Vermittlung – auch wenn sie durch die Heuristiken und Algorithmen schon gering gehalten ist – große Rechenkapazitäten benötigt werden.

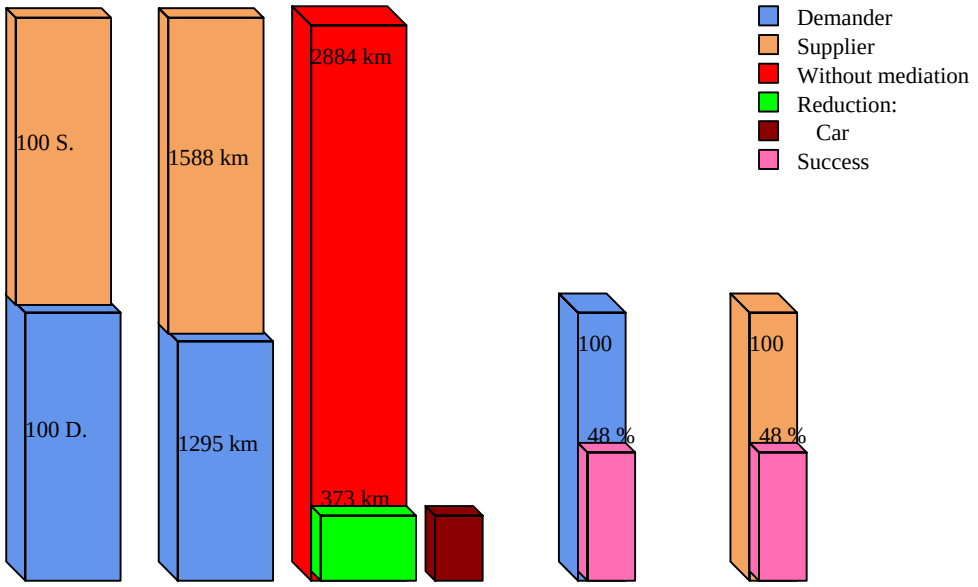
Weiterhin ist ein Großteil der geforderten Eigenschaften einer PersonenLogistik in CORONA implementiert worden und es steht ein leistungsfähiges operatives System zur Verfügung, um zum einen dem VS-Manager ein Softwaresystem zur Verfügungs zu stellen, mit dessen Hilfe er Angebot und Nachfrage nach Transportleistung abgleichen kann, und zum anderen, um Simulationen bezüglich der Fahrleistungsreduktion in einer Stadt durchführen zu können (siehe [HZ98] und Kapitel 9).

8.5 Statistische Daten

Auf den nachfolgenden Seiten sind die von CORONA erzeugten PostScript-Dateien mit den Ergebnissen der weiterhin durchgeführten Testläufe aufgeführt. Da die verwendete Darstellungsweise bereits in Abschnitt 7.4.2 erfolgte, soll an dieser Stelle nicht weiter darauf eingegangen werden. Dieser Abschnitt dient lediglich zur Auflistung der gewonnenen Ergebnisse.

CORONA
Statistics

Statistics for >> bremen <<		
Supplier	:	100
Demander	:	100
dRZ	:	10 %
People without cars	:	0 %
People in public transport	:	0 %
Match-Box survivers	:	95
Mediations in cars	:	48
Mediations intermodal	:	0
Mediations in public transport	:	0
Failed mediations	:	52
Maximal ways to compute	:	1488
Time for mediations	:	103 sec.
Number of computed ways	:	274
Failed ways	:	0
Average time	:	0.376 sec./way
KT Supplier	:	1588 km
KT Demander	:	1295 km
KT Sum	:	2884 km
KT Reduction Cars	:	373 km
KT Reduction in PT	:	0 km
KT Reduction Sum	:	373 km
KT Reduction (fleet)	:	12.96 %

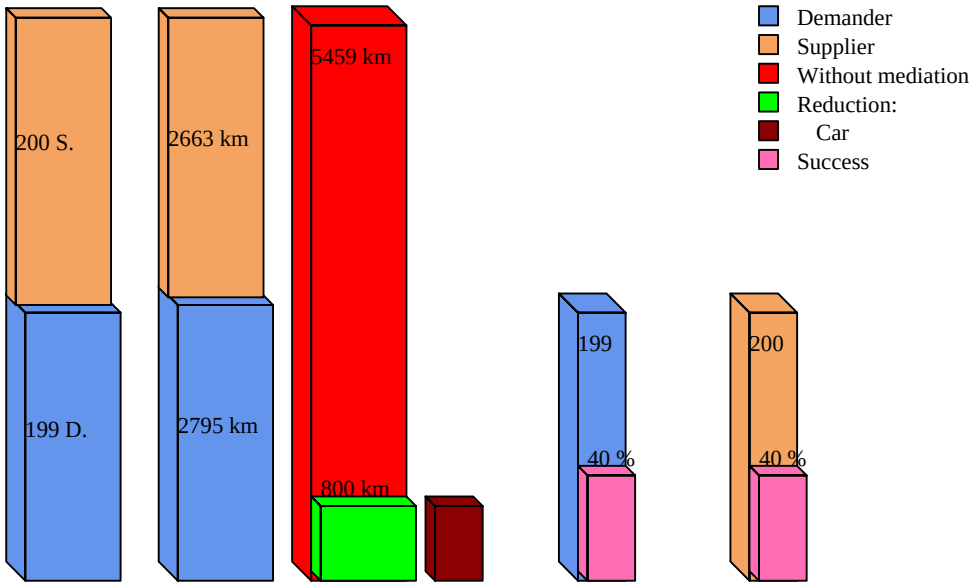


Institute of Transport and Computer Sciences (IIuV), University of Bremen, Dept. 3, D-28359 BREMEN - Germany

Abbildung 8.30: Statistik für Bremen mit 70% Cluster-Hit und drei Start-Clustern sowie vier Ziel-Clustern und 200 Personen/Slot.

CORONA
Statistics

Statistics for >> bremen <<		
Supplier	:	200
Demander	:	199
dRZ	:	10 %
People without cars	:	0 %
People in public transport	:	0 %
Match-Box survivers	:	190
Mediations in cars	:	80
Mediations intermodal	:	0
Mediations in public transport	:	0
Failed mediations	:	119
Maximal ways to compute	:	3430
Time for mediations	:	288 sec.
Number of computed ways	:	800
Failed ways	:	0
Average time	:	0.360 sec./way
KT Supplier	:	2663 km
KT Demander	:	2795 km
KT Sum	:	5459 km
KT Reduction Cars	:	800 km
KT Reduction in PT	:	0 km
KT Reduction Sum	:	800 km
KT Reduction (fleet)	:	14.67 %



Institute of Transport and Computer Sciences (IIuV), University of Bremen, Dept. 3, D-28359 BREMEN - Germany

Abbildung 8.31: Statistik für Bremen mit 70% Cluster-Hit und drei Start-Clustern sowie vier Ziel-Clustern und 400 Personen/Slot.

CORONA
Statistics

Statistics for >> bremen <<		
Supplier	:	300
Demander	:	300
dRZ	:	10 %
People without cars	:	0 %
People in public transport	:	0 %
Match-Box survivers	:	291
Mediations in cars	:	137
Mediations intermodal	:	0
Mediations in public transport	:	0
Failed mediations	:	163
Maximal ways to compute	:	8742
Time for mediations	:	531 sec.
Number of computed ways	:	1310
Failed ways	:	0
Average time	:	0.405 sec./way
KT Supplier	:	4345 km
KT Demander	:	4151 km
KT Sum	:	8496 km
KT Reduction Cars	:	1330 km
KT Reduction in PT	:	0 km
KT Reduction Sum	:	1330 km
KT Reduction (fleet)	:	15.66 %

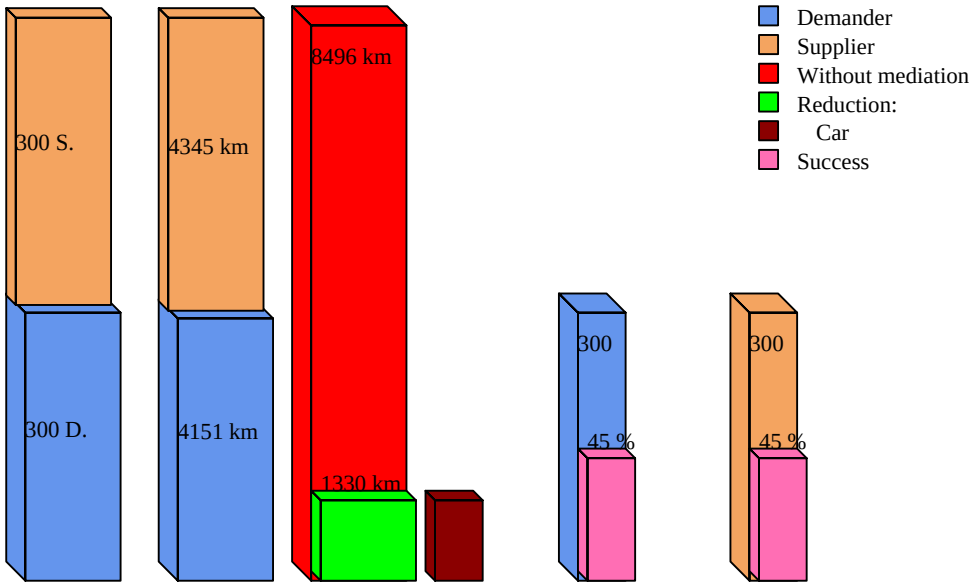
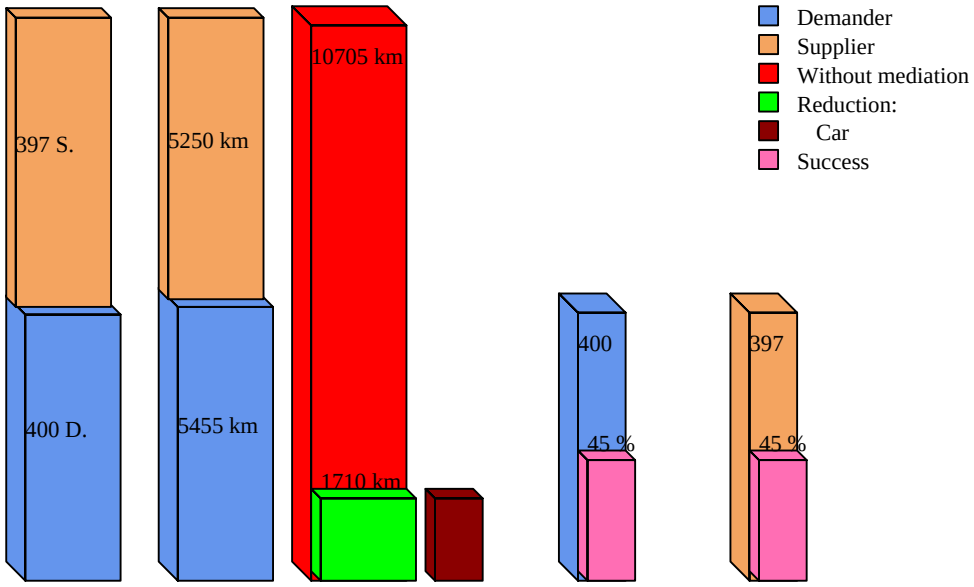


Abbildung 8.32: Statistik für Bremen mit 70% Cluster-Hit und drei Start-Clustern sowie vier Ziel-Clustern und 600 Personen/Slot.

CORONA
Statistics

Statistics for >> bremen <<		
Supplier	:	397
Demander	:	400
dRZ	:	10 %
People without cars	:	0 %
People in public transport	:	0 %
Match-Box survivers	:	369
Mediations in cars	:	181
Mediations intermodal	:	0
Mediations in public transport	:	0
Failed mediations	:	219
Maximal ways to compute	:	12138
Time for mediations	:	1007 sec.
Number of computed ways	:	2090
Failed ways	:	0
Average time	:	0.482 sec./way
KT Supplier	:	5250 km
KT Demander	:	5455 km
KT Sum	:	10705 km
KT Reduction Cars	:	1710 km
KT Reduction in PT	:	0 km
KT Reduction Sum	:	1710 km
KT Reduction (fleet)	:	15.97 %

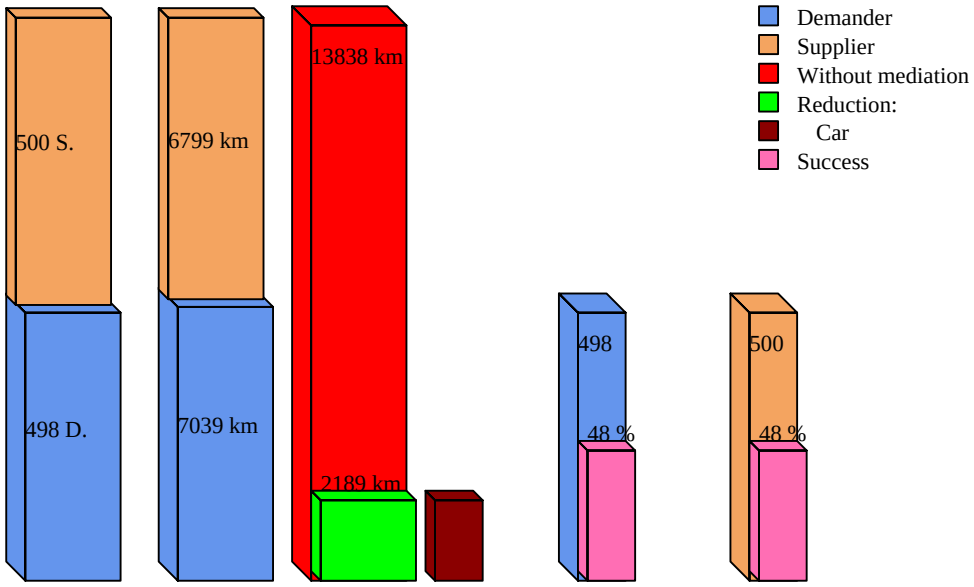


Institute of Transport and Computer Sciences (IIuV), University of Bremen, Dept. 3, D-28359 BREMEN - Germany

Abbildung 8.33: Statistik für Bremen mit 70% Cluster-Hit und drei Start-Clustern sowie vier Ziel-Clustern und 800 Personen/Slot.

CORONA
Statistics

Statistics for >> bremen <<		
Supplier	:	500
Demander	:	498
dRZ	:	10 %
People without cars	:	0 %
People in public transport	:	0 %
Match-Box survivers	:	476
Mediations in cars	:	242
Mediations intermodal	:	0
Mediations in public transport	:	0
Failed mediations	:	256
Maximal ways to compute	:	22014
Time for mediations	:	1472 sec.
Number of computed ways	:	2767
Failed ways	:	0
Average time	:	0.532 sec./way
KT Supplier	:	6799 km
KT Demander	:	7039 km
KT Sum	:	13838 km
KT Reduction Cars	:	2189 km
KT Reduction in PT	:	0 km
KT Reduction Sum	:	2189 km
KT Reduction (fleet)	:	15.82 %

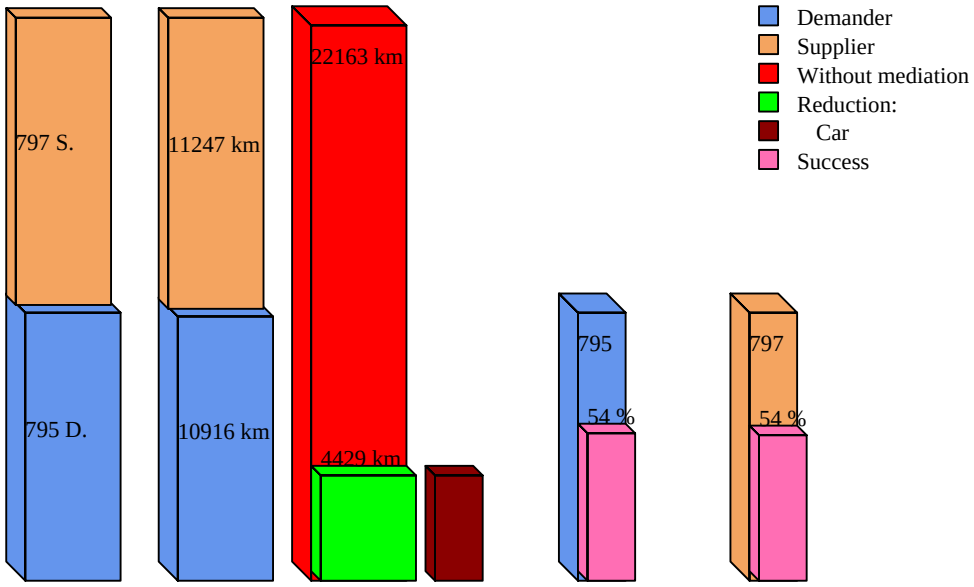


Institute of Transport and Computer Sciences (IIuV), University of Bremen, Dept. 3, D-28359 BREMEN - Germany

Abbildung 8.34: Statistik für Bremen mit 70% Cluster-Hit und drei Start-Clustern sowie vier Ziel-Clustern und 1.000 Personen/Slot.

CORONA
Statistics

Statistics for >> bremen <<		
Supplier	:	797
Demander	:	795
dRZ	:	10 %
People without cars	:	0 %
People in public transport	:	0 %
Match-Box survivers	:	785
Mediations in cars	:	433
Mediations intermodal	:	0
Mediations in public transport	:	0
Failed mediations	:	362
Maximal ways to compute	:	62326
Time for mediations	:	4608 sec.
Number of computed ways	:	7082
Failed ways	:	0
Average time	:	0.651 sec./way
KT Supplier	:	11247 km
KT Demander	:	10916 km
KT Sum	:	22163 km
KT Reduction Cars	:	4429 km
KT Reduction in PT	:	0 km
KT Reduction Sum	:	4429 km
KT Reduction (fleet)	:	19.99 %



Institute of Transport and Computer Sciences (IIuV), University of Bremen, Dept. 3, D-28359 BREMEN - Germany

Abbildung 8.35: Statistik für Bremen mit 70% Cluster-Hit und drei Start-Clustern sowie vier Ziel-Clustern und 1.600 Personen/Slot.

CORONA
Statistics

Statistics for >> bremen <<		
Supplier	:	996
Demander	:	997
dRZ	:	10 %
People without cars	:	0 %
People in public transport	:	0 %
Match-Box survivers	:	987
Mediations in cars	:	546
Mediations intermodal	:	0
Mediations in public transport	:	0
Failed mediations	:	451
Maximal ways to compute	:	96204
Time for mediations	:	6471 sec.
Number of computed ways	:	8758
Failed ways	:	0
Average time	:	0.739 sec./way
KT Supplier	:	14264 km
KT Demander	:	13744 km
KT Sum	:	28009 km
KT Reduction Cars	:	5382 km
KT Reduction in PT	:	0 km
KT Reduction Sum	:	5382 km
KT Reduction (fleet)	:	19.22 %

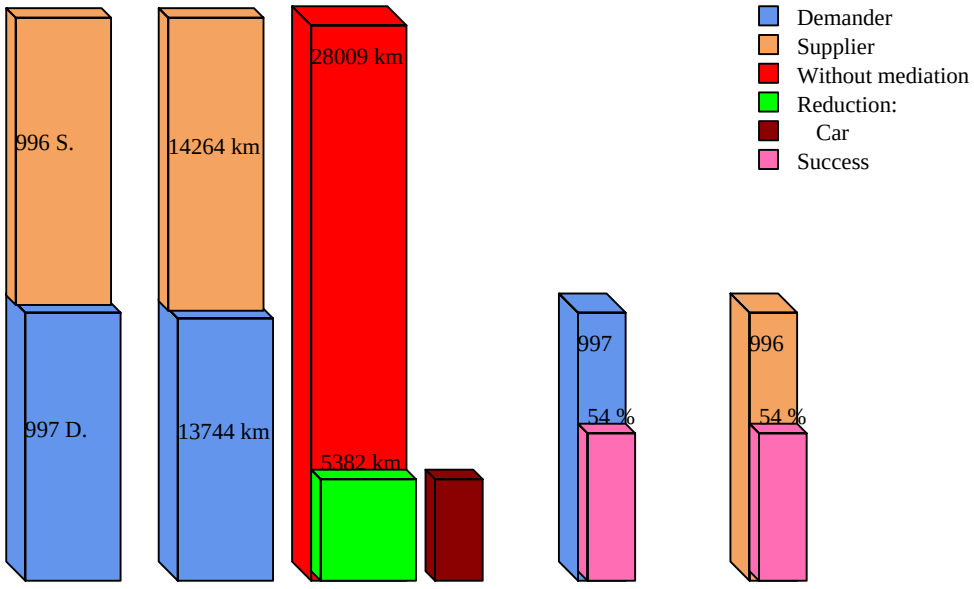
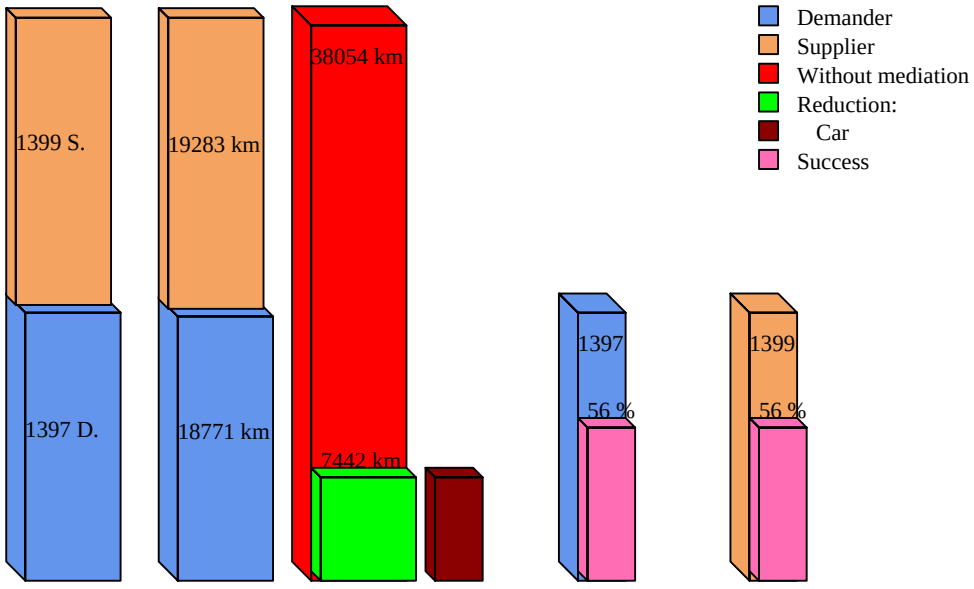


Abbildung 8.36: Statistik für Bremen mit 70% Cluster-Hit und drei Start-Clustern sowie vier Ziel-Clustern und 2.000 Personen/Slot.

CORONA
Statistics

Statistics for >> bremen <<		
Supplier	:	1399
Demander	:	1397
dRZ	:	10 %
People without cars	:	0 %
People in public transport	:	0 %
Match-Box survivers	:	1380
Mediations in cars	:	792
Mediations intermodal	:	0
Mediations in public transport	:	0
Failed mediations	:	605
Maximal ways to compute	:	177500
Time for mediations	:	13319 sec.
Number of computed ways	:	15010
Failed ways	:	0
Average time	:	0.887 sec./way
KT Supplier	:	19283 km
KT Demander	:	18771 km
KT Sum	:	38054 km
KT Reduction Cars	:	7442 km
KT Reduction in PT	:	0 km
KT Reduction Sum	:	7442 km
KT Reduction (fleet)	:	19.56 %

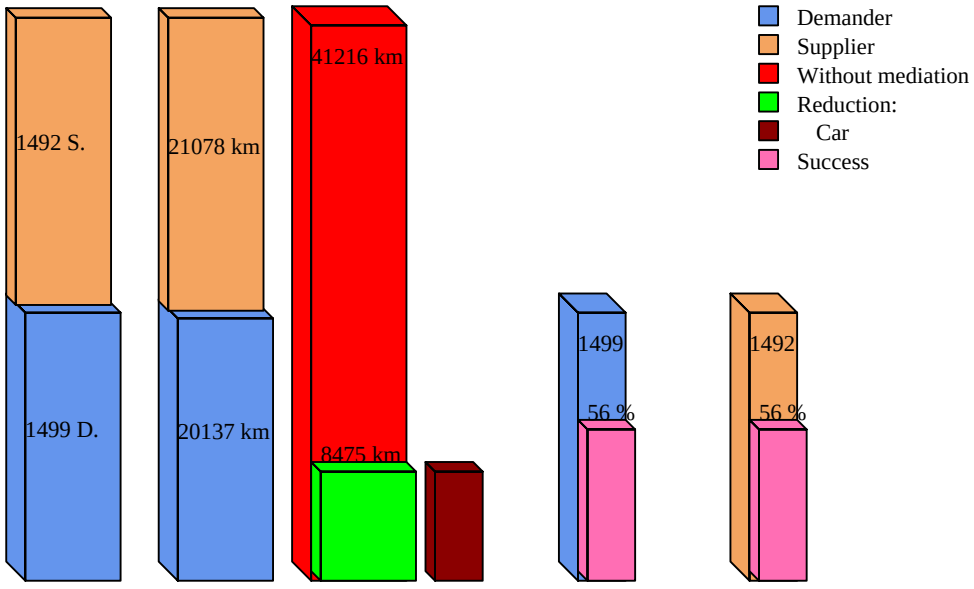


Institute of Transport and Computer Sciences (IIuV), University of Bremen, Dept. 3, D-28359 BREMEN - Germany

Abbildung 8.37: Statistik für Bremen mit 70% Cluster-Hit und drei Start-Clustern sowie vier Ziel-Clustern und 2.800 Personen/Slot.

CORONA
Statistics

Statistics for >> bremen <<		
Supplier	:	1492
Demander	:	1499
dRZ	:	10 %
People without cars	:	0 %
People in public transport	:	0 %
Match-Box survivers	:	1493
Mediations in cars	:	840
Mediations intermodal	:	0
Mediations in public transport	:	0
Failed mediations	:	659
Maximal ways to compute	:	216410
Time for mediations	:	18499 sec.
Number of computed ways	:	20882
Failed ways	:	0
Average time	:	0.886 sec./way
KT Supplier	:	21078 km
KT Demander	:	20137 km
KT Sum	:	41216 km
KT Reduction Cars	:	8475 km
KT Reduction in PT	:	0 km
KT Reduction Sum	:	8475 km
KT Reduction (fleet)	:	20.56 %



Institute of Transport and Computer Sciences (IIuV), University of Bremen, Dept. 3, D-28359 BREMEN - Germany

Abbildung 8.38: Statistik für Bremen mit 70% Cluster-Hit und drei Start-Clustern sowie vier Ziel-Clustern und 3.000 Personen/Slot.

9. Ergebnis und Ausblick

Die in der Motivation gestellte Frage nach der Machbarkeit eines PersonenLogistik-Systems kann durch diese Dissertation eindeutig mit *Ja* beantwortet werden. Es ist mir gelungen, ein – nach derzeitigem Stand des Wissens – weltweit neuartiges Computersystem zu entwickeln, mit dessen Hilfe es möglich ist, eine große Anzahl von Anbietern und Nachfragern für den Personentransport im MIV und gebrochenen Verkehr zu vermitteln. Dabei werden Reisezeit- und Startzeit-Toleranzen ebenso berücksichtigt, wie die reale Topographie einer Stadt in Form einer *digitalen, vektorisierten Straßenkarte*. Mittels dieser Repräsentation des Straßennetzes in Zusammenhang mit einem Mitfahrvermittlungssystem, welches das ÖPNV-Netz der jeweiligen Stadt beinhaltet, und einem Monitor zur Überprüfung der Ergebnisse, ist nun erstmalig ein operatives System für ein VerkehrsSystemManagement im Sinne des Schlanken Verkehrs entstanden.

Eine weitere Neuerung – neben der Integration von Vermittlungssystem und vektorisierter Straßenkarte – ist die Entwicklung von Heuristiken, um einen Abgleich zwischen *vielen* Anbietern und Nachfragern durchführen zu können. Kommerzielle Mitfahrzentralen vermitteln in der Regel relativ wenige Personen und berechnen keine Routen. Kommerzielle Routenplaner berechnen komplexe Routen, aber nur für relativ wenig Fahrzeuge. CORONA hingegen rechnet mit einer sehr großen Anzahl an Personen – und entsprechenden Routen –, da im Sinne des Schlanken Verkehrs davon ausgegangen wird, daß in einer Stadt wie Bremen mindestens 40.000 Pkw am System teilnehmen. Wie in dieser Dissertation gezeigt wurde, ist mittels der vorgestellten Heuristiken der Abgleich unter den geforderten Bedingungen überhaupt erst realisierbar.

Das CORONA-System ist als Pilotimplementation einer PersonenLogistik weiterhin in der Lage, mittels des zusätzlich implementierten Generators, als *Simulationssystem* Aussagen über eine mögliche Fahrleistungsreduktion in einer Stadt oder Region zu machen. Erste Ergebnisse solcher Simulationen finden sich in [HZ98]. Ich möchte an dieser Stelle erwähnen, daß die ebenfalls in der Motivation gestellte Frage nach einer möglichen Fahrleistungsreduktion durch Einführung neuer Verkehrsdienste gemäß Schlankem Verkehr mit *Ja* beantwortet werden kann (siehe Abschnitt 8.5 und [HZ98]).

Ferner ist es durch die Simulation möglich, den Wirkungsgrad des ÖPNV zu testen. Es können Aussagen darüber getätigt werden, welche Rolle der ÖPNV im täglichen Personenverkehr spielt oder, im Sinne des Schlanken Verkehrs, im gebrochenen Verkehr spielen kann. So ist als erstes Ergebnis einer Untersuchung festzuhalten, daß der reine

ÖPNV-Anteil an der Vermittlung innerhalb der Anbieter und Nachfrager nur ca. 6% beträgt.

Durch die Testläufe, die in Kapitel 8 aufgeführt sind, und die Simulationsläufe für [HZ98], ist ein *weitgehend stabiles* Computersystem verfügbar. Da CORONA inklusive aller Zusatzprogramme mehr als 12.500 Codezeilen der Programmiersprache C aufweist, sind selbstverständlich Fehler im Programm enthalten, die noch nicht entdeckt oder zwar bekannt, aber nicht lokalisiert worden sind. Weiterhin stürzt das System in einigen C-Bibliotheken ab, ohne daß ein Programmierfehler gefunden werden konnte.

Es ist leider nicht möglich gewesen, die Ein- und Ausgabekomponenten bzw. deren Schnittstellen zu CORONA zu implementieren, um die vollautomatische Funktionsweise komplett testen zu können. Das Institut für Informatik und Verkehr hat einen Antrag für ein Leitprojekt beim bmb+f bzw. dem Projektträger TÜV-Rheinland gestellt, bei dem es sich um die Demonstration einer Stadtgerechten Mobilität (Projekt: `DESMO`) – sowie deren Realisierung – handelt. In diesem Antrag ist beschrieben, daß unter anderem eine PersonenLogistik mit den definierten Ein- und Ausgabekomponenten eingesetzt werden muß, um das Projektziel zu erreichen. Sollte der Antrag bewilligt werden, ist es möglich, CORONA mit diesen vollautomatischen Komponenten zu betreiben, da in diesem Fall das benötigte Geld zur Verfügung steht, um sie zu entwickeln.

Lutz Zegartowski
Institut für Informatik und Verkehr
Universität Bremen
24. April 1998

Literaturverzeichnis

[AHU87] A. Aho, J.E. Hopcroft, und J.D. Ullman. *Data Structures and Algorithms.* Addison-Wesley, 1987.

[Ehl97] J. Ehlich. *Simulation der Fahrleistungsreduktion durch Mitfahren im Schlanken Verkehr.* Universität Bremen, 1997. Diplomarbeit.

[Ert98] G. Ertl. Shortest path calculations in large road networks. *OR Spektrum*, 20:15–20, 1998.

[GG98] J.M. Golob und G. Giuliano. Smart Traveler Automated Ridematching Service Lessons Learned for Future ATIS Initiatives. *Transportation Research Records*, 1537:23–29, 1998.

[HM94] K. Haefner und G. Marte. *Der Schlanke Verkehr — Handbuch für einen umweltfreundlichen und effizienten Transport von Personen und Gütern.* Erich Schmidt Verlag, Berlin, 1994.

[HMH96] K. Haefner, G. Marte, und D. Heinisch. *Stadtgerechte Mobilität in Hannover — Eine langfristige Aufgabe der üstra.* Institut für Informatik und Verkehr, Universität Bremen, Dezember 1996.

[HRHS94] K. Haefner, C. Riester, C. Hilbig, und M. Sanders. *Der Schlanke Verkehr in der SV-Region Cottbus — Konzept zur Verkehrsvermeidung in der Region Cottbus.* Institut für Informatik und Verkehr, Universität Bremen, Dezember 1994.

[HU92] J.E. Hopcroft und J.D. Ullman. *Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie.* Addison-Wesley, 1992.

[HZ98] K. Haefner und L. Zegartowski. *Welche Reduktion kann durch Mitfahrten in einer Region erreicht werden?* Im Druck, 1998.

[Joh96] John A. Volpe Transportation Systems Center. *Advanced Public Transportation Systems: The State of the Art – Update '96*, Kendall Square, Cambridge, MA 02142, 1996. U.S. Departement of Transportation, Research and Special Programs Administration.

[Jun90] D. Jungnickel. *Graphen, Netzwerke und Algorithmen.* BI Wissenschaftsverlag, 1990.

[Sch92] H. Schittenhelm. *Ein effektiver wegorientierter Algorithmus zur Lösung des kombinierten Verkehrsverteilungs- und Umlegungsproblems.* Universität Bremen, 1992. Dissertation.

[Sed92] R. Sedgewick. *Algorithmen in C.* Addison-Wesley, 1992.

[Tur96] V. Turau. *Algorithmische Graphentheorie.* Addison-Wesley, 1996.