

Institut für Formale Methoden der Informatik

Abteilung Algorithmik

Universität Stuttgart
Universitätsstraße 38
D - 70569 Stuttgart

Masterarbeit Nr. 3622

Analyse von Algorithmen zur Bahnverbindungssuche

Mingyuan Wang

Studiengang:	INFOTECH
Prüfer:	Prof. Dr. Stefan Funke
Betreuer:	Prof. Dr. Stefan Funke / Dr. Martin Thust
begonnen am:	11.02.2014
beendet am:	13.08.2014
CR-Klassifikation:	G.2.2, I.1.2

Kurzfassung

Diese Arbeit liefert einen Einblick in die Thematiken der elektronischen Fahrplanauskunft und beschreibt den Aufbau eines einfachen Reiseplanungssystems. Vorteile des time-expanded- und time-dependent-Modells werden dabei kombiniert und ein dafür geeignet angepasster Dijkstra-Algorithmus vorgestellt. Des Weiteren beschäftigt sich die Arbeit mit der Analyse und dem Vergleich von HAFAS, einem gängigen Reiseplanungssystem, und dem Verbindungssuchalgorithmus von PRIMA, einem Nachfrageprognosemodell der Deutschen Bahn AG. Optimierungsvorschläge zum verbesserten Prognosepotenzial von PRIMA werden unterbreitet.

Inhaltsverzeichnis

Vorwort	1
1 Teil I: Implementierung eines einfachen Reiseplanungssystems für Züge	2
1.1 Einführung	2
1.1.1 Hintergrund und Nutzen des Reiseplanungssystems	2
1.1.2 Mathematische Grundlage	2
1.1.2.1 Datenstruktur des Verkehrssystems	2
1.1.2.2 Verbindungssuche	5
1.1.3 Kommerzielle Reiseplanungssysteme	7
1.1.4 Aufgabenstellung	9
1.2 Graphrepräsentation	12
1.2.1 Modellauswahl	12
1.2.2 Datenstruktur des Programms	13
1.2.2.1 Klassen	13
1.2.3 Datenimport aus Textdateien	16
1.3 Routenplanung	17
1.3.1 Dijkstra-Algorithmus	17
1.3.2 „earliest-arrival“-Suche	18
1.3.3 Fußwege	20
1.3.4 Umsteigeproblem	21
1.3.4.1 Erste Implementierung	22
1.3.4.2 Korrektur	23
1.3.5 „latest-departure“-Suche	25
1.4 Graphische Oberfläche	27
1.4.1 Anfrageeingabe	27
1.4.2 Routendarstellung	27
1.4.3 Visualisierung der Route	28
1.5 Experimentelle Evaluation	30
1.5.1 Basismodus	30
1.5.2 Speed-up	31
1.6 Zusammenfassung	32
1.6.1 Überblick	32
1.6.2 Verbesserungsmöglichkeiten	32
1.6.2.1 Fußweg	32
1.6.2.2 Rückwärtssuche	33

1.6.2.3	Fahrplandaten.....	33
2	Teil II: Wesentliche Unterschiede der Verbindungssuche PRIMA und HAFAS	35
2.1	Einführung	35
2.2	Dokumentation der Algorithmen.....	36
2.2.1	PRIMA-Verbindungssuche [9].....	36
2.2.2	HAFAS-Verbindungssuche.....	38
2.2.3	Weitere Kenntnisse aus den HAFAS-Rohdaten [11]	41
2.2.3.1	Markierungen und Sonderzug.....	41
2.2.3.2	Überschreibung von Mindestübergangszeiten	42
2.3	Unterschiede PRIMA/HAFAS-Verbindungssuche.....	44
2.3.1	Theoretische Unterschiede.....	44
2.3.2	Eigenschaften der Daten	44
2.3.3	Vergleichsvorgang	46
2.3.3.1	Nachfragestärkste EMS-Verbindungen.....	46
2.3.3.2	Nachfragestärkste PRIMA-Verbindungen	48
2.4	Aufnahme und Auswertung der Unterschiede.....	49
2.4.1	Ableitung von Top-EMS-Verbindungen ohne PRIMA-Eintrag.....	49
2.4.1.1	Segment I (BEFP > 10).....	49
2.4.1.2	Segment II (4 < BEFP ≤ 10)	52
2.4.2	Ableitung von Top-PRIMA-Verbindungen ohne EMS-Eintrag.....	61
2.4.3	Sonstige unterschiedliche Verbindungen.....	64
2.5	Zusammenfassung und Optimierungsvorschläge	67
2.5.1	Ergebnisse.....	67
2.5.2	Einzelbewertungen	68
	Quellennachweise	72

Vorwort

Diese Masterarbeit entstand in einer Zusammenarbeit zwischen dem Institut für Formale Methoden der Informatik an der Universität Stuttgart und der Abteilung Kunden-/Marktanalysen und -prognosen bei der DB Fernverkehr AG in Frankfurt.

Im Teil I dieser Arbeit werden die Grundlagen der Verbindungssuche im öffentlichen Verkehr sowie die zugehörigen Algorithmen vorgestellt. Nutzen und Herausforderungen der elektronischen Fahrplanauskunft werden dabei erläutert und ein einfaches Reiseauskunftssystem für den bundesweiten Personenschienenverkehr in Deutschland von Grund auf entwickelt.

Teil II beschäftigt sich mit weiteren Themen realistischer Bahnverbindungen. Eine Eins-zu-Mehr-Verbindungssuche aus einem DB-internen Prognosesystem zur Bestimmung der Nachfrageverteilung zu allen möglichen Verbindungen und eine Eins-zu-Eins-Verbindungssuche zur kommerziellen Reiseauskunft sowie Ticketausstellung werden dargelegt und anhand der jeweils gebildeten Mehr-zu-Mehr-Suchergebnisse verglichen. Der Vergleich beschränkt sich auf wesentliche Unterschiede, welche eine wirtschaftliche Bedeutung aufweisen.

Der Zusammenhang zwischen den beiden Teilen besteht darin, dass der technische erste Teil grundlegende Kenntnisse zur Analyse und Beurteilung von komplexen Verbindungssuchsystemen vermittelt.

1 Teil I: Implementierung eines einfachen Reiseplanungssystems für Züge

1.1 Einführung

1.1.1 Hintergrund und Nutzen des Reiseplanungssystems

Mobilität zählt zu den wichtigsten Bedürfnissen des Menschen im Alltagsleben und gilt als ein besonderer Einflussfaktor auf die Lebensqualität.

Eine Reise zwischen zwei Orten kann neben dem Fußweg entweder durch den **Individualverkehr** mit motorisierten und nicht-motorisierten Verkehrsmitteln oder durch die Nutzung von **öffentlichen Verkehrsmitteln** ermöglicht werden. Im Gegensatz zum Individualverkehr kann der Reisende bei der Nutzung des öffentlichen Verkehrs den Fahrweg sowie seine gewünschte Abfahrts- und Ankunftszeit nicht mehr beliebig gestalten, sondern er ist von einem Fahrplan abhängig, über den sich der Reisende vorab zu informieren hat. Die Haltepunkte des Fahrplans bestimmen die erreichbaren Ziele und die Zeitpunkte geben den möglichen Ein- oder Ausstieg bekannt. Besteht eine Reise zwangsläufig aus mehreren Fahrzeugen mit Umstieg oder sind unterschiedliche Reisewege vorhanden, so müssen alle Fahrplaninformationen geprüft und ggf. miteinander verglichen werden, um zur gewünschten Abfahrtszeit eine **Reisekette** herzustellen.

Um schnell und zielführend die notwendigen Reiseinformationen zu erhalten, können die Berechnungen zur Lösung dieses oft komplexen Sachverhalts von einem Computer durchgeführt werden. Ein **Reiseplanungssystem**, vielerorts auch Fahrplanauskunftssystem genannt, erfüllt diese Aufgabe, indem es anhand der Fahrplandaten von einem angegebenen Ausgangspunkt (*source*) zu einem angegebenen Zielort (*target*) die **schnellste Verbindung** ab einem angegebenen Zeitpunkt ermittelt.

1.1.2 Mathematische Grundlage

1.1.2.1 Datenstruktur des Verkehrssystems

Bevor jegliche Suche durchgeführt wird, kann es vorteilhaft sein, die Daten des Verkehrsnetzes in einer mathematischen Repräsentation abzubilden, um die Problematik gründlich zu analysieren und effiziente Maßnahmen anhand von theoretischen Algorithmen auszuwählen.

Die Konzepte der Graphentheorie findet hierzu Anwendung. Orte (**Knoten**) lassen sich mittels Wegen (**Kanten**) miteinander verknüpfen, wobei jeder Kante ein **Gewicht** zugeschrieben wird, das die vorgesehene Fahrzeit auf dieser Kante beschreibt. Zwei unmittelbar verbundene Knoten verfügen über eine Kante.

Nachstehende Teile dieser Arbeit beschränken sich auf den öffentlichen Verkehr, da die Themen Bezug auf die Bahnverbindungssuche nehmen. Um jedoch den Transport im öffentlichen Verkehr vom Individualverkehr prinzipiell zu unterscheiden, ist es wichtig, im Überblick einen Vergleich der beiden Verkehrssysteme zu ziehen.

- **Im Individualverkehr** sind ununterbrochene Teilwege die Kanten und Kreuzungen wo sich Kanten treffen die Knoten. Das Gewicht (der Zeitaufwand) einer Kante ist von deren Länge sowie Geschwindigkeit des Fahrzeuges abhängig.
- **Im öffentlichen Verkehr** hingegen lassen sich alle angeschlossenen Haltestellen, auch Stationen (*stations*) genannt, des gesamten Fahrplanangebots als Knoten widerspiegeln, während eine durch das Fahrzeug bestimmte Teilstrecke, die ausschließlich zwei Knoten verbinden, als eine Kante bezeichnet wird (Abbildung 1). Zwar hängt das Gewicht dieser fixen Kante auch von der Entfernung der Teilstrecke und durchschnittlicher Geschwindigkeit des genutzten öffentlichen Fahrzeuges ab, ist die Fahrzeit oft vom Anbieter vorberechnet und im Fahrplan vorgegeben.

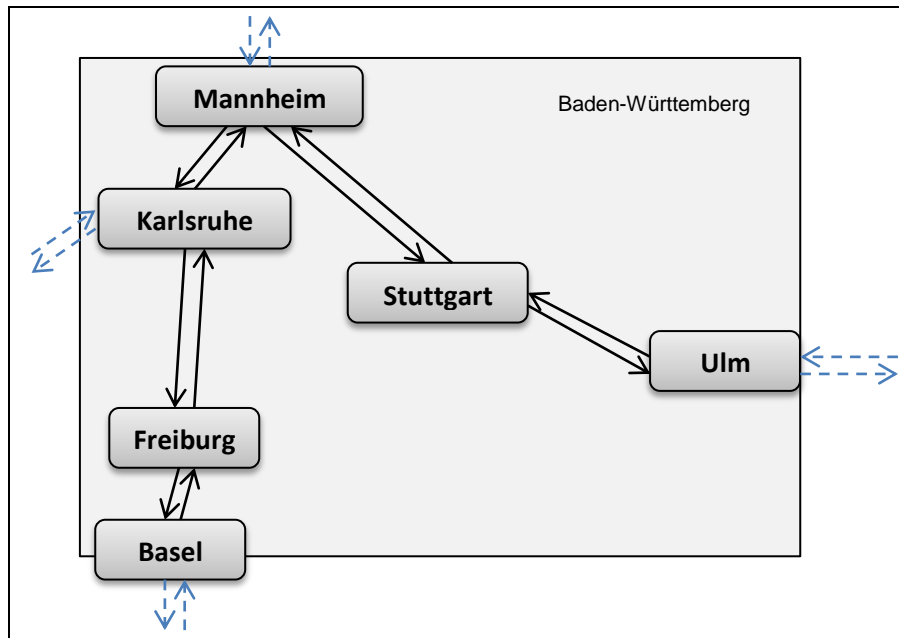


Abbildung 1 Vereinfachter Graph des ICE-Streckennetzes in Baden-Württemberg

Beim Verhalten des Fahrzeuges ist zwischen den zwei Verkehrskategorien zu unterscheiden. Da dem Reisenden im öffentlichen Verkehr an einer Station nur eine begrenzte Auswahl von Abfahrtszeiten zur Verfügung steht, ist eine Bewegung zwischen zwei Knoten auf dem Graphen nur unter bestimmten Zeitkriterien möglich.

Eine Verfeinerung der Graphdarstellung für den öffentlichen Verkehr ist daher notwendig, um eine genaue Routenplanung zu gewährleisten. Dabei kann eines der folgenden zwei Modelle für die Datenstruktur gewählt werden:

- Beim „**time-dependent**“-Modell [1] bleiben die Knoten unverändert, wobei dem Gewicht jeder Kante zwischen zwei Knoten eine mathematische Funktion zugeordnet wird, welche für die Eingabe einer konkreten Startzeit am ausgehenden Knoten die entsprechenden Zeitkosten (einschließlich Fahrzeit und ggf. Wartezeit) bis zum Erreichen des anderen Knotens ausgibt (Abbildung 2 rechts).
- Beim „**time-expanded**“-Modell [2] werden die Stationsknoten der vorherigen Beschreibung durch eine Reihe von zeitgebundenen Abfahrts- oder Ankunftsereignissen an dieser Station ersetzt. Jeder Knoten entspricht nun einem Zeitpunkt (*scheduled time*) an einer Station, in dem ein Fahrzeug hier ankommt (*arrival*) oder abfährt (*departure*). Jede Kante verbindet einen Abfahrtsknoten (*departure node*) einer Station A mit einem Ankunfts-knoten (*arrival node*) einer anderen Station B (Abbildung 2 links).

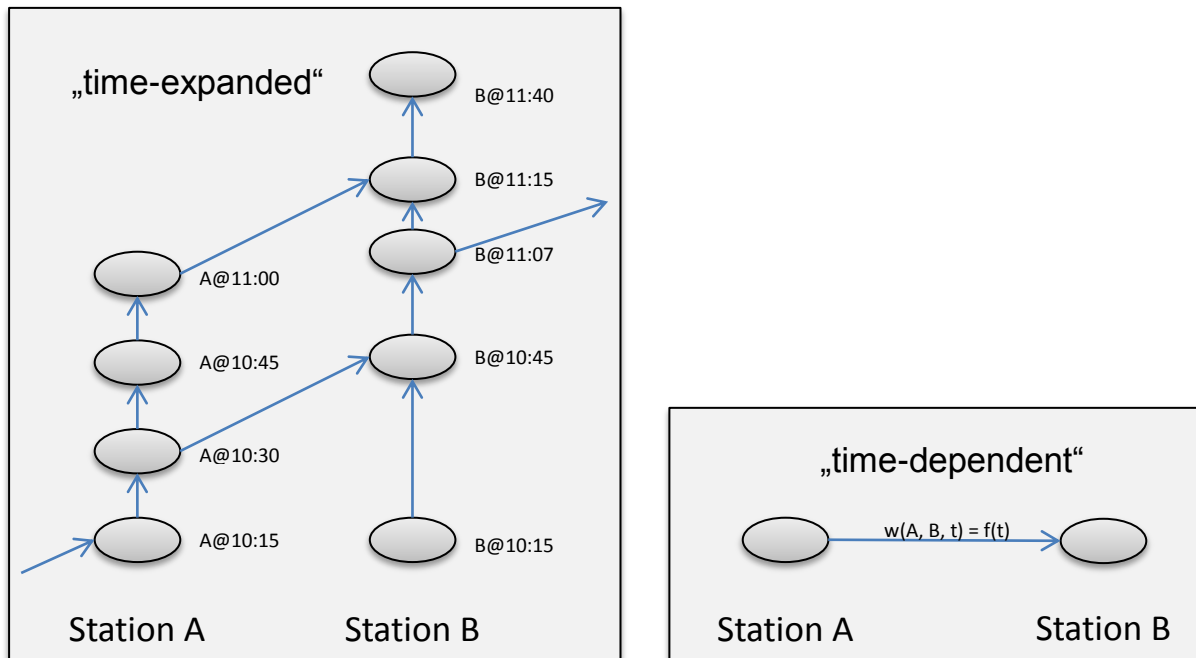


Abbildung 2 Vergleich „time-dependent“- und „time-expanded“-Modell

1.1.2.2 Verbindungssuche

Nachdem das auf dem öffentlichen Verkehr basierenden Verkehrsnetz aus den Fahrplandaten in einen strukturierten Graphen umgewandelt wurde, kann eine Verbindungssuche stattfinden.

- *Begriffsdefinitionen:* Als **Verbindung** zwischen zwei Orten bezeichnet man eine fahrplanabhängige Beförderungsmöglichkeit, die anhand öffentlicher Verkehrsmittel einen konkreten und bestimmten Weg zurücklegt und eine planmäßige Abfahrts- und Ankunftszeit vorgibt. Benutzte Fahrzeuge und erforderliche Fußwege können zu einer Verbindung gehören. Dabei dürfen unterschiedliche Haltestellen an- und wieder abgefahren werden. Die Verbindung entspricht einem **Pfad** auf dem Graphen. Teilwege auf einer Verbindung werden **Teilverbindungen** genannt. Zudem definiert die vorliegende Arbeit eine Kante zwischen zwei unmittelbar verbundenen Knoten als **Primärverbindung**, die kleinste Einheit eines Fahrwegs. Alle (Teil-)Verbindungen bestehen ausschließlich aus Primärverbindungen und Fußwegen. Eine **Fahrt** ist eine spezielle Verbindung, die mittels eines selben Fahrzeuges aus dem öffentlichen Verkehr absolviert wird und im Regelfall keinen Umstieg vorsieht. Alle Primärverbindungen einer Fahrt tragen deshalb eine gleiche **Fahrtnummer**. Daraus lässt sich eine **Betriebsbezeichnung** auslesen.

Das Ziel einer Reise ist in den meisten Fällen die Bewegung von einem Ort zu einem anderen Ort. Wenn ein normaler Reisender nach einer Verbindung sucht, ist es vernünftig davon auszugehen, dass er mit den **geringsten Kosten** an seinen Zielort kommen möchte.

Kosten in erweitertem Sinne umfassen nicht nur ökonomische Konsequenzen, sondern auch Zeitaufwand, Risiko, reduzierten Komfort während der Reise oder sonstige ausfallende Ansprüche. Für einen durchschnittlichen Menschen ist eine gewichtete Summe dieser Kosten entscheidend. D.h. wenn der Zeitaufwand als Basis betrachtet wird, kann alles andere in Reisezeit umgerechnet werden. Da sich die Bedürfnisse der Reisenden aber deutlich unterscheiden, können die Faktoren der jeweiligen Kosten nur schwer bestimmt werden. Im idealen Fall kann man die einfache Annahme treffen, dass alle möglichen Reiseverbindungen, abgesehen von der Reisezeit, konstante weitere Kosten haben, weil die geringste Reisezeit für eine Auswahl meistens ausschlaggebend ist.

Die Suche nach den schnellsten Verbindungen im Verkehrsnetz entspricht dieser Annahme nach einer erweiterten Kürzeste-Pfad-Suche auf dem abgebildeten Graphen. 1959 schlug E. W. Dijkstra hierfür einen Algorithmus vor, heute **Dijkstra-Algorithmus** genannt, den er in der zweiten Problemstellung seiner Veröffentlichung [3] ausführlich beschrieb. Übertragen auf das oben bereits definierte Modell kann der Algorithmus wie folgt interpretiert werden:

- *Zielsetzung*: Von einem Knoten P aus werden Pfade mit der niedrigsten Summe aller befahrenen Kanten zum Knoten Q gesucht.

- Dijkstras Lösung in übertragenem Sinne:

1. *Nehme und aktiviere Knoten P, setze seinen Kostenwert auf 0 und untersuche alle von P aus mit einer Kante erreichbaren Knoten.*
2. *Rechne die Kostenwerte aller im Schritt 1 untersuchten Knoten und speichere diese Knoten (mit Kostenwert und deren Vorgängerknoten P) in eine Liste X.*
3. *Sperre den letzten genommenen Knoten und aktiviere den Knoten R aus der Liste X mit dem geringsten Kostenwert (zu P).*
4. *Untersuche alle von R aus mit einer Kante erreichbaren Knoten und berechne deren (über R) erreichte Kostenwerte in Bezug auf P.*
5. *Füge nur diejenigen im Schritt 4 untersuchten Knoten (mit neuem Kostenwert) zur Liste X hinzu, die hierdurch einen geringeren Kostenwert erlangen oder wenn ihr Kostenwert noch nicht bekannt ist.*
6. *Gehe zu Schritt 3.*

Sobald Q als R im Schritt 3 identifiziert wird, ist die Suche zu Ende und die Schleife kann unterbrochen werden. Die geringsten Kosten von P zu Q werden dadurch bekannt. Der kürzeste Pfad, der dies ermöglicht, kann anhand der gespeicherten Vorgängerknoten aller während der Laufzeit aktivierten Knoten ermittelt werden. Sollte die Schleife vorzeitig beendet sein, weil die Liste X bereits leer ist, lässt sich daraus folgern, dass der Knoten Q von P aus nicht erreicht werden kann.

Die Richtigkeit des Algorithmus kann formal bewiesen werden [4], solange die Kosten (Gewichte) aller Kanten nichtnegativ sind. Dem ursprünglichen Ziel, die schnellste Verbindung von A nach B ab dem Zeitpunkt t zu erhalten, wird dieser Algorithmus nicht notwendigerweise gerecht, da er lediglich die Verbindung mit der frühesten Ankunft in B, **earliest-arrival** genannt, findet.

1.1.3 Kommerzielle Reiseplanungssysteme

Erste elektronische Reiseplanungssysteme wurden in den achtziger Jahren ins Leben gerufen [2]. Seitdem existieren auf dem Markt zur kommerziellen Verwendung verschiedene Programme, die später auch als Online-Versionen verfügbar wurden. Die in Deutschland verbreitetsten und erfolgreichsten Reiseplanungssysteme sind HAFAS der Firma *HaCon* und EFA von *Mentz Datenverarbeitung*, die sich in jeweils unterschiedlichen Versionen bei ihren Bestellern im schwerpunktorientierten Einsatz befinden.

HAFAS (HaCon Fahrplanauskunftssystem) wird überwiegend von europäischen Eisenbahnunternehmen verwendet. Die bekannteste Version von HAFAS ist das Online-Suchportal auf dem Internetauftritt der Deutschen Bahn AG (Abbildung 3). Es erhält täglich über 20 Millionen Verbindungsabfragen [5] und erlaubt eine Multikriterien-Suche, in der nicht nur die zeitschnellsten sondern auch „komfortable“ oder „preiswerte“ Verbindungen gefunden werden. In einem erweiterten Modus Reisenden-Informationen-System (RIS) werden die Fahrplanzeiten gemäß aktueller Ortung des Fahrzeuges mit berechneten Echtzeit-Werten überschrieben und nur die aktualisierten Suchergebnisse ausgegeben. Im zweiten Teil dieser Arbeit (Kapitel 2.2.2) werden das Suchverhalten und die Suchergebnisse von HAFAS ausführlich beschrieben.

The screenshot shows the DB HAFAS search interface. On the left, the search criteria are defined: Start at Stuttgart Hbf, Ziel at Frankfurt(Main)Hbf, and the date/time is set for a departure on Di, 19.08.14 at 06:58. The search results on the right show a table of connections with columns for Bahnhof/Haltestelle, Datum, Zeit, Dauer, Umst., Produkte, and Preis für alle Reisenden*. The table lists several options, including direct ICE services and services with one transfer (IC, RE).

Bahnhof/Haltestelle	Datum	Zeit	Dauer	Umst.	Produkte	Preis für alle Reisenden*
Stuttgart Hbf Frankfurt(Main)Hbf	Di, 19.08.14	ab 06:51 an 08:06	1:17	0	ICE	55,00 EUR 63,00 EUR
Stuttgart Hbf Frankfurt(Main)Hbf	Di, 19.08.14	ab 07:14 an 09:24	2:10	1	IC, RE	19,00 EUR 47,00 EUR
Stuttgart Hbf Frankfurt(Main)Hbf	Di, 19.08.14	ab 07:26 an 08:53	1:27	0	ICE	45,00 EUR 63,00 EUR
Stuttgart Hbf Frankfurt(Main)Hbf	Di, 19.08.14	ab 07:26 an 09:04	1:38	1	ICE	45,00 EUR 63,00 EUR

Abbildung 3 DB-HAFAS-Verbindungssuche [6]

F. Schulz erwähnt in seiner Dissertation [7], dass ARIADNE, der Vorgängeralgorithmus von HAFAS, eine zweistufige Suche durchführt. Durch die statische „Wegesuche“ auf einem rein topographischen Bahnnetz unter Absehung von Zeitbezug wird der Suchbereich der zweiten Stufe auf einen angeblich relevanten Teilgraphen beschränkt. Dieser Teilgraph schließt die Zeitdaten ein und leitet ein time-dependent-Modell ab, auf dem die „Zeitsuche“ nach der frühesten Ankunft stattfindet.

Ein weiteres populäres Reiseplanungssystem EFA (Elektronische Fahrplanauskunft) hält auch einen großen Marktanteil und kommt häufig bei regionalen sowie regionalübergreifenden Verkehrsverbänden zum Einsatz. Hier können bestimmte Tarifzonen an- oder abgewählt werden und ein

Höhenprofil für die hinterlegte Karte ist zusätzlich im System eingepflegt. So werden beispielsweise Fußwege realistisch bewertet und eine sehr präzise Tür-zu-Tür-Suche angeboten. Der Verkehrs- und Tarifverbund Stuttgart (VVS) verwendet eine Version von EFA zur Verbindungssuche in der städtischen Reiseauskunft (Abbildung 4). Für kurze Reisen werden auch reine Fußwege als Alternative vorgeschlagen, falls diese mit einer voraussichtlichen Gehzeit von bis zu 30 Minuten absolviert werden können. VVS-EFA zeigt die Reisezeit auf dem öffentlichen Verkehrsmittel und die Gesamtzeit getrennt an.

Die Datenstruktur sowie algorithmische Methode dieses Systems wurden nicht veröffentlicht.

Fahrplanauskunft

Von: Stuttgart / Universität
Nach: Stuttgart / Berliner Platz (Liederhalle)
Wann: 15.08.2014, 08:13 Uhr
Anfordern

Weitere Einstellungen (Über, zeit kürzeste Verbindung, Verkehrsmittel, Fußwege und Steigung konfigurieren, Umsteigen, ...)

Verkehrsmittel:
 R-Bahn
 S-Bahn
 Stadt-/Zahnradbahn
 Bus
 Nachtbus
 Ruftaxi
 Sonstige (Bürgerbus, Museumsverkehr)

Über: Haltestelle
Aufenthalt: 0 min

Umstieg an dieser Haltestelle ausschließen: Haltestelle

Verbindungsoptionen:
 Zeit kürzeste
 wenig Umsteigen
 kurze Fußwege
 Preis günstigste
 Fahrradmitnahme

Tarifzonen berücksichtigen | alle | keine
 10 20 30 31 32 34 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 77 81
 91

Fahrtempfehlung

Abfahrt: 08:13 am Fr, 15.08.2014
 von: Stuttgart / Universität
 nach: Stuttgart / Berliner Platz (Liederhalle)

Optionen: Standardeinstellungen
 Ändern Rückfahrt Weiterfahrt Neue Fahrt

Fahrt	Abfahrt	Ankunft	Linien	Gesamtzeit	ÖV Fahrzeit	Karten	Preis	Tarif
1	08:04	08:23	S3, 43	00:19	00:19	Karte	2,70 €	0,00 €
2	08:14	08:33	S1, 43	00:19	00:19	Karte	2,70 €	0,00 €
3	08:19	08:39	S3, U14	00:20	00:20	Karte	2,70 €	0,00 €
4	08:24	08:43	S2, 43	00:19	00:19	Karte	2,70 €	0,00 €

Später Letzte Fahrt Ticketberater | Preisvergleich Auto - VVS | Empfehlen | Höhenprofil | Drucken

1. Fahrt Empfehlen | Kalendereintrag | Drucken | Kaufen/Tarif

Früher ab 08:04 an 08:13
S-Bahn S3 Richtung Backnang
 Universität Gleis 2
 Stadtmittelpunkt Gleis 2
 Bad Cannstatt - Waiblingen: Umleitungen und Fahrplanänderungen wegen Bauarbeiten

Später ab 08:22 an 08:23
Bus 43 Richtung Killesberg
 Niederflerbus
 Rotebühlpl. (Stadm.) Pos. 1
 Berliner Platz (Liederhalle)
 Stuttgart: Montags Behinderungen auf den Innenstadtbustlinien

Abbildung 4 VVS-EFA-Verbindungssuche [8]

1.1.4 Aufgabenstellung

Ziel für Teil I dieser Arbeit ist es, anhand einer gängigen Programmiersprache ein einfaches Reiseplanungssystem zu implementieren, welches auf den Fahrplandaten für ganz Deutschland (einschließlich dem kompletten schienengebundenen Personalnahverkehr sowie dem bundeseigenen schienengebundenen Personalfernverkehr) aus einem Fahrplanjahr basiert.

Zum Erreichen dieses Ziels wird die Arbeit in folgende Aufgabenblöcke eingeteilt und ebenso gliedern sich die kommenden Kapitel:

- Aufbereitung der Fahrplandaten der DB AG in eine geeignete Graphrepräsentation (z.B. time-expanded).
- Implementierung und Anpassung des Suchverfahrens, wie z.B. Dijkstra, welches auf der entsprechenden Graphrepräsentation operiert.
- Erweiterung des Programms um eine einfache GUI zur Spezifikation von Anfragen und Visualisierung des Suchergebnisses.

Dabei muss die Umsetzung weitere Anforderungen erfüllen:

- den kompletten Fahrplandatenbestand auf einem Standard-Desktop-PC im Hauptspeicher (16GB RAM) halten können,
- Anfragen maximal im Sekundenbereich beantworten,
- zunächst „earliest-arrival“-Verbindungen ermitteln, und darauf beruhend realistische Verbindungen unter Berücksichtigung von minimalen Umsteigeübergangszeiten sowie möglichem Fußweg.

Bereitgestellte Daten:

- umfassen drei Textdateien „*dbStops*“, „*dbTrips*“ und „*dbConns*“,
- stammen aus dem Fahrplanjahr 2012 (gültig vom 11.12.2011 bis zum 08.12.2012),
- enthalten den gesamten Schienenpersonenverkehr der DB AG (samt Nah- und Fernverkehr) sowie den Nahverkehr (mit kleinen Ausnahmen und Abweichungen) der nichtbundeseigenen Eisenbahnen in Deutschland an allen Verkehrstagen.

Eine ausführliche Datenstruktur dieser Textdateien kann der folgenden Tabelle 1 entnommen werden.

Tabelle 1 Struktur der bereitgestellten Textdateien

Datei „dbStops“ (6.919 Zeileneinträge)	
Feldbezeichnung	Feldfunktionsbeschreibung
<i>stopID</i>	vergebene Stationsnummer
<i>stopLongitude</i>	Längengrad der Station
<i>stopLatitude</i>	Breitengrad der Station
<i>stopName</i>	offizieller Name (eventuelle Leerzeichen bzw. Umlaute im Namen durch „_“ bzw. „a“, „u“, „o“ und „s“ ersetzt)
<i>stopChangeTime</i>	minimale Umsteigeübergangszeit in Sekunden
<i>stopOwner</i>	Betreibername der Station
Datei „dbTrips“ (132.994 Zeileneinträge)	
Feldbezeichnung	Feldfunktionsbeschreibung
<i>tripID</i>	vergebene Fahrtnummer
<i>tripName</i>	offizielle Betriebsbezeichnung der Fahrt
<i>tripVehicle</i>	Gattungsbezeichnung des Fahrzeuges
<i>tripOwner</i>	Betreiber der Fahrt
<i>tripLine</i>	gebündelte Liniennummer gleichen Fahrwegs
Datei „dbConns“ (1.514.874 Zeileneinträge)	
Feldbezeichnung	Feldfunktionsbeschreibung
<i>connID</i>	vergebene Primärverbindungsnummer
<i>connDepStop</i>	<i>stopID</i> der Abfahrtsstation aus „dbStops“
<i>connArrStop</i>	<i>stopID</i> der Ankunftsstation aus „dbStops“
<i>connDepTime</i>	in Sekunden umgerechnete Abfahrtszeit an <i>connDepStop</i>
<i>connArrTime</i>	in Sekunden umgerechnete Ankunftszeit an <i>connArrStop</i>
<i>connTripID</i>	zugehörige Fahrtnummer <i>tripID</i> aus „dbTrips“

1.2 Graphrepräsentation

In diesem Kapitel werden die Datengrundlagen und Struktur des aufzusetzenden Reiseplanungssystems erstellt.

Zunächst wird ein geeignetes Graphmodell gewählt, dann erfolgt eine Erläuterung zur Struktur des Programms. Darauf beruhend können Daten aus den Textdateien in den Graphen eingelesen werden.

1.2.1 Modellauswahl

In 1.1.2.1 wurden zwei mögliche Modelle zur Umwandlung eines öffentlichen Verkehrsnetzes in einen mathematischen Graphen vorgestellt.

Das „time-dependent“-Modell bietet den Vorteil, dass der Graph aufgrund der nicht sehr großen Anzahl von Knoten (Stationen) für die spätere Suche eindeutig klein gehalten wird. Dafür muss aber auf jeder Kante eine gültige Funktion hinterlegt werden, die einen tagesuhrzeitabhängigen Wert zur Bestimmung des Kantengewichts ausliefert. Dementsprechend muss diese Funktion alle aus- und eingehenden Planzeiten berücksichtigen und stellt damit eine hohe Komplexität und zusätzlichen Rechenaufwand dar.

Das „time-expanded“-Modell weicht diesem Problem aus, indem es die Abfahrts- und Ankunftszeiten bereits bei dem Aufbau des Graphen einbezieht und aus jeder Station alle ihre Abfahrts- und Ankunftsevents als Knoten ableitet. Da die Kantengewichte nun vorberechnet werden können, lässt sich die Kürzeste-Wege-Suche aus der Graphentheorie mit geringfügiger Anpassung direkt anwenden. Dies hat allerdings zur Folge, dass nicht nur jeder Knoten sondern auch jede Kante aus dem „time-dependent“-Modell vervielfacht wird, auch wenn sie die absolut gleiche Station oder den absolut gleichen Fahrweg repräsentiert.

Einer ersten Analyse der bereitgestellten Daten zu entnehmen, ergibt sich die Speicherung der Primärverbindungen in der Textdatei „*dbConns*“ wie in der Abbildung 5.

connID	connDepStop	connArrStop	connDepTime	connArrTime	connTripID
...
2267	239434	240630	240	1620	1101
2268	239422	241165	240	2040	1102
2269	239422	241165	240	2040	1103
2661	241379	241382	270	300	600
2662	243255	243256	270	300	880
2762	240300	241353	270	360	746
...

Abbildung 5 Auszug von 6 Zeilen aus der Textdatei „dbConns“

Die 1.514.874 Primärverbindungen werden erst nach aufsteigender Abfahrtszeit und dann nach aufsteigender Ankunftszeit sortiert. Das vorhandene Format befähigt direktes Einlesen von Knoten und Kanten nach Definition des „time-expanded“-Modells. Da das Parsen von Textdateien einen sehr hohen Rechenaufwand beansprucht, ist eine effiziente Verwendung des Parsers zu bevorzugen. Die vorliegende Arbeit baut den Graphen deshalb auf einem „time-expanded“-Modell auf.

1.2.2 Datenstruktur des Programms

Das Programm wird in der objekt-orientierten Programmiersprache C++ in einer Linux-Umgebung entwickelt. Als Unterstützung wird überwiegend auf die Datentype und Funktionen der C++-Standardbibliothek zurückgegriffen. Für die graphische Benutzeroberfläche verwendet das Programm zusätzlich Klassen aus der Qt-Bibliothek.

Im Folgenden werden die selbstdefinierten Klassen sowie deren Attribute und Methoden vorgestellt, die für die Erzeugung des Graphen in einem „time-expanded“-Modell benötigt werden. Eine nähere Erläuterung der auf dem Dijkstra-Algorithmus basierenden Suchfunktionen erfolgt in dem nächsten Kapitel.

1.2.2.1 Klassen

a. Station

Station
<code>_stationId: int</code> <code>_stationName: String</code> <code>_minTransferTime: int</code> <code>_longitude: double</code> <code>_latitude: double</code>
+ <code>getStationId(): int</code> + <code>getStationName(): String</code> + <code>getMinTransferTime(): int</code> + <code>getStationLongitude(): double</code> + <code>getStationLatitude(): double</code>

Die Klasse *Station* erhält alle Informationen der entsprechenden Station. Eine Trennung dieser Klasse von den zugehörigen Knoten ermöglicht ein effizientes Einlesen und ist für das Debuggen besonders hilfreich.

b. *Node*

Node
<code>_stationId: int</code> <code>_nodeType: String</code> <code>_timeDepArr: int</code>
+ <code>getStationId(): int</code> + <code>getNodeType(): String</code> + <code>getScheduleTime(): int</code>

Die Klasse *Node* repräsentiert einen Knoten und beinhaltet die Stationsnummer, den Knotentyp („*DEP*“ für einen Abfahrtsknoten und „*ARR*“ für einen Ankunfts-knoten) sowie eine in Sekunden umgerechnete planmäßige Abfahrts- bzw. Ankunftszeit.

c. *Arc*

Arc
<code>_headNode: Node</code> <code>_cost: int</code> <code>_tripld: int</code>
+ <code>getHeadNode(): Node</code> + <code>getCost(): int</code> + <code>getTripld(): int</code>

Die Klasse *Arc* bezeichnet eine gerichtete Kante und speichert neben Gewicht und Fahrnummer nur einen **Kopfknoten**, den Ankunfts-knoten, jedoch nicht den Abfahrtsknoten. Sie wird später in Verbindung mit einer Adjazenzliste zusammen verwendet, in welcher jedes Element einen einzigartigen Abfahrtsknoten symbolisiert, welchem ferner eine Liste von *Arcs* zugeordnet wird.

d. *Trip*

<i>Trip</i>
<code>_tripId: int</code> <code>_tripName: String</code>
+ <code>getTripId(): int</code> + <code>getTripName(): String</code>

Klasse *Trip* wird zur Übersetzung einer Fahrnummer in die offizielle Betriebsbezeichnung (z.B. *ICE_513*) des Zuges verwendet.

e. *TrainNetwork*

<i>TrainNetwork</i>
<code>_stations: map<int, Station></code> <code>_nodes: map<int, Node></code> <code>_trips: map<int, Trip></code> <code>_adjacentArcs: map<Node, vector<Arc> ></code> <code>_stationDepNodesMap: map<int, vector<Node> ></code> <code>_stationNameIdMap: map<String, int></code> <code>_walkableStationsFromStations: map<int, vectpr<pair<int,int> > ></code> ...
- <code>addStation(): void</code> - <code>addNode(): void</code> - <code>addTrip(): void</code> + <code>getStations(): map<int, Station></code> + <code>readFromDBFiles(stopsFile, connsFile, tripsFile): void</code> + <code>computeShortestPath(sourceName, targetName, startTime): int</code> - <code>convertSecToTime(seconds): String</code> ...

Die Klasse *TrainNetwork* ist der zusammengestellte Graph des Verkehrsnetzes. ***_stations***, ***_nodes***, ***_trips*** und ***_adjacentArcs*** sind jeweils Sammlungslisten von den gesamten Stationen, Knoten, Fahrten und Kanten.

Über ***_stationDepNodesMap*** kann auf alle Abfahrtsknoten an einer bestimmten Station zugegriffen werden und ***_stationNameIdMap*** ermöglicht

zusätzlich die Identifizierung einer Station durch die Nennung des Bahnhofsnamens.

Zur Operation auf dem Graphen außerhalb der Klasse verfügt *TrainNetwork* über die Funktionen ***readFromDBFiles(...)*** für den Datenimport und ***computeShortestPath(...)*** zur Umsetzung einer Kürzeste-Wege-Suche.

1.2.3 Datenimport aus Textdateien

Zur Bildung des Graphen wird die Methode *readFromDBFiles(...)* eines *TrainNetwork*-Objekts aufgerufen. Mithilfe eines Csv-Parsers werden Stations-, Primärverbindungs- und Fahrtdaten aus den Textdateien über die Funktionen *addStation()*, *addNode()* und *addTrip()* in den Graphen eingelesen und in die jeweilige Sammlungsliste eingefügt.

Daten aus „*dbStops*“ und „*dbTrips*“ können ohne weitere Änderung eins zu eins übernommen werden. Zur Erstellung einer Adjazenzliste *_adjacentArcs* in der Klasse *TrainNetwork* sind hingegen anstehende Knoten in der Datei „*dbConns*“ bereits während des Imports auf eine Prüfung angewiesen, da solch ein Knoten auch mehrfach vorkommen kann, z.B. gleichzeitige Abfahrt zweier Züge aus einem Bahnhof.

Auf folgende Weise wird sichergestellt, dass jeder Eintrag in *_adjacentArcs* nur einfach vorkommt und mehrfach auftretende Abfahrtszeiten darin gebündelt werden.

```
void TrainNetwork::readFromDBFiles(...)
...
1 // Abschnitt Parser liest in Datei „dbConns“
2 while Parser nicht am Ende
3     Lese die Zeile und extrahiere Abfahrts- und Ankunfts-knoten
4     Erstelle ihre Kante Arc
5     if neuer Abfahrtsknoten
6         Füge neues Element in _adjacentArcs ein
7         Hänge Arc an dieses Element an
8     else
9         Finde in _adjacentArcs das Element zum Abfahrtsknoten
10        Hänge Arc an dieses Element an
...
```

1.3 Routenplanung

1.3.1 Dijkstra-Algorithmus

Wie bereits in 1.1.2.2 erklärt, kann die Suche nach den kürzesten Pfaden auf dem Graphen nach dem Dijkstra-Algorithmus erfolgen. Der genaue Ablauf des Algorithmus lässt sich wie folgt im Pseudocode beschreiben.

Die Funktion namens *Funktion_Dijkstra* sucht vom Startknoten *sourceNode* zum Endknoten *targetNode* den kürzesten Pfad und errechnet den Wegeaufwand zwischen den beiden Knoten.

- *PQ*: Prioritätswarteschlange zur Speicherung von Knoten mit bereits bekannter kürzester Entfernung zu *sourceNode*.
- *distance[]*: Array zur einzelnen Speicherung dieser bekannten kürzesten Entfernungen zu *sourceNode*.
- *previousNode[]*: Array zur einzelnen Speicherung des Vorgängerknotens von Knotens mit bereits bekannter kürzester Entfernung zu *sourceNode*.
- *vTestDistance*: temporäre Speicherung der alternativen Entfernung eines auszuwertenden erreichbaren Knotens.

```
Funktion_Dijkstra (Node sourceNode, Node targetNode)
1  priority_queue PQ
2  for jeder Knoten i
3      distance[i] = ∞
4      previousNode[i] = leer
5  distance[sourceNode] = 0
6  Füge sourceNode zu PQ hinzu
7  while PQ nicht leer
8      u = Knoten aus PQ mit geringstem Wert in distance[]
9      Lösche u aus PQ // u wird aktiviert
10     if u == targetNode
11         break // Endknoten gefunden
12     for jeder von u aus unmittelbar erreichbare Knoten v
13         vTestDistance = distance[u] + Gewicht der Kante (u,v)
14         if vTestDistance < distance[v]
15             Füge v zu PQ hinzu
16             distance[v] = vTestDistance
17             previousNode[v] = u
18 if distance[targetNode] != ∞
19     return 1
```

Eine **Prioritätswarteschlange** wird zwecks effizienter Ermittlung des Knotens mit geringster Entfernung zu *sourceNode* (Zeile 8) eingerichtet. Sie erhält während der Laufzeit Elemente, die zur Bestimmung des kürzesten Pfades notwendig sind (Zeile 15). Intern verfügt sie über eine Baumstruktur und ordnet neu hinzugefügte Elemente nach ihrer Priorität, in diesem Fall dem Entfernungswert, und zieht das Element höchster Priorität zur Baumspitze vor.

Diese Vorgehensweise wirkt als Grundfaden der nachstehenden Suchalgorithmen und wird anpassend erweitert.

1.3.2 „earliest-arrival“-Suche

Die „earliest-arrival“-Suche strebt ein Prinzip an, das zum Erreichen des Zielknotens versucht, auch alle Zwischenknoten frühestmöglich zu erreichen. Dadurch dass jeder berührte Knoten vom Startknoten mit kleinstem Aufwand erlangt wird, kann auch sichergestellt werden, dass der Zielknoten darin fällt. Dieses Konzept wird als **Greedy-Prinzip** bezeichnet.

Zwar ist der „time-expanded“-Graph bereits im Kapitel 1.2 aufgestellt, fehlen die Kanten zwischen Ankünften und Abfahrten innerhalb einer Station, weil sich ein Fahrzeug zum Ein- und Ausstieg an der Station eine vorgegebene Zeit aufhalten könnte. Darüber hinaus soll das Abwarten zwischen Abfahrten zwecks Umstiegs ermöglicht werden, da die optimale Abfahrt nicht unbedingt die nächstliegende ist, z.B. wenn Letztere in eine andere Richtung führt.

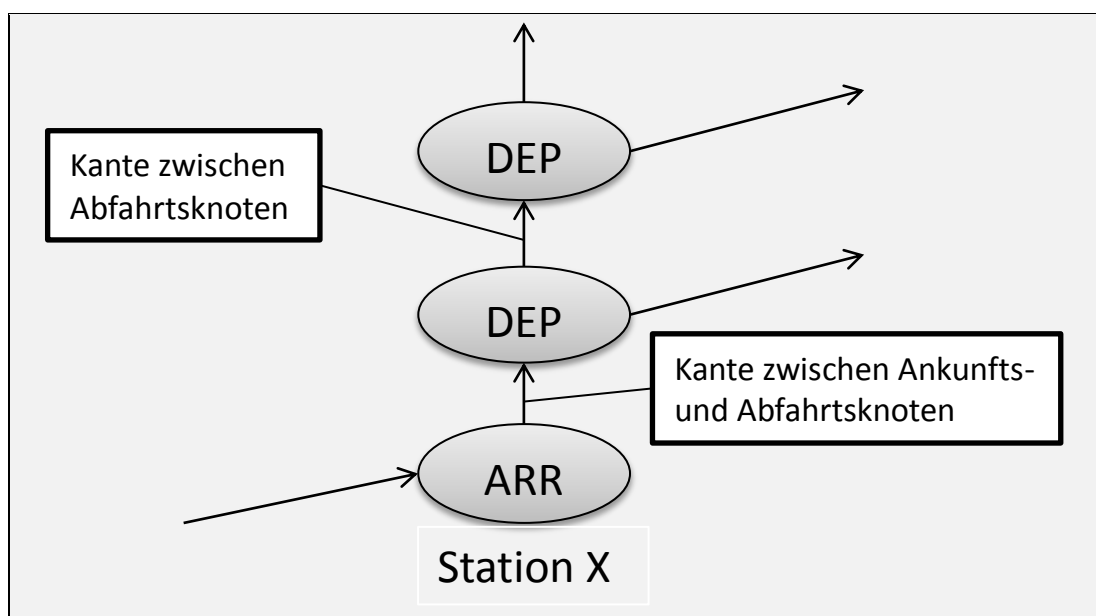


Abbildung 6 Kanten innerhalb einer Station

Diese zwei Arten von Kanten (siehe Abbildung 6) müssen gesondert konstruiert werden, da kein natürlich existierender Zusammenhang zwischen ihnen besteht.

Im klassischen „time-expanded“-Modell werden alle Abfahrts- und Ankunfts-knoten im Voraus zusammengeknüpft und während der Laufzeit des Suchalgorithmus unabhängig voneinander auf die Prioritätswarteschlange geschoben. Folglich ist die Länge dieser Warteschlange besonders groß, was dann schließlich dafür verantwortlich ist, dass die Anzahl der Iteration in der Schleife in einem Worstcase-Szenario potenziell hoch sein kann.

Diese Arbeit verwaltet im Gegensatz dazu in der Prioritätswarteschlange nur Stationen. Begründung dafür, ist dass der Graph *TrainNetwork* insgesamt 3.029.748 Knoten beinhaltet, im Vergleich zu lediglich 6.919 Stationen. Die Durchbindung zwischen Knoten innerhalb einer Station wird während der Laufzeit ermöglicht und deren Gewicht liefern betroffene Knoten selbst.

Der Dijkstra-Algorithmus kann wie folgt angepasst werden.

```
Earliest_Arrival (Station sourceStation, Station targetStation, int startTime)
1  priority_queue PQ
2  for jede Station i
3      distance[i] = ∞
4      previousStation[i] = leer
5  distance[sourceStation] = startTime
6  Füge (sourceStation, startTime) zu PQ hinzu
7  while PQ nicht leer
8      u = Station aus PQ mit geringstem Wert in distance[]
9      Lösche u aus PQ // u wird aktiviert
10     if u bereits einmal aus PQ gezogen
11         continue // verwerfe dieses alte Update
12     if u == targetStation
13         break // Endstation gefunden
14     for jeder Abfahrtsknoten m an Station u ab distance[u]
15         for jeder über m erreichbare Ankunfts-knoten n
16             v = angehörige Station des Ankunfts-knotens n
17             vTestDistance = distance[u] + cost(m,n)
18             if vTestDistance < distance[v]
19                 Füge (v, vTestDistance) zu PQ hinzu
20                 distance[v] = vTestDistance
21                 previousStation[v] = u
22 if distance[targetStation] != ∞
23     return 1
```

1.3.3 Fußwege

Fußweg ist ein Teilabschnitt der Reise, den der Reisende zu Fuß zurücklegen kann oder muss, um zwischen zwei nahliegenden Stationen (effizient) zu bewegen. Dies erfolgt in den meisten Fällen im Rahmen eines Umstiegs oder wenn die Reise an einer nicht durch das benutzte Fahrzeug bedienten Station starten oder enden soll.

Fußwege zwischen Stationen kann man theoretisch vermeiden, indem man auf Umwege ausweicht und dabei eine Drittstation als gemeinsamen Halt zweier Fahrzeuge benutzt. Voraussetzung dafür sind meistens ein zusätzlicher Umstieg und zeitlicher Mehraufwand.

Stuttgart_Hbf und *Stuttgart_Hbf_(tief)* sind beispielsweise zwei eigenständige Stationen. *Stuttgart_Hbf* wird überwiegend durch Fernzüge bedient, während in *Stuttgart_Hbf_(tief)* ausschließlich S-Bahnen anhalten. Ein Durchgang zwischen den beiden Stationen kann je nach Person und Gleis zwischen 2 - 7 Minuten absolviert werden. Möchte man auf diesen Übergang verzichten, kann man von *Stuttgart_Hbf* mit einem Regional-Express bis *Stuttgart-Bad_Cannstatt* fahren und von dort mit der S-Bahn zurück bis *Stuttgart_Hbf_(tief)*. Da dies in der Regel länger dauert, soll der Fußweg für die Mehrheit der Reisenden von einem Reiseplanungssystem bevorzugt vorgeschlagen werden.

Die vorliegende Arbeit definiert je zwei innerhalb von 500 Metern liegende Stationen als ein **Stationspaar**. Unter 6.919 Stationen bilden sich 279 Stationspaare. In den Stationspaaren soll jeweils eine doppelt gerichtete Kante mit uneingeschränkter Abfahrtszeit eingerichtet werden. Das Gewicht entspricht der Zeitdauer bei einer Gehgeschwindigkeit von 6 km/h.

Um diese Daten zu beziehen wird die Methode *readFromDBFiles(...)* der Klasse *TranNetwork* weiter ergänzt.

- *geographic_dist(i, k)*: aus den geographischen Koordinaten berechenbare Distanz zwischen Station *i* und *k*.
- *walkCost*: in Sekunden umgerechnete Gehzeit.
- *wArc*: Fußwegkante.
- *_walkableStationsFromStations*: eine Map-Struktur zur Speicherung aller Stationspaare mit Fußwegbindungsmöglichkeit.


```
void TrainNetwork::readFromDBFiles(...)
    ...
1 // Abschnitt Berechnung von Fußwegen
2 for jede Station i
3     for jede Station k
4         if geographic_dist(i, k) < 0,5 km && i != k
5             walkCost = geographic_dist(i, k) / (6 km/h) * 3600
6             wArc = pair(k, walkCost)
7             _walkableStationsFromStations[i].push_back(wArc)
    ...
```

Auch der Suchalgorithmus muss die neuen Fußwege einschließen. Zwischen den Zeilen 13 und 14 der Funktion *Earliest_Arrival(...)* muss eine Überprüfung auf Fußwegmöglichkeit (Zeilen a - i) durchgeführt werden.

- *wTestDistance*: temporäre Speicherung der über den Fußweg (*u*, *w*) erreichten Distanz an Station *w*.

```
Earliest_Arrival (...)
11 ...
12     if u == targetStation
13         break // Endstation gefunden
    // Abschnitt Fußwegmöglichkeitsprüfung
a     if u existiert in _walkableStationsFromStations[]
b         for jedes Stationspaar (u, w) an u
c             w = die von u aus zu Fuß erreichbare Station
d             walkCost = die Fußwegdauer von u zu w
e             wTestDistance = distance[u] + walkCost
f             if wTestDistance < distance[w]
g                 Füge (w, wTestDistance) zu PQ hinzu
h                 distance[w] = wTestDistance
i                 previousStation[w] = u
    // Abschnittsende
14     for jeder Abfahrtsknoten m an Station u ab distance[u]
15 ...
```

1.3.4 Umsteigeproblem

Der **Umstieg** beschreibt einen Prozess während der Reise, in dem der Reisende an einer Station aus seinem benutzten Fahrzeug aussteigt und in ein anderes Fahrzeug wieder einsteigt. Zum Erreichen des Reiseziels kann ein

Umstieg vorteilhaft sein, wenn die Reise dadurch gekürzt wird, oder er ist erforderlich, im Falle, dass kein Fahrzeug zwei Endpunkte der Reise direkt verbindet.

Unter dem Begriff **Umsteigezeit** versteht man die Dauer dieses Prozesses. Die „earliest-arrival“-Suche schließt bereits Umstiege ein. Dabei beträgt die Zeit eines vorteilhaften oder erforderlichen Umstiegs in den meisten Fällen weniger als 3 Stunden. Für jede Station ist aber eine Mindestumsteigezeit vorgegeben, die im Falle eines Umstiegs eingehalten werden muss, da das ankommende und das abfahrende Fahrzeug häufig nicht am gleichen Bahnsteig anhalten, so dass ein zusätzlicher Zeitaufwand für den Übergang entsteht.

1.3.4.1 Erste Implementierung

Die Mindestumsteigezeit darf aber auch nicht auf allen Ankünften an einer Station pauschal implementiert werden, weil eine Weiterfahrt im selben Fahrzeug keines Ausstiegs und somit keiner Zusatzzeit bedarf.

In diesem Zusammenhang soll eine Überprüfung der Fahrtnummer der fortzusetzenden Fahrt (Kante) an der Station des Abfahrtsknotens in das Programm integriert werden. Ein Abgleich mit der vorigen Fahrtnummer entscheidet ob die Mindestumsteigezeit dieser Station abgewartet werden muss. Dafür wird ein zusätzliches Array *bestTripld[]* zur Speicherung von Fahrtnummern der bereits erreichten Stationen benötigt.

Zur Umsetzung dieser Idee kann ein Kriterium zwischen Zeile 15 und Zeile 16 der Funktion *Earliest_Arrival(...)* eingefügt werden (Zeilen a - c).

- *minTransferTime[u]*: Mindestumsteigezeit an Station *u*.

```
Earliest_Arrival (...)  
14 ...  
15 for jeder über m erreichbare Ankunfts-knoten n  
    // Abschnitt Prüfung der Fahrtnummer  
a     if bestTripld[u] != tripld(m, n) && u != sourceStation  
b         if Abfahrtszeit von m < distance[u] + minTransferTime[u]  
c         continue  
    // Abschnittsende  
16     v = angehörige Station des Ankunfts-knotens n  
17 ...
```

1.3.4.2 Korrektur

Diese einfache Implementierung scheint auf den ersten Blick das Umsteigeproblem gelöst zu haben, denn viele Verbindungen mit sehr knappen Umstiegen, die von der einfachen „earliest-arrival“-Suche gefunden wurden, können hier die engeren Kriterien nicht mehr erfüllen und müssen spätere Abfahrtszeiten und letztendlich eine spätere Ankunftszeit am Zielort in Kauf nehmen. Wenn man aber einige Szenarien näher betrachtet, ist es ersichtlich, dass ein **globales Optimum** aufgrund des Greedy-Prinzips des Algorithmus verpasst werden kann, wenn stets das **lokale Optimum** gewählt wird.

Ein selbsterklärendes Beispiel dazu siehe Abbildung 7.

Es werden Abfahrts- und Ankunftszeiten des gleichen Fahrzeugs an einer Station vereinfacht ohne Unterscheidung dargestellt. Die tatsächliche Abweichung liegt bei ca. 30 Sekunden.

Start: Stuttgart_Hbf_(tief) **Ziel:** Stuttgart_Universität **Startzeit:** 12:20

Erwartetes **globales Optimum:**

(S2) Stuttgart_Hbf_(tief)[12:25] -> Stuttgart_Stadtmitte[12:27] ->
Stuttgart_Feuersee[12:29] -> Stuttgart_Schwabstraße[12:30] ->
Stuttgart_Universität[12:35]

Gefundene Verbindung aus **lokalen Optima:**

(S4) Stuttgart_Hbf_(tief)[12:23] -> Stuttgart_Stadtmitte[12:24] ->
Stuttgart_Feuersee[12:26] -> Stuttgart_Schwabstraße[12:27]
(Mindestumsteigezeit 5 Minuten) (S1) Stuttgart_Schwabstraße[12:40] ->
Stuttgart_Universität[12:45]

Abbildung 7 Beispiel Abweichung vom Optimum

Zur Behebung dieses Problems müssen weitere Knoten, die sich nicht als lokales Optimum erweisen, jedoch in Betracht gezogen werden, wenn ihre Ankunftszeit innerhalb des Bereichs von Minimalumsteigezeit der bisher bekannten frühesten Ankunftszeit an entsprechenden Stationen liegt. Daher ist die Funktion *Earliest_Arrival* aus 1.3.2 in Zeilen 18-21 korrekturbedürftig.

Diese einfache Korrektur hat allerdings keine Auswirkung, wenn die Schleifensteuerung (Zeilen 10-11) unverändert bleibt. Daher müssen auch schon zuvor aktivierte und anschließend gesperrte (besetzte) Stationen, zu denen der kürzeste Weg bereits bekannt ist, die aber aufgrund der abweichenden Ankunftszeit (gleich oder später) oder Fahrtnummer erneut aus der Prioritätswarteschlange gezogen werden, entsperrt und nochmal bewertet werden. Die wiederholte Suche an diesen Stationen ist auf zwei Kriterien eingeschränkt:

- nur Abfahrten innerhalb der Mindestumsteigezeit ab der tatsächlichen frühesten (nicht der aktuellen) Ankunft
- nur wenn die Fahrtnummer der nächsten Kante (Primärverbindung) mit der aktuell ankommenden Fahrtnummer übereinstimmt

Alle anderen Möglichkeiten wurden bereits in der Iteration der tatsächlichen frühesten Ankunft an der Station ausgeschöpft.

Der erweiterten Prüfung zufolge müssen neben der Station und deren Ankunftszeit (d.h. Kopfknoten der nächsten Kante) auch die Fahrtnummer $\text{tripld}(m, n)$ der Kante in einem zusammengesetzten Eintrag in die Prioritätswarteschlange aufgenommen werden. In der Implementierung lassen sich diese zwei Funktionsblöcke wie folgt hinzufügen bzw. umschreiben.

```
Earliest_Arrival (...)  
15 ...  
    // Neuer Abschnitt: Suche eingrenzen  
    i if u schon besetzt && tripld(m,n) != uArrivingTripld  
    ii continue // verdoppelte Suche unterbinden  
    // Abschnittsende  
16 v = angehörige Station des Ankunfts-knotens n  
17 vTestDistance = uArrivingDistance + cost(m,n)  
18 if vTestDistance < distance[v]  
19     Füge (v, vTestDistance, tripld(m, n)) zu PQ hinzu  
20     distance[v] = vTestDistance  
21     previousStation[v] = u  
    /* Neuer Abschnitt:  
       nicht optimale Ankünfte wenn sinnvoll auch behalten */  
    a else if vTestDistance < distance[v] + minTransferTime[v]  
    b     Füge (v, vTestDistance, tripld(m, n)) zu PQ hinzu  
    // Abschnittsende  
22 ...
```

Zwischen Zeile 10 und Zeile 11 der Funktion *Earliest_Arrival(...)* ist die neu aktivierte aus der Prioritätswarteschlange gezogene Station zu messen, ob

ihre Ankunftszeit auch tatsächlich innerhalb der Mindestumsteigezeit ab der frühesten Ankunftszeit an dieser Station liegt (Zeilen a-c).

```
Earliest_Arrival (...)  
6  ...  
7  while PQ nicht leer  
8      u = Station aus PQ mit geringstem Zeitstempel  
+      uArrivingDistance = Zeitstempel von u // Neue Zeile  
+      uArrivingTripld = Fahrnummerstempel von u // Neue Zeile  
9      Lösche u aus PQ // u wird aktiviert  
10     if u bereits einmal aus PQ gezogen  
        // Neuer Abschnitt: sinnvolle Ankünfte einschließen  
a         if uArrivingDistance < distance[u] + minTransferTime[u]  
b             Merke dass u schon besetzt war  
c         else  
11             continue // verwerfe nur wenn nicht sinnvoll  
        // Abschnittsende  
12     if u == targetStation  
13     ...
```

1.3.5 „latest-departure“-Suche

Die durch 1.3.3 und 1.3.4 erweiterte „earliest-arrival“-Suche auf Basis des Dijkstra-Algorithmus berechnet von einer beliebigen Ausgangsstation A ab einem angegebenen Zeitpunkt *t* die früheste Ankunft an der Zielstation B, wobei mögliche Fußwege zwischen nahliegenden Stationen und minimale Umsteigezeiten berücksichtigt werden.

Bedingt durch das Greedy-Prinzip des Algorithmus, versucht das vorliegende Programm stets auch alle Zwischenstationen, die für den Aufbau des Pfades benötigt werden, frühestmöglich zu erreichen. Es könnte aber daraus folgern, dass an einigen Stationen, insbesondere an der Ausgangsstation, auch eine spätere Abfahrt die früheste Ankunft an der Zielstation garantieren würde. Dadurch dass sich die Abfahrt in A verzögert trotz rechtzeitiger Ankunft in B, kann die Reisezeit der Verbindung auf dem gesamten öffentlichen Verkehrsmittel (erster Einstieg bis zum letzten Ausstieg), **Beförderungszeit** genannt, verkürzt werden, was der Reisende in einem angenommen pünktlichen Verkehrsnetz auch erwarten würde.

Angesichts dessen stellen wir unsere Anforderung um: Gesucht wird nun eine schnellste Verbindung ab *t*, die als nächstes kommt. Die konkrete

Formulierung lautet folgendermaßen. Unter allen Verbindungen mit der frühesten Ankunftszeit in B, suchen wir eine solche Verbindung ab t von A, die eine möglichst späte Abfahrt in A erlaubt. Eine Suche, welche diese Aufgabe erfüllt, wird als „**latest-departure**“-Suche bezeichnet.

Diese **Rückwärtssuche** strebt an, anhand einer bereits errechneten frühesten Ankunftszeit am Zielort zurückkehrend dem gleichen Prinzip folgend die späteste Abfahrt zu bestimmen. Daher wird der vorhandene Suchalgorithmus mit geringfügiger Anpassung übernommen.

In dieser Suche ist der ganze Ablauf umgekehrt. Ein einziger asymmetrischer Unterschied ist es, dass in der Rückwärtssuche die Mindestumsteigezeit auch an der Station nach dem Fußweg betrachtet werden soll, diese aber in dem tatsächlichen umgekehrten Suchvorgang schon vor dem gefundenen Fußweg vorkommt. Um in beiden Suchen einen einheitlichen Sinn zu ergeben, muss dabei am Ausgangspunkt des Fußwegs die Mindestumsteigezeit des Fußwegziels verwendet werden.

1.4 Graphische Oberfläche

Um das im Kapitel 1.3 aufgebaute Programm auf eine benutzerfreundliche Weise bedienen und Suchergebnisse verständlich darstellen zu können, wird in diesem Abschnitt eine graphische Oberfläche eingeführt.

1.4.1 Anfrageeingabe

Zur Eingabe von Stationen sowie Startzeit werden Objekte der Qt-Klasse *QPlainTextEdit* vergeben. Das Programm verwendet außerdem *QPushButton* zum Auslösen der Abfrage sowie zur Unterstützung des Tausches von Start- und Zielstation für eine schnelle Rückfahrtsuche.

Die als String eingelesene Startzeit im Format (HH:MM) wird erst in Sekunden ab Tageszeit 00:00:00 umgerechnet und als int eingespeichert.

Die als String eingelesenen Stationen werden unberührt an die Funktion des Suchalgorithmus übergeben und mithilfe von *_stationNameIdMap* in die Stationsnummern übersetzt. Dabei muss die Eingabe des Namens mit der im Graphen gespeicherten Bezeichnung genau passen (siehe Abbildung 8). Anderenfalls kommt eine Rückmeldung mit dem Hinweis, den Stationsnamen zu korrigieren.

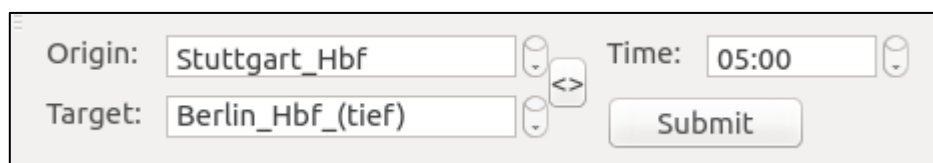


Abbildung 8 Eingabebereich

1.4.2 Routendarstellung

Anhand der angeketteten Stationen aus dem assoziativen Datenfeld (map) der Dijkstra-Rückwärtssuche kann der gefundene Pfad ermittelt werden. In dem Textfeld (Abbildung 9) werden nur Start- / Zielstation und erforderliche Umstiege dargestellt, nicht die einzelnen durchfahrenen Zwischenstationen.

Route of the trip / Test information:		
05:08	Stuttgart_Hbf	ICE_774
09:17	09:31 Hannover_Hbf	ICE_553
11:05	Berlin_Hbf_(tief)	

Abbildung 9 Routendarstellung

1.4.3 Visualisierung der Route

Diese gefundene Route (lastest-departure) lässt sich auf einer Landkarte veranschaulichen, damit der Nutzer einen einfachen Eindruck bekommt, wie seine gewählte Verbindung führt und ob ein großer Umweg gegenüber der Luftlinie besteht. Als Hintergrund wird ein Abbild einer Deutschlandkarte mit Grenzen der Bundesländer verwendet (Abbildung 10).



Abbildung 10 Deutschlandkarte mit Ländergrenzen

Zur Darstellung wird zunächst auf die geographischen Koordinaten aller Zwischenstationen des gefundenen Pfades zurückgegriffen. Jede Station verbindet ihre Nachbarstation auf dem Pfad mit einer roten Linie (siehe Abbildung 11). Der Pfad wird anschließend auf der hinterlegten Karte visualisiert.

Eine Kalibrierung aus zwei Punkten (Stationen) mit bekannten Koordinaten, die manuell den entsprechenden Positionen auf der Karte zugeordnet werden, berechnet die Koordinaten der vier Eckpunkte des Bildes. Näherungsweise wird das Verhältnis zwischen Distanz und Grad auf der jeweiligen Ebene (senkrecht und horizontal) als linear angenommen. So können alle Stationen, deren Koordinaten im Bereich der vier eingrenzenden Eckpunkte liegen, einbezogen und dargestellt werden.

Für den Einbau dieser Visualisierung findet die Klasse *QGraphicsView* Anwendung.



Abbildung 11 Visualisierung des Suchergebnisses

1.5 Experimentelle Evaluation

In diesem Kapitel werden die Leistungsfähigkeit und Zuverlässigkeit des oben eingeführten Reiseplanungssystems überprüft. Dazu soll das Programm eine Aufgabe von 1000 kontinuierlichen Reiseanfragen aus zufallsbedingt gewählten Ausgangs- und Zielstationen sowie einer beliebigen Startzeit zwischen 8 und 10 Uhr am Vormittag ausführen.

1.5.1 Basismodus

Die Länge der in der Klasse *TrainNetwork* (siehe 1.2.2.1-e) gebildeten Listen *_stationDepNodesMap* und *_stationArrNodesMap* (für die Rückwärtssuche) beläuft sich jeweils auf genau 6.919, die Anzahl Stationen, sprich jede Station verfügt über mindestens einen Abfahrts- und einen Ankunfts-knoten.

Unter den Anfragen, die das Programm erhält, können jedoch unlösbare Szenarien vorkommen. Dies kann beispielsweise daran liegen, dass eine Region aus mehreren Stationen nur unter sich selbst verbunden und von dem größeren Verkehrsnetz abgetrennt ist. Für eine Abfrage von dieser Region in den restlichen Teil steht daher keine Verbindung zur Verfügung. Des Weiteren bieten viele Stationen nur sehr begrenzte Anschlussmöglichkeiten in das große Netz, so dass sie unter Umständen bei einer späteren Tageszeit bis zum Betriebsschluss des Tages nicht mehr erreicht oder verlassen werden können.

In diesem Zusammenhang soll bei der Evaluation auch die Anzahl der lösbaren sowie nicht lösbaren Anfragen vermerkt werden. Theoretisch können die willkürlich gewählte Ausgangs- und Zielstation identisch sein. Die Anzahl dieser Gruppe darf daher auch nicht ignoriert werden.

Drei Ausführungen mit einem Umfang von jeweils 1000 Anfragen ergeben folgende Daten in der Tabelle 2:

Tabelle 2 Evaluationsergebnisse im Basismodus

Nr.	Anzahl Anfragen	davon gelöst	davon ungelöst	davon ungültig	Gesamtzeit (Sekunden)	Durchschnitt pro Anfrage (Sekunden)
1	1000	976	24	0	2277,36	2,227
2	1000	982	18	0	2418,64	2,418
3	1000	978	22	0	2289,61	2,29
D	1000	978,67	21,33	0,0	2381,67	2,312

1.5.2 Speed-up

Im Basismodus sucht das Programm an besetzten (zum ersten Mal aktivierten) Stationen nach Abfahrtsmöglichkeiten bis zum Betriebschluss des Tages mit der Annahme, es könnte eine schnellere Möglichkeit bestehen, trotz späterer Abfahrt früher am Ziel anzugelangen. In der Realität ist dies ab einem Warten von über 3 Stunden sehr unwahrscheinlich, es sei denn, es bestünde in dieser Zeit überhaupt keine Abfahrtsmöglichkeit, so dass die Suche ab hier nicht mehr fortgesetzt würde. Solche Stationen, die zwischen 8 und 20 Uhr nach einer Ankunft eine Lückenzeit ohne Abfahrtsknoten innerhalb von 10.000 Sekunden (2,78 Stunden) haben, bezeichnet diese Arbeit als **Sonderstationen**.

Zur Beschleunigung des Suchprozesses führen wir eine **maximale Umsteigezeit** ein, die 10.000 Sekunden beträgt. Berührt die Suche eine Sonderstation und wird deshalb eine Weiterfahrt verhindert, so dass das Ziel nicht mehr erreicht wird, setzt das Programm den maximalen Umsteigewert zurück und führt nochmal eine normale Suche durch. Wenn das Ziel immer noch nicht gefunden wird, ist die Suche zu Ende.

Dieser Prozess erhöht zwar den Rechenaufwand einzelner betroffenen Anfragen, senkt aber die Zeit erheblich bei der Suche nach Wegen, die keinen erforderlichen Umstieg über Sonderstationen führen, welche auch die Mehrheit der Anfragen bilden. Die Ergebnisse unter den gleichen Bedingungen wie im Basismodus können der Tabelle 3 entnommen werden.

In der allgemeinen Einzelanfrage ist der Vorteil noch größer, da die manuelle Eingabe von Stationen meistens keine willkürlichen Randstationen auf dem Land einbezieht.

Tabelle 3 Evaluationsergebnisse im Speed-up-Modus

Nr.	Anzahl Anfragen	davon gelöst	davon ungelöst	davon ungültig	Gesamtzeit (Sekunden)	Durchschnitt pro Anfrage (Sekunden)
1	1000	976	24	0	1182,1	1,182
2	1000	981	19	0	1141,95	1,142
3	1000	982	18	0	1114,25	1,114
D	1000	979,67	20,33	0,0	1146,1	1,146

1.6 Zusammenfassung

1.6.1 Überblick

In Teil I der Arbeit wurden zunächst die Problemstellungen der Verbindungssuche im öffentlichen Verkehr vorgestellt und dadurch motiviert die Grundsätze sowie Modelle der Verkehrsnetzabbildung präsentiert. Anschließend beschäftigte sich die Arbeit aufgrund der bereitgestellten Fahrplandatenstruktur mit einer time-expanded-artigen Graphrepräsentation, auf der wiederum eine time-dependent-nahe Variante des Dijkstra-Algorithmus zur Suche nach den kürzesten Pfaden angewendet wurde, um die Prioritätswarteschlange von einer massiven Überflutung durch unbrauchbare Knoten zu bewahren.

Zur Einbindung von Mindestumsteigezeiten, sowie von Fußwegen zwischen Stationen, wurde der Algorithmus mehrmals erweitert und angepasst. Fortan garantierte dieser die früheste Ankunft am Ziel. Eine gleichermaßen funktionierende Rückwärtssuche konnte die Abfahrtszeit am Ausgangsort möglichst nach hinten verschieben, ohne dass sich die Ankunft am Ziel dadurch verzögert.

Eine graphische Oberfläche wurde zur Spezifizierung von Anfragen und Darstellung der Ergebnisse zum bestehenden Programm hinzugefügt.

In der Evaluation zeigte das Programm eine hohe Zuverlässigkeit und eine zufriedenstellende durchschnittliche Leistungsfähigkeit. Zur Reduzierung der Rechenzeit wurde eine Speed-up-Methode vorgestellt, welche in den überwiegenden Fällen ebenfalls die optimale Lösung liefert.

1.6.2 Verbesserungsmöglichkeiten

1.6.2.1 Fußweg

Der Fußweg zwischen zwei nahliegenden Stationen wurde in diesem Reiseplanungssystem als ein einfaches Verkehrsmittel implementiert, das zu jeder Zeit verfügbar ist und eine konstante Geschwindigkeit hat, mit welcher die geographische Luftliniendistanz zwischen beiden Stationen zurückzulegen ist. Das Antreten des Fußwegs hat keine Voraussetzung und nach dem

Übergang muss die Mindestumsteigezeit der Station, an welcher der Fußweg endet, bis zum nächsten Einstieg eingehalten werden.

Solche Fußwege sind ggf. unrealistisch, da viele Stationspaare zwar sehr nahe Koordinaten haben, sich aber in unterschiedlichen Ebenen eines Bauwerks befinden. So ist die Luftliniendistanz z.B. zwischen Berlin_Hbf und Berlin_Hbf_(S-Bahn) 0, wobei in der Realität ein Übergang zwischen ihnen mindestens 5 Minuten beträgt.

Die Dauer eines Fußwegs sollte deshalb lage- und bahnhofsabhängig sein und mit Umsteigepufferzeit zweiseitig symmetrisch vorgegeben werden.

1.6.2.2 Rückwärtssuche

Da die Rückwärtssuche einen Dijkstra-Algorithmus verwendet, folgt sie auch dem Greedy-Prinzip und versucht, solange die früheste Ankunft am Zielort gewährt werden kann, jede Station so spät wie möglich zu verlassen. Zwar gibt es gute Gründe, an der Ausgangsstation möglichst spät abzufahren, ist es für den Reisenden kein logisches Verhalten, sofern die früheste Ankunft am Ziel nicht verpasst wird, unterwegs an einer Station auszusteigen, nur um auf eine spätere Abfahrtsmöglichkeit zu warten. Man möchte lieber etwas Pufferzeit für den nächsten unvermeidbaren Umstieg einsparen.

Es wäre eine Lösung gefragt, welche die Anzahl Umstiege möglichst gering hält. Damit ändert sich das Problem in eine Multi-Kriterien-Suche, die während der Laufzeit in der Hauptschleife nicht unterbrochen werden darf, bis alle Elemente aus der Prioritätswarteschlange entfernt wurden.

1.6.2.3 Fahrplandaten

Die bereitgestellten Fahrplandaten enthalten die Gesamtmenge der Fahrten aus dem Fahrplanjahr 2012 an allen Verkehrstagen. D.h. die Suche kombiniert alle möglichen Fahrzeugfahrten aus dem gesamten Jahr und findet die schnellste Verbindung aus einer Vereinigungsmenge aller Verkehrstage. In das Ergebnis kommen auch Verbindungen, welche einen Zug, der nur samstags fährt, und einen anderen Zug, der nur sonntags fährt, kombinieren.

Zur Behebung dieses Problems muss man anhand einer Bitfeld-Datei die Verkehrstage aller Züge aus den HAFAS-Rohdaten entschlüsseln und nur Züge anwählen, die an einem bestimmten Tag verkehren.

Da die vorliegenden Fahrplandaten auch innerhalb Deutschlands nicht vollständig sind, weichen die Suchergebnisse teilweise von der Realität ab. Im Fernverkehr fehlen bestimmte Züge mit Start oder Ziel im Ausland, die allerdings als normale Fernverkehrszüge im deutschen Binnenverkehr fahren. Im Nahverkehr sind viele Züge der nichtbundeseigenen Eisenbahnen nicht völlig in diesen Fahrplan integriert.

2 Teil II: Wesentliche Unterschiede der Verbindungssuche PRIMA und HAFAS

2.1 Einführung

PRIMA (Prozessunterstützung im Angebotsmanagement) ist ein Prognosemodell bei der DB Fernverkehr AG, welches die wirtschaftliche Bewertung von Fahrplanszenarien für die Zukunft unterstützt. Prognostiziert werden beförderte Personen auf jeder Zugverbindung, die vom Verbindungssuchalgorithmus in PRIMA gefunden wird. Dieser Algorithmus soll möglichst alle Verbindungen finden, die von den Kunden gewählt und somit auf denen Tickets ausgestellt werden können. Dies hängt wiederum von dem Auskunftssystem HAFAS (HaCon Fahrplanauskunftssystem) ab, welches auf bahn.de, an jedem DB-Automaten und beim Verkaufssystem bundesweit in Reisezentren sowie DB-Agenturen Anwendung findet.

Das **Auskunftssystem HAFAS** ist ein zentral verwalteter Dienst zur effizienten und aktuellen Verbindungssuche, ausschließlich nach dem die Vertriebskanäle fahrplanbasierte Tickets ausstellen können. Bedient wird dasselbe System mit unterschiedlichen Benutzeroberflächen, die in dem jeweiligen Vertriebskanal angeboten werden. Diese Arbeit bezieht sich, wenn nicht explizit erwähnt, auf das online abfragbare Suchportal (<http://reiseauskunft.bahn.de/bin/query2.exe>), welches für Jedermann leicht zugänglich ist.

TReK (Toolunterstützung Reisekettenanalyse) ist ein Berichtssystem für Analyse und Reporting von Reisenden, Reiseverhalten und Reiseketten. TReK setzt sich aus zwei Datenquellen zusammen: Zahlen basierend auf Reiseketten aus expliziten Verbindungs- und Ticketdaten, die bei der Buchung erfasst werden (**EMS**), und eine systematische Ergänzung modellierter Reiseketten für Tickets und Reisen aus nicht fahrplanbasierten Verkäufen (**HPE**). EMS-Daten sind eine konsistente Reflexion der buchbaren Verbindungen aus HAFAS und werden hier zwecks Vergleichs als Bezug genommen.

Ziel für Teil II der Arbeit ist es, die wesentlichen Unterschiede zwischen den jeweils von PRIMA und von HAFAS verwendeten Verbindungssuchmethoden und -auswahlkriterien hinsichtlich Kundenbuchungen darzustellen und darauf beruhend potenzielle Verbesserungsvorschläge für PRIMA abzuleiten.

2.2 Dokumentation der Algorithmen

2.2.1 PRIMA-Verbindungssuche [9]

PRIMA verwendet die Verkehrsmodellierungssoftware VISUM der PTV Group. Das Branch and Bound Verfahren [10] kommt für den Verbindungsaufbau zwischen zwei Bezirken zum Einsatz. Unter dem Begriff **Bezirk** werden in der vorliegenden Arbeit Landkreise sowie kreisfreie Städte (Stadtkreise) anstatt Regierungsbezirken bezeichnet. So befinden sich beispielsweise fünf Bezirke – Stuttgart Stadtkreis, Landkreis Ludwigsburg, Landkreis Esslingen, Landkreis Böblingen und Rems-Murr-Kreis – im Großraum Stuttgart. Maßgebend für den Start und Ziel der Berechnung ist immer der jeweils definierte **Schwerpunkt eines Bezirkes**, der durch eine bestimmte Anbindungszeit an einen Knoten (Bahnhof) angebunden ist. Ein Bezirksschwerpunkt kann mehrere Bahnhofsknoten mit unterschiedlicher Anbindungszeit nach Entfernung verbinden.

Ausgehend von jedem Quellbezirk wird ein Suchbaum von Teilverbindungen generiert, die parallel verglichen und bewertet werden, um alle hinreichend guten Verbindungen im Sinne von geringem Suchwiderstand aufzubauen.

Für jede gefundene Verbindung c wird ein **Suchwiderstand** zum Zweck der Qualitätsbewertung aus einer gewichteten Summe von Reisezeitwiderstand mit Faktor 1 (Min), Umsteigewiderstand mit Faktor $Fact_{transf}$ (Min/Umstieg) und Preiswiderstand mit Faktor $Fact_{price}$ (Min/Preiseinheit) berechnet.

$$\begin{aligned} \mathbf{Suchwiderstand}(c) &= 1 * \mathbf{Reisezeit}(c) \\ &+ \mathbf{Fact}_{transf} * \mathbf{Umsteigehäufigkeit}(c) \\ &+ \mathbf{Fact}_{price} * \mathbf{VSys-Wid}(c) \end{aligned}$$

$VSys-Wid$ (**Verkehrssystemspezifische Widerstände**) ist ein nach Verkehrssystem differenzierter Widerstand, der näherungsweise den Fahrpreis berücksichtigt. Die Ermittlung von $VSys-Wid$ erfolgt durch das Summieren eines linearen Teils (Produkt aus Tarifpunkten der Verbindung und $linear-pr$ Preiseinheiten/Tarifpunkt nach Verkehrssystem) und eines konstanten Zuschlags $base-pr$ Preiseinheiten nach Verkehrssystem. $base-pr$ richtet sich immer nach der höchsten genutzten Zuggattung der Verbindung.

$$\mathbf{VSys-Wid(c) = linear-pr(Verkehrssystem) * Tarifpunkte(c) + base-pr(Verkehrssystem)}$$

Der Faktor jeder Gewichtung kann in VISUM individuell eingestellt werden. Für die Berechnung des Suchwiderstandes verwendet PRIMA folgende Parameter:

- $Fact_{transf} = 35$ (Min/Umstieg)
- $Fact_{price} = 8$ (Min/Preiseinheit)
- $linear-pr$ (Produktklasse IC/EC) = $7 * 10^{-5}$ (Preiseinheit/Tarifpunkt)
- $linear-pr$ (Produktklasse ICE) = $7 * 10^{-5}$ (Preiseinheit/Tarifpunkt)
- $linear-pr$ (Produktklasse NV) = $5 * 10^{-5}$ (Preiseinheit/Tarifpunkt)
- $linear-pr$ (Nachtzug) = $3 * 10^{-5}$ (Preiseinheit/Tarifpunkt)
- $linear-pr$ (Fußweg) = 0
- $base-pr$ (höchste Produktklasse IC/EC) = 4 (Preiseinheit)
- $base-pr$ (höchste Produktklasse ICE) = 6 (Preiseinheit)
- $base-pr$ (höchste Produktklasse NV) = 1,65 (Preiseinheit)
- $base-pr$ (Nachtzugverkehr) = 8 (Preiseinheit)

Unabhängig von der zeitlichen Ausgangslage werden Verbindungen gelöscht, die zu stark vom Optimum abweichen. Eine gefundene Verbindung c auf einer Relation r wird in PRIMA gelöscht, wenn mindestens eines der folgenden Kriterien erfüllt wird:

- $Suchwiderstand(c) > 1,5 * [minimaler\ Suchwiderstand\ der\ Relation] + 10$
- $Reisezeit(c) > 2 * [minimale\ Reisezeit\ der\ Relation] + 10$
- $Umsteigehäufigkeit(c) > [minimale\ Umsteigehäufigkeit\ der\ Relation] + 2$

Als **Überholung** bezeichnet man ein Szenario, in dem sich eine schnellere Verbindung con_A innerhalb der zeitlichen Lage einer anderen langsameren Verbindung con_B befindet. In diesem Fall ist con_B eine **überholte Verbindung** und con_A eine **überholende Verbindung**. Verbindung con_B ist nicht notwendigerweise unattraktiv, wenn con_B weniger Umstiege als con_A beinhaltet.

Es besteht daher ein **Dominanzkriterium**, nach welchem die ersten Ergebnisse gefiltert werden. Eine Verbindung wird dominiert und verworfen,

soweit sie in jeder Hinsicht ungünstiger als eine andere Verbindung in der gleichen zeitlichen Lage wirkt. In einer konkreten Bedingung lautet dies:

Verbindung c' dominiert Verbindung c , falls:

- c' innerhalb des Zeitintervalls von c liegt, und
- $\text{Umsteighäufigkeit}(c') \leq \text{Umsteighäufigkeit}(c)$, und
- $\text{Suchwiderstand}(c') \leq \text{Suchwiderstand}(c)$, und
- in mindestens einem der drei Kriterien echte Ungleichheit gilt

Außerdem besteht die Möglichkeit, eine **Vorauswahl** unter gleichen oder ungleichen Konditionen durchzuführen, um den Suchbaum beim Aufbau jeder Teilverbindung einzugrenzen.

1) Eine Teilverbindung pc wird dabei gelöscht, wenn:

- $\text{Suchwiderstand}(pc) > 1,5 * [\text{minimaler Suchwiderstand}] + 10$

2) Unter allen verbliebenen Teilverbindungen wird pc gelöscht, wenn:

- $\text{Reisezeit}(pc) > 2 * [\text{minimale Reisezeit}] + 10$, und
- $\text{Umsteighäufigkeit}(pc)$ ist nicht minimal

oder

- $\text{Umsteighäufigkeit}(pc) > [\text{minimale Umsteighäufigkeit}] + 2$, und
- $\text{Reisezeit}(pc)$ ist nicht minimal

2.2.2 HAFAS-Verbindungssuche

Zur Verbindungsauskunft bietet HAFAS eine Vielzahl von Zusatzoptionen an. Bei der Standardeinstellung führt HAFAS eine Kürzeste-Wege-Suche unter Einhaltung der individuellen Einstellung des Nutzers durch und ändert die Bedingungen innerhalb eines angemessenen Rahmens, um alternative Verbindungen einzuschließen.

Nachstehend werden die Ergebnisse einer **Online-Suchphase** diskutiert. Die erste Online-Suchphase stellt die Suchergebnisse aus dem Suchportal (<http://reiseauskunft.bahn.de/bin/query2.exe>) dar, nachdem die Relation und eine Wunschabfahrtszeit eingetragen wurden und auf „Suchen“ gedrückt wird. Weitere Online-Suchphasen werden in Gang gebracht, wenn man nach einer

bereits durchgeführten Abfrage auf „Aktualisieren“ drückt. Eine Online-Suchphase beinhaltet 1 bis maximal 3 Suchvorgänge.

Ein **Suchvorgang** ist die minimale Einheit in der HAFAS-Verbindungssuche. Dieser kann im „Offline-HAFAS“ (DB-Fahrplaninformation) bei *Optionen->Suchoption->Sonstiges* als „*mindestens eine Verbindung suchen*“ eingestellt und genau abgebildet werden. Der Suchvorgang findet in der Regel zum angegebenen Zeitpunkt eine nächste schnellste Verbindung (Grundregel), die unter Anwendung mehrerer nacheinander folgender Zusatzregeln ggf. um weitere Verbindungen ergänzt wird.

Eine **nächste schnellste Verbindung** auf der Relation ab einem definierten Zeitpunkt (t_0) ist eine mit Abfahrtszeit ab t_0 gefundene Verbindung, die unter allen Verbindungen mit der frühesten Ankunftszeit (t_{Arr-e}) die späteste Abfahrtszeit (t_{Dep-l}) hat. Der Zeitraum der beiden Grenzen ergibt die **nächste schnellste Fahrzeit** ($T_{next} = t_{Arr-e} - t_{Dep-l}$) ab t_0 .

Das komplette Suchverfahren kann wie folgt beschrieben werden.

- **Grundregel:** Ab dem eingegebenen Zeitpunkt (t_0) wird für die aktuelle Relation (X->Y) eine nächste schnellste Verbindung mit einer Begrenzung von maximaler Umsteigeanzahl $Num_{transf-max}$, Basisverbindung (con_{bas}) genannt, gesucht und das Ergebnis in eine Liste *Result* gespeichert. $Result = \{con_{bas}\}$. con_{bas} hat die Abfahrtszeit t_{Dep-l} und die Ankunftszeit t_{Arr-e} .
- **Zusatzregel 1:** Ist diese Basisverbindung keine Direktverbindung und hat sie eine Umsteigehäufigkeit von Num_{transf} , so wird versucht ab Zeitpunkt t_0 bis t_{Dep-l} eine nächste schnellste Verbindung mit einer Begrenzung von maximaler Umsteigeanzahl ($Num_{transf-max} := Num_{transf} - 1$), Komfortverbindung (con_{comf}) genannt, aufzubauen. Sie wird zur Ergebnisliste hinzugefügt. $Result = \{con_{bas}, con_{comf}\}$. Dieser Schritt wird mehrmals rekursiv angewendet bis keine Verbindung mit geringerer Umsteigeanzahl mehr gefunden werden kann. Die Ergebnisliste kann theoretisch für jede Verbindung mit einer Umsteigeanzahl, die kleiner als Num_{transf} ist, eine oder sogar mehrere zugehörige Komfortverbindungen enthalten, solange deren Abfahrtszeit zwischen t_0 und t_{Dep-l} liegt. Je zwei Komfortverbindungen mit gleicher Umsteigeanzahl müssen dennoch zeitlich nacheinander liegen. Alle qualifizierten Komfortverbindungen werden zur Ergebnisliste hinzugefügt. $Result = \{con_{bas}, con_{comf1}, con_{comf2} \dots\}$.
- **Zusatzregel 2:** Sollte in der Basisverbindung ein Abschnitt eines Zuges innerhalb der Reisekette markiert sein, so wird erneut ohne alle markierten Zugfahrtabschnitte ab Zeitpunkt t_0 bis t_{Dep-l} nach der nächsten oder ggf. auch mehreren schnellsten Verbindungen, Ersatzverbindungen ($con_{replace1}, con_{replace2} \dots$) genannt, gesucht. Je zwei Ersatzverbindungen müssen dabei

zeitlich nacheinander liegen. $Result = \{con_{bas}, con_{comf1}, con_{comf2} \dots\dots, con_{replace1}, con_{replace2} \dots\dots\}$.

- **Zusatzregel 3:** Wenn die Option „schnelle Verbindungen bevorzugen“ ausgeschaltet ist (bzw. im Offline-HAFAS „Preiswertsuche“ eingeschaltet), wird ab Zeitpunkt t_0 bis t_{Dep-l} nach nächsten schnellsten Verbindungen ohne Produktklasse ICE gesucht. Im Anschluss daran findet eine weitere Suche ohne die Produktklassen ICE und IC/EC statt. Bis zu einer Entfernung von 200 km wird auch eine dritte Suche ohne die Produktklassen ICE, IC/EC und D-Zug ausgelöst, d.h. nur NV-Verbindungen werden berücksichtigt. Darüber hinaus werden auch mögliche Kurzstreckenwege durchsucht, um die nächsten schnellen und bequemen Verbindungen aufzubauen, bei denen die Umsteigebahnhöfe innerhalb einer konfigurierbaren Ellipse liegen. Preiswerte Verbindungen (con_{econom}) und Kurzstreckenverbindungen (con_{short}) werden in die Liste aufgenommen. $Result = \{con_{bas}, con_{comf1}, con_{comf2} \dots\dots, con_{replace1}, con_{replace2} \dots\dots, con_{econom1}, con_{econom2} \dots\dots, con_{short1}, con_{short2} \dots\dots, \}$.
- **Nachauswahl:** Gleiche und wiederholte Verbindungen innerhalb der Ergebnisliste werden gelöscht. Sehr langsame Verbindungen werden auch gelöscht, wenn es eine schnelle (con_{bas}) oder komfortable Verbindung (con_{comf}) gibt, deren Fahrzeit X-mal kürzer ist.

Je nach Art der verwendeten Software erfolgt eine Fortsetzung:

- **Fortsetzung Offline-HAFAS:** Bei der HAFAS-Offline-Version unter Basiseinstellung (*Optionen->Suchoptionen->Sonstiges->Mindestens 01 Verbindungen suchen*) wird am Ende des Suchvorgangs die Liste *Result* von gefundenen Verbindungen nach sortierten Abfahrtszeiten angezeigt. Drückt man auf die Taste F3 (Nächste Verbindung vorwärts), wird ein neuer Suchvorgang mit $t_0 := t_{Dep-l} + 1 \text{ Minute}$ ab „Grundregel“ fortgesetzt.
- **Fortsetzung Online-HAFAS:** Bei der HAFAS-Online-Version werden die obigen Schritte von „Grundregel“ bis „Nachauswahl“ durchgehend wiederholt (min. 1 Mal bis max. 3 Mal, jeweils mit $t_0 := t_{Dep-l} + 1 \text{ Minute}$). Alle gefundenen Verbindungen werden erst zu dem Moment angezeigt, wenn *Result* nach einem kompletten Suchvorgang bereits mindestens 3 Verbindungen enthielt. Damit endet die aktuelle Online-Suchphase. Drückt man auf „Später“, wird eine weitere Suchphase ausgelöst, welche die neu gefundenen Verbindungen an das Ergebnis der letzten Suchphase anhängt und es ohne Neusortierung anzeigt.

2.2.3 Weitere Kenntnisse aus den HAFAS-Rohdaten [11]

2.2.3.1 Markierungen und Sonderzug

HAFAS bietet die Möglichkeit, bestimmte Züge streckenabschnittsweise zu markieren. Zwei Varianten von Markierung sind dafür vorgesehen, „00“-Markierung und „01“-Markierung.

„00“-Markierung:

Schnelle, hochwertige oder reservierungspflichtige Züge/Busse, die bei der Verbindungssuche auf bestimmten Streckenabschnitten andere parallel fahrende (meistens günstige aber nicht zu langsame) Züge unterdrücken können, werden „00“ markiert. Eine zusätzliche Suche ohne alle markierten Züge findet statt (siehe Kapitel 2.2.2 Zusatzregel 2). Davon betroffen sind u.a. **ICE-Sprinter, IC-Busse zwischen München ZOB und Prag**, ICE3-Fahrten (Linie 41, 42, 43, 45, 49, 78, 79) im Abschnitt der **Schnellfahrstrecke Köln-Rhein-Main (KRM) zwischen Frankfurt(M) Flughafen Fern** (oder Wiesbaden) **und KÖLN** (je nach Zug und Halt Köln Hbf, Köln Messe/Deutz oder Köln/Bonn Flughafen), ICE1-Fahrten (nur Linie 12, 22) im Abschnitt der Schnellfahrstrecke auf der **Mittelachse zwischen Frankfurt(M) Hbf und Göttingen oder Kassel-Wilhelmshöhe** (Linie 12) **bzw. Hamburg Hbf/Altona oder Hannover Hbf** (Linie 22). (Stand Fahrplan 2014)

Welche Fernverkehrszüge in welchen Abschnitten genau eine „00“-Markierung tragen, kann in den HAFAS-Rohdaten (%Rohdatenpfad%\fahrten\d0i80.101) eingesehen werden. Die Markierung ist als Attribut einer Zugfahrt hinterlegt und entspricht dem Format „*A 00 [Von-Bahnhof-IBNR] [Nach-Bahnhof-IBNR]“.

„01“-Markierung:

Erwartet stark ausgelastete Züge, die erfahrungsgemäß eine 100% Besetzung verursachen können, werden auf bestimmten Streckenabschnitten an bestimmten Verkehrstagen „01“ markiert, so dass ein graues Zeichen „R“ mit der Beschriftung „voraussichtlich starke Nachfrage“ auf bahn.de neben dem betroffenen Zug angezeigt wird und für diesen Zugabschnitt alternative Verbindungen ohne markierten Züge parallel vorgeschlagen werden.

Welche Fernverkehrszüge an welchen Tagen in welchen Abschnitten genau eine „01“-Markierung tragen, kann in den HAFAS-Rohdaten (%Rohdatenpfad%\fahrten\d0i80.101) eingesehen werden. Die Markierung ist als Attribut einer Zugfahrt hinterlegt und entspricht dem Format „*A 01[Von-Bahnhof-IBNR][Nach-Bahnhof-IBNR] [Verkehrstag als Bitfeld-Nr]“.

Sonderzug

Wenn die Standardmarkierungen „00“ und „01“ den Zweck nicht erfüllen können, steht in HAFAS noch eine Sondermarkierungsweise zur Verfügung. Diese Funktion ist speziell für den Fall eingerichtet, wenn ein Zug den anderen Zug derselben Produktklasse überholt aber beide Züge auf der betroffenen Strecke bereits markiert sind, oder wenn eine überholte Verbindung, die deutlich langsamer als die überholende Verbindung ist und daher bei der Nachauswahl entfernt wird aber trotzdem aufgrund des niedrigeren Fahrpreises aus Sicht des Kunden attraktiv sein könnte.

Als Option bietet HAFAS an, den schnelleren Zug als Stammzug und den hierdurch überholten Zug als Entlastungszug paarweise zu vermerken, so dass mögliche Verbindungen des Entlastungszuges auch gleichzeitig gefunden und bei der Ausgabe stets angezeigt werden, wenn eine Verbindung mit dem Stammzug gefunden wird. Eine bei dieser Zusatzsuche gefundene Verbindung wird behalten und nicht bei der Nachauswahl gelöscht.

HAFAS pflegt in den Stammdaten (%Rohdatenpfad%\stamm\sonderzg.101) eine Liste, auf welche die Stamm- und dazugehörigen Entlastungszüge paarweise eingetragen werden. Jeder Eintrag entspricht dem Format "[Stammzug-Nummer] [Stammzug-Verwaltungsnummer] [Entlastungszug-Nummer] [Entlastungszug-Verwaltungsnummer]".

2.2.3.2 Überschreibung von Mindestübergangszeiten

Für jeden Bahnhof wird eine **Mindestübergangszeit** festgelegt, so dass für jeden ankommenden Zug an diesem Bahnhof kein Umstieg in einen innerhalb dieser Zeit abfahrenden Zug möglich ist. Die Übergangszeiten sind unterschiedlich eingestellt. Größere Bahnhöfe haben meistens eine längere Mindestübergangszeit als kleinere Bahnhöfe, da der Fußweg zwischen Bahnsteigen an einem großen Bahnhof entsprechend länger sein kann.

Um einen betrieblich garantierten Anschluss zu gewähren, z.B. wenn der Umstieg am gleichen Bahnsteig stattfindet, kann die Übergangszeit an einem bestimmten Bahnhof zwischen zwei bestimmten Zügen mit einem kleineren Wert **überschrieben** werden. Typische Bahnhöfe, an denen ein kürzerer Übergang zwischen Fernverkehrszügen häufig stattfindet, sind Dortmund Hbf, Mannheim Hbf, Nürnberg Hbf, Köln Hbf und Leipzig Hbf. Die Überschreibung erfolgt über eine Liste, auf welche alle solche Sonderfälle eingetragen werden können. Sowohl PRIMA als auch HAFAS pflegen diese Liste, allerdings mit teilweise unterschiedlichen Einträgen. Ein wichtiger Grund dafür ist es, dass in HAFAS unterjährlich kurzfristige Änderungen aufgrund von Bauarbeiten und Zusatzbestellungen vorgenommen werden müssen, wohingegen PRIMA nur einen nichtänderbaren Fahrplan an einem korrekten Verkehrstag berücksichtigt.

Des Weiteren können in HAFAS Übergangszeiten nicht nur verringert werden, sondern es ist auch möglich, die Überschreibung so einzurichten, dass ein normaler Umstieg zwischen zwei Zügen ausgeschlossen wird. Die Absicht dieser Maßnahme ist es, auf einigen Strecken häufig vorkommende Verspätungen im betrieblichen Ablauf des auszustiegenden Zuges am Umsteigebahnhof zu berücksichtigen und Reisende auf einen gleichwertigen Zug, der wenige Minuten später die gleiche Strecke befährt, zu verweisen, so dass der Umstieg als sicherer gilt (z.B. Umstieg in Mannheim aus München kommend und nach Freiburg fahrend). Dies hat zur Folge, dass eine in PRIMA gefundene Umsteigeverbindung in HAFAS auch ausgeschlossen werden kann und eine Alternativverbindung mit längerer Fahrzeit vorgeschlagen wird.

Die Liste überschriebener Übergangszeiten in HAFAS ist in den Rohdaten (%Rohdatenpfad%\stamm\umsteigz.101) hinterlegt. Jeder Eintrag entspricht dem Format „ [An-Bahnhof-IBNR] [Von-Zug-Nummer] [Von-Zug-Verwaltungsnummer] [Nach-Zug-Nummer] [Nach-Zug-Verwaltungsnummer] [aktualisierte Übergangszeit]“. Eine ähnliche Liste in PRIMA kann aus VISUM exportiert werden.

2.3 Unterschiede PRIMA/HAFAS-Verbindungssuche

2.3.1 Theoretische Unterschiede

HAFAS und PRIMA-Verbindungssuche sind im Ablauf unterschiedlich aufgebaut: HAFAS sucht in der Grundregel nur zwischen zwei angegebenen Bahnhofsknoten eine ab dem angegebenen Zeitpunkt nicht überholte nächste Verbindung und ergänzt sie um weitere Alternativverbindungen, wohingegen PRIMA aus einem Quellbezirk zu allen anderen Zielbezirken alle möglichen und nach Auswahl guten Verbindungen an dem angegebenen Verkehrstag findet.

Dies ist allerdings nicht entscheidend für das Ergebnis. Beide Systeme folgen der gleichen Überholungsregel und führen ein vergleichbares Dominanzkriterium zur Suche nach „sinnvollen“ Verbindungen mit unterschiedlichen Definitionen für „sinnvoll“. Aufgrund derselben Zielsetzung werden auch Gemeinsamkeiten erkennbar. Wenn in einem gegebenen Idealfall auf einer Relation A->B nur streckengleiche Züge benutzbar sind, findet HAFAS und PRIMA immer dieselben Verbindungen, welche nicht überholt werden. Eine überholte Verbindung kann ggf. auch gefunden werden, wenn sie weniger Umstiege hat oder Züge einer niedrigeren Produktklasse einschließt. Dieser Ansatz sieht von der Markierung in HAFAS ab, und nimmt an, dass die nicht überholten Verbindungen das Auswahlkriterium von PRIMA erfüllen.

PRIMA sieht durch das Anlegen von aggregierbaren Tarifpunkten bereits bei der Verbindungssuche einen grob geschätzten Fahrpreis vor. HAFAS kennt im Gegensatz dazu keinen konkreten Fahrpreis einer Verbindung, sucht aber explizit erneut ohne die angeblich teuren Produktklassen. Beide Systeme bilden somit eine vereinfachte Näherung des echten Fahrpreises nach und verwenden dabei unterschiedliche Approximationsmethoden.

Eine Vergleichbarkeit der beiden Datenbanken muss deshalb vorerst hergestellt werden.

2.3.2 Eigenschaften der Daten

Die verwendeten **PRIMA-Daten** beinhalten erst nach Relation, dann nach Verbindung sortierte Reisedaten (einschließlich Quell- und Zielbezirk,

Betriebsnummern und -typ sowie Abfahrts- und Ankunftszeiten der einzelnen Züge an jedem Bahnhof, Anbindungszeiten zum Bezirk) und deren bei der Verkehrsumlegung zugeteilte Nachfrage als **Personenfahrten** (PFahrt). Hierbei werden alle Reiseanlässe (Berufspendler, Wochenendpendler, Geschäftsreisende, Privatreisende eintägig und mehrtägig) des erwarteten Verkehrsaufkommens völlig integriert. Gefundene Verbindungen, welchen keine Nachfrage zugeteilt wird, erscheinen nicht in den Daten.

Da aber nur rückwärts identifiziert werden kann, wie gut die Prognose auf den Ist-Fall zutrifft, werden Vergangenheitsdaten von TReK mit Plan-Daten von PRIMA verglichen.

TReK-EMS-Daten enthalten ausführliche Reiseinformationen sämtlicher gebuchten Regeltickets mit mindestens einem Fernverkehrsanteil aus den meisten Vertriebskanälen. Dazu gehören Fernverkehrsfahrkarten mit oder ohne BahnCard-Rabatt zum Normalpreis, (Kultur-)Sparpreis, Europa-Spezial-Tarif sowie fahrplanbasierte Aktionsangebote der DB Fernverkehr und weitere zuggebundene Fahrkarten einiger Vertriebspartner (L'tur, Busliniensuche usw.). Nicht in den EMS-Daten enthalten sind BahnCard100-Fahrten, Streckenzeitkarten, AiRail-Codeshare-Tickets, RIT, Rail & Fly, Blanko-Tickets deren Strecke und/oder Gültigkeit von Hand eingetragen werden kann (z.B. Lidl-Ticket, Mitfahrerfreifahrt, Freifahrt für Dritte, Mitarbeiter-TagesTicket), da sie (bis auf AiRail) nicht nach einem konkreten Fahrplanangebot ausgestellt werden.

Die aufgeführten Reiseinformationen einer Fahrkarte umfassen Quell- und Zielbezirk der Reise, Betriebsnummern, Ein- und Ausstiegszeiten an jedem Bahnhofsknoten aller benutzten Fernverkehrszüge. Einträge gleicher Kennzahlen werden gruppiert und als **beförderte Personen** (BEFP) aufsummiert. Reine Nahverkehrsfahrten, Berufspendler mit einer DB-Monatskarte oder einer BahnCard100, Fluggäste mit Anschlussbahnfahrt im Fernverkehr mit einem der o.g. Sondertickets werden aufgrund nicht vorhandener Daten in EMS nicht erfasst. Nach systematischem Abgleich mit RES2-Daten aus dem Reisendenerfassungssystem (RES), einem DB-internen Fahrgastzählungssystem, sind 60% der insgesamt beförderten Personen bzw. 70% der Personenkilometer (PKM) in EMS enthalten.

Da EMS-Daten von einer ziemlich großen Verkaufsbasis aufgebaut werden und der Verkauf solcher Tickets ausschließlich über den in HAFAS hinterlegten Fahrplan erfolgt, sind diese Daten eine gute Widerspiegelung der Verbindungssuchergebnisse in HAFAS. Daher spricht diese Arbeit von

buchbaren Verbindungen, die den HAFAS-Verbindungssuchalgorithmus vertreten.

Es ist zu erwähnen, dass Umwegverbindungen sowie Verbindungen mit längerem Zwischenaufenthalt auch buchbar sind, deren beabsichtigte Buchung durch individuelle Einstellung in HAFAS ermöglicht wird. In der großen Summe betrifft dies aber nur einzelne Fälle, während solche Verbindungen vernünftigerweise nicht in PRIMA zu finden und somit für den Vergleich irrelevant sind. Um nicht den eigentlichen Sinn und Fokus zu verlieren und nur die erheblichen Unterschiede herauszufinden, wird in diesem Vergleich nur auf Verbindungen eingeschränkt, die hohe Bedeutung im Sinne von beförderten Personen (TReK-EMS) oder Personenfahrten (PRIMA) in zumindest einem der beiden Systeme aufweisen. Der Einfachheit halber sprechen wir von einer allgemeinen **Nachfrage** der Verbindung.

2.3.3 Vergleichsvorgang

Um die jeweils 50 bis 100 nachfragestärksten Verbindungen herauszufinden, die nur mit einem der beiden Algorithmen aufgebaut werden, werden Vergleiche in zwei Richtungen durchgeführt.

2.3.3.1 Nachfragestärkste EMS-Verbindungen

Vorgehensweise Abgleich Top-EMS -> PRIMA:

- 1) EMS-Daten werden nach beförderten Personen gefiltert, um eine Liste von wichtigsten EMS-Verbindungen zu generieren.
- 2) Für jede EMS-Verbindung dieser Liste werden anschließend die PRIMA-Daten hinsichtlich der vergleichbaren Kennzahlen (Quell-/Zielbezirk, Abfahrts-/Ankunftszeit und Betriebsnummer jedes Fernverkehrszuges) durchsucht und abgeglichen, um einen übereinstimmenden PRIMA-Verbindungseintrag zu finden.
- 3) Sollte kein Eintrag in PRIMA ermittelt werden, ist es sehr möglich, dass diese EMS-Verbindung eine bedeutende Verbindung ist, die nur mit dem HAFAS-Algorithmus aufgebaut wird.
- 4) Durch weitere Analysen kann dies bestätigt oder widerlegt werden.

Als Grundlage von EMS wird ein Datenbankabzug verwendet, dessen Dateninhalt sich auf den Verkehrszeitraum 05.03.2012 - 29.04.2012 (8 ganze Wochen) bezieht, um ein größeres Dateninput zu gewinnen, so dass die Buchungen auf möglichen Verbindungen auch etwas gleichmäßiger verteilt werden. Die PRIMA-Daten stammen aus dem Reisekettenpünktlichkeitsfahrplan RKP-2012_V17_DTg mit Stichtag am 24.04.2012 (Dienstag) nach einer Durchschnittstagsumlegung.

Da der PRIMA-Fahrplan einem festen Fahrplan von einem konkreten Dienstag, dem 24.04.2012, entspricht, während der EMS-Fahrplan eine Vereinigung aller Fahrtmöglichkeiten aus dem 8-wöchigen Zeitraum darstellt, soll durch diesen Abgleich hauptsächlich folgender Unterschied auffallen:

- EMS-Verbindungen, die nicht am Stichtag vom PRIMA-Fahrplan verkehren bzw. abweichende Abfahrts-/Ankunftszeiten oder Zugnummern haben aber große EMS-Nachfrage herbeiführen.
Das gilt besonders für Freitags- und Sonntagsentlastungsfahrten, Winterfahrten (bis 31.03) einiger Züge, Messfahrten (Hannover)

Angesichts der Tatsache, dass ausschließlich der abweichende Fahrplan diese große Menge von unterschiedlichen Verbindungen zur Folge hat, sind sie für den Verbindungssuchalgorithmus irrelevant und werden deshalb im Folgenden nicht berücksichtigt und verworfen.

Beachtenswert sind dagegen:

- Überholte Umsteigeverbindungen, die dank „00“-Markierung erst in der Alternativsuche in HAFAS gefunden werden. Diese werden in PRIMA grundsätzlich nicht mehr gefunden, da sie aus Sicht von PRIMA weder komfortabel (in Bezug auf Umstiege) noch schnell oder günstig sind.
Erklärendes Beispiel: 1) Durch einen ICE-Sprinter überholte ICE-Umsteigeverbindungen zwischen Frankfurt und Berlin. 2) IC-Fahrten auf der Rheinstrecke mit Umstieg in Mannheim von/auf ICE zwischen Stuttgart und Köln
- Durch eine Direktverbindung überholte Direktverbindungen, die dank „01“-Markierung oder Preiswertsuche erst gefunden werden. Diese werden in PRIMA nicht gefunden, wenn sie in PRIMA bei der Bewertung viel zu langsam sind oder einen zu hohen Suchwiderstand aufweisen.
Erklärendes Beispiel: 1) Frankfurt – München, durch einen ICE der Linie 41 überholte ICE-Direktverbindung der Linie 42 über Stuttgart. 2) München – Berlin, durch einen ICE überholte IC-Direktverbindung um 07:10

2.3.3.2 Nachfragestärkste PRIMA-Verbindungen

Vorgehensweise Abgleich Top-PRIMA -> EMS:

- 1) PRIMA-Verbindungsinformationen ohne Nahverkehrsteilwege (Quell-/Zielbezirk, Bahnhofsknoten mit Abfahrts-/Ankunftszeit und die Betriebsnummer jedes Fernverkehrszuges) und deren zugeteilte Nachfrage sowie Relationsnachfrage in PRIMA werden den jeweils entsprechenden EMS-Datensätzen zugeordnet und nebeneinander gestellt (*Match*), um eine zusammengefügte Datenbank mit vergleichbaren Einträgen aus beiden Systemen herzustellen.
- 2) Nicht zugeordnete Datensätze aus einem System werden mit leeren Daten aus dem anderen System ergänzt. Jeder Datensatz ist eine Verbindung. Wenn er keine PRIMA-Information enthält, so kann dessen Verbindung nicht in PRIMA gefunden werden. Wenn er keine EMS-Information enthält, gibt es zwei Möglichkeiten: entweder wird die Verbindung nicht in HAFAS gefunden oder sie wurde aus verschiedenen Gründen am Stichtag nicht von einem Kunden gebucht.
- 3) Daten (Verbindungen) mit leerem EMS-Eintrag werden nach PRIMA-Personenfahrten gefiltert, um eine Liste von wichtigsten PRIMA-Verbindungen zu generieren, die nicht in EMS nachzuweisen sind.
- 4) Durch weitere Analysen kann bestätigt werden, ob diese Verbindungen nur mit dem PRIMA-Algorithmus aufbaubar sind.

Im Gegensatz zu 2.3.3.1 wird in diesem Abgleich ein Datenbankabzug als Grundlage von EMS verwendet, dessen Dateninhalt sich auf genau den Verkehrstag des PRIMA-Fahrplans (24.04.2012) bezieht. Die PRIMA-Daten entstammen derselben Quelle wie im vorigen Abgleich.

Folgende Probleme können bei diesem Abgleich auftreten und somit die Analyse beeinträchtigen:

- Verbindungen mit mindestens einem fremden Fernverkehrszug außerhalb des DB-Tarifs (Thalys, InterConnex, HKX) können zwar sowohl in PRIMA als auch in HAFAS gefunden werden, sind aber nicht aus den EMS-Daten ermittelbar, da sie nicht buchbar sind.
- Zwischen zwei nah liegenden (etwa bis 100 km) Bezirken, auf denen überwiegend Pendler mit Streckenzeitkarten fahren, können zwar Verbindungen in PRIMA und HAFAS gefunden werden, werden diese aber häufig nicht aus den EMS-Daten ermittelt, wenn kein Reisender gebucht hat.

2.4 Aufnahme und Auswertung der Unterschiede

2.4.1 Ableitung von Top-EMS-Verbindungen ohne PRIMA-Eintrag

Nach der im Abschnitt 2.3.3.1 beschriebenen Vorgehensweise wird der erste Vergleich in Richtung EMS->PRIMA durchgeführt. Im Schritt 1) werden wichtigste EMS-Verbindungen in zwei Gruppen unterteilt:

- a. **Segment I:** EMS-Verbindungen mit tagesdurchschnittlichen beförderten Personen von über 10.
- b. **Segment II:** EMS-Verbindungen mit tagesdurchschnittlichen beförderten Personen von über 4 bis 10.

Jede Gruppe durchläuft unabhängig von der anderen die Schritte 2) bis 3). In PRIMA nicht gefundene EMS-Verbindungen werden dokumentiert und folgendermaßen analysiert.

2.4.1.1 Segment I (BEFP > 10)

Aus insgesamt rund 2000 wichtigsten gebuchten EMS-Verbindungen im Segment I sind rund 60 Verbindungen in PRIMA nicht wiederfindbar, welche näher in folgende Kategorien eingeordnet werden können:

- Zug verkehrt nicht am Stichtag des PRIMA-Fahrplans: 25
- Zug stammt aus Hannover-Messe-Fahrten, die wegen der kurzfristigen Aufnahme nicht im PRIMA-Fahrplan hinterlegt sind: 8
- Nachtzug stammt aus einem anderen Fahrplanstand als in PRIMA hinterlegt und ist deshalb nicht zu finden: 4
- Reisende steigen an einem anderen Start/Zielbahnhof des Bezirks ein/aus als in PRIMA nachgebildet: 18
- Zug wird ausschließlich wegen Markierung in HAFAS gefunden: 2

Da die aus einem größeren Zeitraum stammenden EMS-Daten auch Züge einschließen, die nicht im PRIMA-Fahrplan (24.04.2012) enthalten sind, sind dadurch entstandene Unterschiede für den Vergleich von Suchalgorithmen irrelevant. Die Darstellung der Anzahl solcher Abweichungen dient lediglich der Veranschaulichung der Bedeutsamkeit von anderen interessanten Unterschieden. Daher werden nur die letzten zwei Kategorien weiter ausgewertet.

Anbindungsabhängige Unterschiede

Einige größere Städte verfügen innerhalb des Stadtbezirks über mehrere Fernverkehrsbahnhöfe, die ein Fernverkehrszug zum Einstieg oder Ausstieg von Fahrgästen nacheinander anfährt. Dies betrifft u. a. folgende Städte und deren Fernverkehrshalte in einer Kette:

- Berlin: Gesundbrunnen->Hauptbahnhof->Südkreuz oder Ostbahnhof->Hauptbahnhof->Spandau oder auch Spandau->Hauptbahnhof->Südkreuz
- Hamburg: Altona->Dammtor->Hauptbahnhof->Harburg
- München: Pasing->Hauptbahnhof->Ostbahnhof
- Dresden: Hauptbahnhof->Neustadt
- Karlsruhe: Hauptbahnhof->Durlach

Daneben existieren auch populäre Urlaubsziele mit Fernverkehrsbahnhöfen, die zu einem selben übergeordneten Bezirk gehören:

- Nordfriesland: Westerland(Sylt)->Niebüll->Husum

Alle o.g. Fernverkehrsknoten sind in HAFAS als Start- oder Zielbahnhof anwählbar. In PRIMA sind sie mit unterschiedlicher Anbindungszeit nach Entfernung und Bedeutung an den jeweiligen Bezirk angebunden, wobei in einer konkreten Verbindungssuche meistens nur ein Knoten als Einstiegs- oder Ausstiegsbahnhof überlebt, weil die Verbindungen ab oder bis zu den anderen Knoten in den meisten Fällen überholt werden und einen höheren Suchwiderstand erzeugen.

Kunden, die aufgrund ihrer näheren Lage an einem anderen Fernverkehrsknoten als in PRIMA diese alternative Verbindung buchen, hinterlassen in den EMS-Daten eine von PRIMA abweichende Verbindung. Wenn die Summe solcher Kunden groß genug ist, wird die besagte Verbindung in diesem Segment sichtbar. Diese Ursache machen etwa 30% der größten Unterschiedsverbindungen zwischen HAFAS und PRIMA aus.

Unter den 18 unterschiedlichen Verbindungen in diesem Segment, die durch die Anbindung verursacht werden, starten 5 in Hamburg, 11 in Berlin, jeweils 1 in Dresden und Nordfriesland.

Bei Fernverkehrsfahrten aus Hamburg in Richtung Süden können in PRIMA Verbindungen ab Altona und Dammtor nicht gefunden werden, wobei in Richtung Norden nur Verbindungen ab Dammtor in PRIMA zu finden sind, wenn der benutzte Zug dort anhält. Grund dafür ist die in Dammtor kürzere Anbindungszeit als Hauptbahnhof (27 Min vgl. 30 Min) und der längere

Zeitaufwand von Dammtor nach Hauptbahnhof (Abfahrt bis Abfahrt mind. 6 Min) als Anbindungszeitunterschied ($30 - 27 = 3$ Min).

Bei Fernverkehrsfahrten aus Berlin in Richtung Süden findet PRIMA dagegen bis auf einige Ausnahmen Verbindungen sowohl ab Hauptbahnhof als auch ab Südkreuz, während in Richtung Westen nur Verbindungen ab Hauptbahnhof in PRIMA zu finden sind. Grund dafür ist der kürzere Zeitaufwand von Hauptbahnhof nach Südkreuz (Abfahrt bis Abfahrt 7 Min) als deren Anbindungszeitunterschied (8 Min). Da Spandau in PRIMA nicht an Berlin angebunden ist, sind auch EMS-Verbindungen mit Start/Ziel Berlin-Spandau in/aus Richtung Hamburg in dieser Kategorie vertreten.

In Segment II sind auch zahlreiche ähnliche Unterschiede wegen Anbindungszeit zu finden. Da es sich um das gleiche Prinzip handelt, werden sie im nächsten Abschnitt nicht mehr ausführlich diskutiert, sondern zum Zweck der Bedeutungsdarstellung dieses Problems nur deren Anzahl gelistet.

Markierungsabhängige Unterschiede

Wie bereits im Abschnitt 2.2.3.1 beschrieben, können einzelne Züge streckenabschnittsweise markiert werden mit der Idee, voraussichtlich volle Züge dadurch zu entlasten oder eine umsteigefreie bzw. kostengünstige Alternativverbindung nicht wegen der Überholung zu unterdrücken.

1) Eine EMS-Verbindung mit tagesdurchschnittlichen BEFP über 10, die nicht in PRIMA gefunden wird aber dank Markierung in HAFAS erscheint, ist folgende Verbindung $con_{F-S-M(691)}$

Frankfurt(M) Hbf(ab 17:50) -> [VIA Stuttgart] -> München Hbf(an 21:29), ICE 691

Sie wird durch eine andere Verbindung $con_{F-N-M(725)}$ in PRIMA überholt und dominiert, weil $con_{F-S-M(691)}$ in keiner Hinsicht besser als $con_{F-N-M(725)}$ ist:

Frankfurt(M) Hbf(ab 17:54) -> [VIA Nürnberg] -> München Hbf(an 21:09), ICE 725

D.h. das Dominanzkriterium wird erfüllt:

- $con_{F-N-M(725)}$ liegt innerhalb des Zeitintervalls von $con_{F-S-M(691)}$
- Umsteigehäufigkeit($con_{F-N-M(725)}$) = Umsteigehäufigkeit($con_{F-S-M(691)}$)
- Suchwiderstand($con_{F-N-M(725)}$) < Suchwiderstand($con_{F-S-M(691)}$)

Grund dafür, dass in HAFAS *con*_{F-S-M(691)} am 24.04.2012 gefunden wird, ist die „01“-Markierung des Zuges ICE 725 im Abschnitt Frankfurt(M) Hbf -> Aschaffenburg Hbf.

*con*_{F-S-M(691)} hat eine Buchungsquote von 10,0892 BEFP/Tag. Betrachtet man aber die Gegebenheit, dass HAFAS nur an einzelnen Tagen der Woche den genannten Abschnitt „01“-markiert, so dass HAFAS auch nur für die Reiseauskunft an diesen Tagen die überholte Verbindung *con*_{F-S-M(691)} findet und anzeigt, ist die Bedeutung bzw. verkehrlicher Einfluss dieser Alternativverbindung noch höher als die Quote dies darstellt.

2) Eine weitere EMS-Verbindung, die nicht in PRIMA gefunden wird aber dank Markierung in HAFAS erscheint, ist die folgende Verbindung *con*_{D-[F]-MA(721,377)}

Düsseldorf Hbf(ab 14:21) -> [Umstieg Frankfurt(M) Hbf(an 15:48)], ICE 721
[Weiterfahrt Frankfurt(M) Hbf(ab 16:05)] -> **Mannheim Hbf**(an 16:42), ICE 377

Sie wird durch eine andere Verbindung *con*_{D-[KJ]-MA(2217,209)} in PRIMA überholt und dominiert, weil *con*_{D-[F]-MA(721,377)} in keiner Hinsicht besser als *con*_{D-[KJ]-MA(2217,209)} ist:

Düsseldorf Hbf(ab 14:27) -> [Umstieg Köln Hbf(an 14:50)], IC 2217
[Weiterfahrt Köln Hbf(ab 14:55)] -> **Mannheim Hbf**(an 16:24), ICE 209

Zug ICE 209 ist im Abschnitt Frankfurt(M) Flughafen -> Mannheim Hbf „01“-markiert.

2.4.1.2 Segment II ($4 < BEFP \leq 10$)

Aus den nächsten 4675 wichtigsten gebuchten EMS-Verbindungen im Segment II sind (abgesehen von nicht verkehrenden Zügen am Stichtag von PRIMA) etwa 110 Verbindungen in PRIMA nicht identisch. EMS-Verbindungen mit Bezug zu nicht verkehrenden Zügen am Stichtag von PRIMA wurden bereits ignoriert, da diese eine sehr große Menge bildet und wie bereits erklärt für unseren Vergleich irrelevant sind. Die restlichen Unterschiede können wie folgt in Kategorien eingeordnet werden:

- Zug stammt aus Hannover-Messe-Fahrten, die wegen der kurzfristigen Aufnahme nicht im PRIMA-Fahrplan hinterlegt sind: 32

- Nachtzug/Nahverkehrszug stammt aus einem anderen Fahrplanstand als in PRIMA hinterlegt und ist deshalb nicht zu finden: 16
- Reisende steigen an einem anderen Start/Zielbahnhof des Bezirks ein/aus als in PRIMA nachgebildet: mehr als 30
- Fehlende IC-Aufnahme in PRIMA: 2 (Betrifft IC 2315)
- Reisekette der Verbindung wird ausschließlich wegen Überschreibung der Übergangszeit in einem der beiden Systeme gebrochen oder ermöglicht: 11
- Zug wird ausschließlich wegen Markierung oder preiswerter Verbindungssuche in HAFAS gefunden in HAFAS: 18
- Andere Gründe: 5

Hiervon werden nur die letzten drei Kategorien ausgewertet.

Unterschiede aufgrund von Überschreibung der Übergangszeit

7 EMS-Verbindungen zwischen München und Freiburg können in PRIMA nicht zugeordnet werden, weil der in PRIMA übliche Umstieg in HAFAS wegen Überschreibung der Mindestübergangszeit mit einem höheren Wert ausgeschlossen wird und HAFAS deswegen andere Alternativverbindungen mit längerem Übergang in Mannheim als Ersatz findet und anzeigt.

1) HAFAS findet z.B. folgende zwei Verbindungen:

Freiburg(B) Hbf(ab 09:49) -> [Umstieg Mannheim Hbf(an 11:14)], ICE 76
[Weiterfahrt Mannheim Hbf(ab 11:32)] -> **München Hbf(an 14:27)**, ICE 515

Freiburg(B) Hbf(ab 09:57) -> [Umstieg Karlsruhe Hbf(an 10:58)], ICE 600
[Weiterfahrt Karlsruhe Hbf(ab 11:06)] -> [Umstieg Stuttgart Hbf(an 12:03)], IC 2067
[Weiterfahrt Stuttgart Hbf(ab 12:12)] -> **München Hbf(an 14:27)**, ICE 515

PRIMA findet im Gegensatz dazu nur die Verbindung mit dem unmittelbaren Umstieg in Mannheim Hbf, die in HAFAS ausgeschlossen wird:

Freiburg(B) Hbf(ab 09:57) -> [Umstieg Mannheim Hbf(an 11:23)], ICE 600
[Weiterfahrt Mannheim Hbf(ab 11:32)] -> **München Hbf(an 14:27)**, ICE 515

2) Umgekehrt, 2 EMS-Verbindungen mit kurzem Umstieg in Nürnberg bzw. Dortmund können in PRIMA nicht zugeordnet werden, weil der kurze Umstieg, der den Übergang in HAFAS ermöglicht, unterhalb der

Bahnknotenanzahl liegt aber nicht in der Übergangsliste von PRIMA aufgenommen wurde.

3) Für den Umstieg in Nürnberg findet HAFAS folgende Verbindung:

München Hbf(ab 15:16) -> [Umstieg Nürnberg Hbf(an 16:30)], ICE 584
[Weiterfahrt Nürnberg Hbf(ab 16:34)] -> **Berlin Hbf(an 21:21)**, ICE 1104

PRIMA kann diese Verbindung nicht herstellen und findet dafür:

München Hbf(ab 14:50) -> [Umstieg Nürnberg Hbf(an 15:57)], ICE 622
[Weiterfahrt Nürnberg Hbf(ab 16:34)] -> **Berlin Hbf(an 21:21)**, ICE 1104

und

München Hbf(ab 15:16) -> [Umstieg Göttingen(an 18:54)], ICE 584
[Weiterfahrt Göttingen(ab 19:03)] -> **Berlin Hbf(an 21:25)**, ICE 370

4) Für den Umstieg in Dortmund findet HAFAS folgende Verbindung:

Münster(W) Hbf(ab 08:01) -> [Umstieg Dortmund Hbf(an 08:33)], ICE 515
[Weiterfahrt Dortmund Hbf(ab 08:37)] -> **Düsseldorf Hbf(an 09:25)**, ICE 27

PRIMA kann diese Verbindung nicht herstellen und findet dafür:

Münster(W) Hbf(ab 08:01) -> [Umstieg Dortmund Hbf(an 08:33)], ICE 515
[Weiterfahrt Dortmund Hbf(ab 08:44)] -> **Düsseldorf Hbf(an 09:37)**, RE 10112

Unterschiede aufgrund von „00“-Markierungen

Einige wichtige Verbindungen z.B. über die langsamere Rheinstrecke, die nicht in PRIMA gefunden werden können, sind in EMS-Daten dagegen mit hoher Buchungsquote nachzuweisen. HAFAS findet diese Verbindungen ausschließlich aufgrund von „00“-Markierung der Züge über die Schnellfahrstrecke Köln-Rhein-Main.

(Bei der Auflistung sind die in PRIMA nicht gefundenen HAFAS-Verbindungen unterstrichen)

1) HAFAS findet gleichzeitig:

Köln Hbf(ab 18:53) -> [Umstieg Mannheim Hbf(an 21:21)], IC 2213
[Weiterfahrt Mannheim Hbf(ab 21:32)] -> **Stuttgart Hbf(an 22:08)**, ICE 615

Köln Hbf(ab 18:55) -> [Umstieg Mannheim Hbf(an 20:24)], ICE 603
[Weiterfahrt Mannheim Hbf(ab 20:30)] -> **Stuttgart Hbf(an 21:08)**, ICE 693

PRIMA findet nur *con* K-[MA]-S(603,693).

con F-[MA]-S(2213,615) hat eine Buchungsquote von 4,1785 BEFP/Tag.

2) HAFAS findet gleichzeitig:

Frankfurt(M) Hbf(ab 06:18) -> [Umstieg Naumburg(S) Hbf(an 09:23)], IC 2153
[Weiterfahrt Naumburg(S) Hbf(ab 09:29)] -> **Berlin Hbf(an 11:16)**, ICE 1514

Frankfurt(M) Hbf(ab 06:58) -> [Umstieg Hannover Hbf(an 09:17)], ICE 774
[Weiterfahrt Hannover Hbf(ab 09:31)] -> **Berlin Hbf(an 11:11)**, ICE 543

PRIMA findet nur *con* F-[H]-B(774,543).

con F-[NMB]-B(2153,1514) hat eine Buchungsquote von 4,6785 BEFP/Tag.

3) HAFAS findet gleichzeitig:

Köln Hbf(ab 07:53) -> [VIA Bonn / Nürnberg] -> **München Hbf(an 13:51)**, ICE 1021

Köln Hbf(ab 07:55) -> [VIA SFS-KRM / Stuttgart] -> **München Hbf(an 12:27)**, ICE 513

PRIMA findet nur *con* K-S-M(513).

con K-BN-N-M(1021) hat eine Buchungsquote von 7,5714 BEFP/Tag.

4) HAFAS findet gleichzeitig:

Köln Hbf(ab 07:53) -> [VIA Bonn] -> **Nürnberg Hbf(an 12:24)**, ICE 1021

Köln Messe/Deutz(ab 08:44) -> [VIA SFS-KRM] -> **Nürnberg Hbf(an 11:59)**, ICE 529

PRIMA findet nur *con* K-KRM-N(529).

con K-BN-N(1021) hat eine Buchungsquote von 4,1964 BEFP/Tag.

5) HAFAS findet gleichzeitig:

Köln Hbf(ab 07:53) -> [VIA Bonn] -> Frankfurt(M) Hbf(an 10:13), ICE 1021

**Köln Hbf(ab 07:55) -> [Umstieg Frankfurt(M) Flughafen(an 08:51)], ICE 513
[Weiterfahrt Frankfurt(M) Flughafen(ab 09:01)] -> Frankfurt(M) Hbf(an 09:12), ICE
1597**

Köln Hbf(ab 08:20) -> [VIA SFS-KRM] -> Frankfurt(M) Hbf(an 09:24), ICE 11

PRIMA findet nur *con* K-KRM-N(529).

con K-BN-F(1021) hat eine Buchungsquote von 6,6785 BEFP/Tag.

6) HAFAS findet gleichzeitig:

Köln Hbf(ab 11:53) -> [VIA Bonn] -> Frankfurt(M) Hbf(an 14:12), ICE 1023

**Köln Hbf(ab 11:55) -> [Umstieg Frankfurt(M) Flughafen(an 12:51)], ICE 517
[Weiterfahrt Frankfurt(M) Flughafen(ab 13:02)] -> Frankfurt(M) Hbf(an 13:13), ICE
1651**

Köln Hbf(ab 12:20) -> [VIA SFS-KRM] -> Frankfurt(M) Hbf(an 13:30), ICE 15

PRIMA findet nur *con* K-[F-Flug]-F(517,1651) und *con* K-KRM-F(15).

con K-BN-F(1023) hat eine Buchungsquote von 9,3928 BEFP/Tag.

In einem ähnlichen Fall, findet HAFAS:

Köln Hbf(ab 09:53) -> [VIA Bonn] -> Frankfurt(M) Hbf(an 12:13), ICE 27

**Köln Hbf(ab 09:55) -> [Umstieg Frankfurt(M) Flughafen(an 10:51)], ICE 515
[Weiterfahrt Frankfurt(M) Flughafen(ab 11:02)] -> Frankfurt(M) Hbf(an 11:13), ICE
1559**

Köln Hbf(ab 10:19) -> [VIA SFS-KRM] -> Frankfurt(M) Hbf(an 11:48), ICE 623

wobei PRIMA auch genau alle diese 3 Fahrten findet.

Mögliche Erklärung: *con* K-BN-F(27) kann deshalb auch in PRIMA gefunden werden, weil die überholende Direktverbindung *con* K-KRM-F(623) über Flughafen Köln/Bonn fährt und bei der PRIMA-Preisbildung zusätzliche Tarifpunkte

einschließt. Zusammen mit der verlängerten Fahrzeit stellt dies einen größeren Suchwiderstand dar gegenüber der Verbindung mit Abfahrt vor 2 Stunden oder nach 2 Stunden.

7) HAFAS findet gleichzeitig:

Köln Hbf(ab 19:53) -> [VIA Bonn] -> Frankfurt(M) Hbf(an 22:13), IC 2029

**Köln Hbf(ab 19:57) -> [Umstieg Frankfurt(M) Flughafen(an 20:51)], ICE 615
[Weiterfahrt Frankfurt(M) Flughafen(ab 21:02)] -> Frankfurt(M) Hbf(an 21:13), ICE
1659**

Köln Hbf(ab 20:05) -> [VIA Bonn] -> Frankfurt(M) Hbf(an 22:43), EN 421

Köln Hbf(ab 20:28) -> [VIA SFS-KRM] -> Frankfurt(M) Hbf(an 21:30), ICE 19

PRIMA findet nur *con*_{K-[F-Flug]-F(615,1659)}, *con*_{K-BN-F(421)} und *con*_{K-KRM-F(19)}.

*con*_{K-BN-F(2029)} hat eine Buchungsquote von 6,6785 BEFP/Tag.

Mögliche Erklärung: Dass die erste Rhein-Verbindung *con*_{K-BN-F(2029)} in PRIMA nicht gefunden werden kann, während dagegen aber die zweite noch langsamere Rhein-Verbindung *con*_{K-BN-F(421)} trotz vergleichbarer Überholung durch *con*_{K-KRM-F(19)} gefunden wird, liegt daran, dass für den Nachtzug EN 421 in PRIMA pro Tarifkilometer viel niedrigere Preiseinheit (siehe 2.2.1) gegenüber dem normalen IC angerechnet wird, wobei HAFAS in der Regel nicht zwischen IC und Nachtzug unterscheidet. Demzufolge hat *con*_{K-BN-F(2029)} einen höheren Suchwiderstand als die überholende Direktverbindung *con*_{K-KRM-F(19)} und wird in PRIMA somit dominiert.

8) HAFAS findet gleichzeitig:

**Frankfurt(M) Hbf(ab 19:44) -> [Umstieg Frankfurt(M) Flughafen(an 19:56)], ICE
1022**

[Weiterfahrt Frankfurt(M) Flughafen(ab 20:09)] -> Köln Hbf(an 21:05), ICE 502

Frankfurt(M) Hbf(ab 19:44) -> [VIA Bonn] -> Köln Hbf(an 22:05), ICE 1022

**Frankfurt(M) Hbf(ab 20:10) -> [VIA SFS-KRM] -> Köln Messe/Deutz(an 21:13), ICE
528**

PRIMA findet nur *con*_{F-[F-Flug]-K(1022,502)}, und *con*_{F-KRM-K(528)}.

*con*_{F-BN-K(1022)} hat eine Buchungsquote von 5,0714 BEFP/Tag.

Unterschiede aufgrund von „01“-Markierungen

2 weitere EMS-Verbindungen mit tagesdurchschnittlichen BEFP zwischen 4 und 10, die PRIMA nicht findet, werden dank „01“-Markierung von HAFAS gefunden.

1) HAFAS findet gleichzeitig:

Frankfurt(M) Hbf(ab 13:50) -> [VIA Stuttgart] -> München Hbf(an 17:27), ICE 597

Frankfurt(M) Hbf(ab 13:54) -> [VIA Nürnberg] -> München Hbf(an 17:10), ICE 627

PRIMA findet nur *con*_{F-N-M(627)}.

*con*_{F-S-M(597)} ist eigentlich in keiner Hinsicht besser als *con*_{F-N-M(627)}. Grund dafür, dass HAFAS *con*_{F-S-M(597)} an einigen Tagen (Fr/So, z.B. am 20.04.2014) findet, ist die „01“-Markierung des überholenden Zuges ICE 627 im Abschnitt Frankfurt(M) Hbf -> Würzburg Hbf.

*con*_{F-S-M(597)} hat eine Buchungsquote von 4,6071 BEFP/Tag. Wenn man aber die Gegebenheit betrachtet, dass HAFAS nur an einzelnen Tagen der Woche den genannten Abschnitt „01“-markiert, so dass HAFAS auch nur für die Reiseauskunft an diesen Tagen die überholte Verbindung *con*_{F-S-M(597)} findet und anzeigt, ist die Bedeutung bzw. verkehrlicher Einfluss dieser Alternativverbindung noch höher als die Quote dies darstellt.

2) HAFAS findet gleichzeitig:

Frankfurt(M) Hbf(ab 15:50) -> [VIA Stuttgart] -> München Hbf(an 19:27), ICE 599

Frankfurt(M) Hbf(ab 15:54) -> [VIA Nürnberg] -> München Hbf(an 19:10), ICE 721

PRIMA findet nur *con*_{F-N-M(721)}.

Grund dafür, dass HAFAS *con*_{F-S-M(599)} an einigen Tagen (Do, z.B. am 26.04.2014) findet, ist die „01“-Markierung des überholenden Zuges ICE 721 im Abschnitt Frankfurt(M) Hbf -> Würzburg Hbf.

con_{F-S-M(599)} hat eine Buchungsquote von 5,9821 BEFP/Tag. Wenn man aber die Gegebenheit betrachtet, dass HAFAS nur an einzelnen Tagen der Woche den genannten Abschnitt „01“-markiert, so dass HAFAS auch nur für die Reiseauskunft an diesen Tagen die überholte Verbindung con_{F-S-M(599)} findet und anzeigt, ist die Bedeutung bzw. verkehrlicher Einfluss dieser Alternativverbindung noch höher als die Quote dies darstellt.

Unterschiede aufgrund von Sonderzug-Markierungen

2 wichtige EMS-Verbindungen, die PRIMA nicht findet, sind in dieser Kategorie betroffen. Es handelt sich um einen schnellen ICE-Zug, der den anderen langsameren ICE-Zug mit mehr Zwischenhalten überholt, wobei beide Züge ihre Fahrt auf der gleichen Schnellfahrstrecke absolvieren. Die fahrplanmäßige Überholung findet in Montabaur statt.

1) HAFAS findet gleichzeitig:

Frankfurt(M) Hbf(ab 18:16) -> [VIA SFS-KRM] -> Köln Hbf(an 19:56), ICE 812

Frankfurt(M) Hbf(ab 18:29) -> [VIA SFS-KRM] -> Köln Hbf(an 19:39), ICE 10

PRIMA findet nur con_{F-KRM-K(10)}.

con_{F-KRM-K(812)} hat eine Buchungsquote von 6,9464 BEFP/Tag.

con_{F-KRM-K(812)} ist in keiner Hinsicht besser als con_{F-KRM-K(10)}. In den HAFAS-Rohdaten wird ICE 812 als Entlastungszug für ICE 10 registriert. Die langsamere Verbindung hat in den meisten Fällen einen großen Preisvorteil gemäß Sparpreis-Kontingent.

2) HAFAS findet gleichzeitig:

Frankfurt(M) Hbf(ab 18:16) -> [VIA SFS-KRM] -> Siegburg/Bonn(an 19:27), ICE 812

Frankfurt(M) Hbf(ab 18:29) -> [VIA SFS-KRM] -> Siegburg/Bonn(an 19:21), ICE 10

PRIMA findet nur con_{F-KRM-SU(10)}.

con_{F-KRM-SU(812)} hat eine Buchungsquote von 7,75 BEFP/Tag.

Unterschiede aufgrund der Preiswertsuche

Insgesamt 5 wichtige überholte EMS-Verbindungen, die bei der Preiswertsuche in HAFAS zusätzlich gefunden werden, werden von PRIMA nicht als vernünftige Reisemöglichkeit vorgeschlagen.

1) HAFAS findet gleichzeitig:

Frankfurt(M) Hbf(ab 08:22) -> [VIA Stuttgart] -> München Hbf(an 12:11), EC 1113

Frankfurt(M) Hbf(ab 08:54) -> [VIA Nürnberg] -> München Hbf(an 12:08), ICE 527

PRIMA findet nur *con*_{F-N-M(527)}.

con_{F-S-M(1113)} hat eine Buchungsquote von 6,2142 BEFP/Tag. Diese in PRIMA fehlende Verbindung hätte aufgrund des Umweges und der längeren Fahrtzeit trotz niedriger Produktklasse schließlich einen geringfügig höheren Suchwiderstand gegenüber *con*_{F-N-M(527)} und wirkt deshalb in PRIMA als dominiert. Dieselbe Ursache gilt auch für das Fehlen der nachfolgenden 4 Verbindungen in PRIMA.

2) HAFAS findet gleichzeitig:

Frankfurt(M) Hbf(ab 12:20) -> [VIA Stuttgart] -> München Hbf(an 16:11), EC 219

Frankfurt(M) Hbf(ab 12:54) -> [VIA Nürnberg] -> München Hbf(an 16:09), ICE 625

PRIMA findet nur *con*_{F-N-M(625)}.

con_{F-S-M(219)} hat eine Buchungsquote von 7,0535 BEFP/Tag.

3) HAFAS findet gleichzeitig:

Frankfurt(M) Hbf(ab 14:20) -> [VIA Stuttgart] -> München Hbf(an 18:11), EC 117

Frankfurt(M) Hbf(ab 14:54) -> [VIA Nürnberg] -> München Hbf(an 18:09), ICE 629

PRIMA findet nur *con*_{F-N-M(629)}.

con_{F-S-M(117)} hat eine Buchungsquote von 4,7678 BEFP/Tag.

4) HAFAS findet gleichzeitig:

Frankfurt(M) Hbf(ab 16:20) -> [VIA Stuttgart] -> München Hbf(an 20:11), EC 391

Frankfurt(M) Hbf(ab 16:54) -> [VIA Nürnberg] -> München Hbf(an 20:09), ICE 723

PRIMA findet nur *con_{F-N-M(723)}*.

con_{F-S-M(391)} hat eine Buchungsquote von 7 BEFP/Tag.

5) HAFAS findet gleichzeitig:

München Hbf(ab 07:10) -> [VIA Augsburg] -> Berlin Hbf(an 14:06), IC 2208

München Hbf(ab 07:16) -> [VIA Ingolstadt] -> Berlin Hbf(an 13:09), ICE 1512

PRIMA findet nur *con_{M-IN-B(1512)}*.

con_{M-A-B(2208)} hat eine Buchungsquote von 7,4821 BEFP/Tag.

2.4.2 Ableitung von Top-PRIMA-Verbindungen ohne EMS-Eintrag

Nach der im Abschnitt 2.3.3.2 beschriebenen Herangehensweise wird der zweite Vergleich in Richtung PRIMA->EMS durchgeführt.

Da es sich in diesem Vergleich um denselben Fernverkehrsfahrplanstand eines gleichen Verkehrstages handelt, sind Unterschiede in reinen Fernverkehrsverbindungen, die auf das abweichende Zugfahrangebot zurückzuführen sind, ausgeschlossen. Auf den Fahrplan zurückführende Unterschiede können allerdings bei gemischten Nah- und Fernverkehrsverbindungen weiterhin einschreiten.

Aus insgesamt 3776 wichtigsten PRIMA-Verbindungen gemäß zugewiesenen Personenfahrten sind etwa 350 Verbindungen ohne EMS-Eintrag. Für Verbindungen mit genau einem Fernverkehrszug, **FV-Direktverbindung** genannt, werden hierbei nur Relationen betrachtet, auf denen die EMS-Relationsnachfrage größer als 100 ist, sprich Relation ist von großer

Bedeutung und über EMS buchbar. Für Verbindungen mit mehr als einem Fernverkehrszug, **FV-Umsteigeverbindung** genannt, werden hingegen alle Relationen einbezogen, denn der Einfluss des Umsteigefaktors soll auch in beiden Systemen angeglichen werden.

Außer den uns bereits bekannten Reisekettenabbrüchen im Umstieg wegen Übergangszeitüberschreitung an bestimmten Bahnhöfen sowie unterschiedlichen Anbindungsmöglichkeiten innerhalb eines Bezirkes, wurden auch weitere Gründe auffällig.

Auf Pendlerrelationen

Ein großer Teil der unterschiedlichen Verbindungen, die durch diesen Abgleich festgestellt wurden, stehen in Zusammenhang mit Relationen, auf denen überwiegend Pendler fahren. Häufig fahrende Pendler mit einer Streckenzeitkarte oder BahnCard100 können nicht in von EMS erfasst werden. Ihre gewählten Verbindungen sind dementsprechend nicht in den EMS-Daten enthalten. Eindeutige Pendlerrelationen, auf welche sich Pendler verteilen und auf vielen deren Verbindungen zu Hauptberufszeiten kein EMS-Ticket ausgestellt wird, sind:

- *Düsseldorf – Duisburg*
- *Düsseldorf – Essen*
- *Köln – Rhein-Sieg-Kreis (Siegburg/Bonn)*
- *Köln – Bonn*
- *Frankfurt a. M. – Limburg-Weilburg (Limburg Süd)*
- *Kassel Stadt – Göttingen*
- *München Stadt – Rosenheim Stadt*
- *München Stadt – Rosenheim Landkreis (Prien a Chiemsee)*
- *München Stadt – Augsburg*

Da auch HAFAS alle davon betroffenen Verbindungen findet, werden sie in dieser Arbeit nicht mehr extensiv diskutiert. Ihre Schilderung dient lediglich als eine Einschätzung des Anteils der EMS-buchenden Reisenden gegenüber dem gesamten Reisebedarf auf Kurzrelationen.

Nichtbuchbare Fernverkehrszüge außerhalb des DB-Tarifs

Einige eigenwirtschaftlich betriebene Fernverkehrszüge außerhalb des DB-Konzerns sind sowohl im HAFAS- als auch im PRIMA-Fahrplan integriert und

können bei der Verbindungssuche entweder als alleinstehende oder Teil einer gesamten Verbindung berücksichtigt werden. Dies betrifft in der Realität Thalys und InterConnex (seit 2013 auch HKX). Da der Verkauf dieser Verbindungen nicht über ein DB-System erfolgt, sind sie nicht in den EMS-Daten enthalten, während PRIMA die Buchbarkeit einer Verbindung bei der Umlegung nicht überprüft. D.h. diese Verbindungen selbst können zwar auch in HAFAS gefunden werden, wird ihnen aber nur in PRIMA eine Nachfrage zugeteilt, so dass sie bei dem Vergleich zwischen EMS- und PRIMA-Daten als Unterschiede erscheinen.

Unterschiede aufgrund des Algorithmus

Es wurden nur 4 bedeutsame PRIMA-Verbindungen ohne EMS-Eintrag bei dem Abgleich eingefangen, bei denen es sich ausschließlich um einen algorithmischen Unterschied handelt.

1) **Stuttgart Hbf(ab 06:30) -> Mannheim Hbf(an 07:31), IC 1114**

Diese Verbindung wird von einer schnelleren Fahrt überholt:

Stuttgart Hbf(ab 06:51) -> Mannheim Hbf(an 07:29), ICE 694

In PRIMA wird die Verbindung $con_{S-MA(1114)}$ trotz Überholung behalten, weil sie dank geringerem Suchwiderstand nicht durch $con_{S-MA(694)}$ dominiert wird.

- Suchwiderstand ($con_{S-MA(1114)}$) = 153,8
- Suchwiderstand ($con_{S-MA(694)}$) = 158,9

In HAFAS müsste die Verbindung $con_{S-MA(1114)}$ nach Regeln auch bei der Preiswertsuche gefunden werden. Da sie aber vermutlich bei der Nachauswahl ein zu schlechtes Reisezeitverhältnis aufweist, ist die Verbindung im normalen Suchergebnis nicht enthalten. Eine sichere Begründung kann leider nicht gegeben werden, weil das Nachauswahlkriterium nicht veröffentlicht wurde und uns nicht vorliegt.

2) **Bremen Hbf(ab 06:09) -> [Umstieg Hannover Hbf(an 07:14)], ICE 1743** **[Weiterfahrt Hannover Hbf(ab 07:21)] -> Berlin Hbf(an 09:19), IC 2241**

Diese Verbindung wird von einer schnelleren Fahrt überholt:

Bremen Hbf(ab 06:09) -> [Umstieg Hannover Hbf(an 07:14)], ICE 1743

[Weiterfahrt Hannover Hbf(ab 07:21)] -> **Berlin Hbf(an 09:11)**, ICE 541

In PRIMA wird die Verbindung $con_{HB-[H]-B(1743,2241)}$ trotz Überholung behalten, weil sie dank geringerem Suchwiderstand nicht durch $con_{HB-[H]-B(1743,541)}$ dominiert wird.

- Suchwiderstand ($con_{HB-[H]-B(1743,2241)}$) = 493,0
- Suchwiderstand ($con_{HB-[H]-B(1743,541)}$) = 537,2

In HAFAS kann die Verbindung $con_{HB-[H]-B(1743,2241)}$ bei der Preiswertsuche nicht gefunden werden, da es sich um eine Mischung aus Produktklassen ICE und IC/EC handelt und deshalb aus Sicht von HAFAS genauso teuer ist wie eine reine ICE-Verbindung.

3) **Berlin Hbf(ab 16:37)** -> [Umstieg Hannover Hbf(an 18:37)], IC 140
[Weiterfahrt Hannover Hbf(ab 18:45)] -> **Bremen Hbf(an 19:51)**, ICE 1744

Diese Verbindung wird in PRIMA gefunden aber nicht in HAFAS. Dies kann genauso wie in 2) begründet werden, da sie von folgender Verbindung überholt wird:

Berlin Hbf(ab 16:49) -> [Umstieg Hannover Hbf(an 18:28)], ICE 544
[Weiterfahrt Hannover Hbf(ab 18:45)] -> **Bremen Hbf(an 19:51)**, ICE 1744

4) **Siegburg/Bonn(ab 08:11)** -> [Umstieg F-Flughafen(an 08:51)], ICE 513
[Weiterfahrt F-Flughafen(an 09:13)] -> **Frankfurt(M) Hbf(an 09:24)**, ICE 11

Diese Verbindung wird von einer schnelleren Fahrt überholt:

Siegburg/Bonn(ab 08:11) -> [Umstieg F-Flughafen(an 08:51)], ICE 513
[Weiterfahrt F-Flughafen(an 09:01)] -> **Frankfurt(M) Hbf(an 09:12)**, ICE 1597

Grund weshalb in PRIMA Verbindung $con_{SU-[F-Flug]-F(513,11)}$ nicht durch Verbindung $con_{SU-[F-Flug]-F(513,1597)}$ dominiert wird, bedarf weiterer Untersuchung.

2.4.3 Sonstige unterschiedliche Verbindungen

Weitere typische unterschiedliche Verbindungen, die auf einen algorithmischen Bezug zurückführen aber außerhalb des Kernvergleiches aufgefallen sind, werden in diesem Abschnitt dargestellt.

Lüneburg – Berlin

Für die Relation Lüneburg – Berlin bzw. umgekehrt werden in PRIMA deutlich weniger Verbindungen gefunden. Verbindungen, die HAFAS zusätzlich findet, schließen überwiegend eine ICE-Fahrt zwischen Hamburg/Hannover und Berlin ein. Diese Verbindungen werden zwar nicht überholt, bilden in PRIMA aber aufgrund des großen Umwegs und der höheren Produktklasse einen deutlich größeren Fahrpreis nach, so dass ihnen ein sehr hoher Suchwiderstand zugerechnet wird. Täglich verbindet eine direkte EC-Fahrt beide Städte mit einer sehr geringen Fahrzeit. Der Suchwiderstand der langsameren ICE-Umwegverbindungen mit Umstieg ist infolge dessen so groß, dass bei der Vorauswahl folgende Grenze überschritten wird:

$$\text{Suchwiderstand(ICE-Verbindung)} > 1,5 * [\text{minimaler Suchwiderstand}] + 10$$

Beispielsweise wird folgende nicht überholte Verbindung in HAFAS gefunden, jedoch nicht PRIMA:

Berlin Hbf(ab 10:17) -> [Umstieg Hamburg Hbf(an 11:57)], ICE 1596
[Weiterfahrt Hamburg Hbf(ab 12:28)] -> **Lüneburg**(an 12:57), IC 2373

Erhöht man aber den Verwurfgrenzfaktor bei der Vorauswahl und Verbindungssuche von 1,5 auf 1,75, kann $con_{B-[HH]-LG(1596,2373)}$ auch in PRIMA gefunden werden.

Paris – München

Für die Relation Paris und München bzw. umgekehrt werden pro Richtung 2 zusätzliche überholte Verbindungen mit großer Nachfrage in PRIMA gefunden.

HAFAS findet:

Paris Est(ab 07:25) -> [Umstieg Stuttgart Hbf(an 11:04)], TGV 9571
[Weiterfahrt Stuttgart Hbf(ab 11:12)] -> **München Hbf**(an 13:27), ICE 1091

PRIMA findet zusätzlich:

Paris Est(ab 07:06) -> [Umstieg Mannheim Hbf(an 10:17)], ICE 9551
[Weiterfahrt Mannheim Hbf(ab 10:30)] -> **München Hbf**(an 13:27), ICE 1091

Verbindung *con* *Paris-[MAJ-M(9551,1091)* wird zwar überholt, weist aber in PRIMA aufgrund der abweichend fahrenden Strecke weniger Tarifpunkte und somit ausgeglichen auch einen geringeren Suchwiderstand auf.

2.5 Zusammenfassung und Optimierungsvorschläge

Nachdem alle Unterschiede der wichtigsten Verbindungen untersucht worden sind, wird in diesem Kapitel eine Schlussfolgerung gezogen. Aus diesen werden potenziell ergreifbare Maßnahmen abgeleitet.

2.5.1 Ergebnisse

Aus verschiedenen Gründen suchen HAFAS und PRIMA mit unterschiedlichen Prioritäten auf ganz andere Weisen nach „sinnvollen“ Verbindungen. Es gibt dennoch keine eindeutige Antwort darauf, welches System „besser“ oder „richtig“ sucht, denn HAFAS und PRIMA werden anders aber zweckgemäß eingesetzt.

HAFAS soll, unabhängig von der tatsächlichen Nachfrage, möglichst alle aus der Perspektive des Kunden attraktiven Verbindungen effektiv finden.

PRIMA dagegen plant für die Zukunft. Eine Verbindungssuche wird für PRIMA benötigt, um die Reiestromverteilung auf dem gesamten wesentlichen Bahnangebot zu prognostizieren. Dabei kann es aber nicht jede (insbesondere kurzfristig aufzutretende) Feinheit des nur begrenzt bekannten zukünftigen Fahrplans berücksichtigen. Wenn sich jedoch eine erkennbare Regel ermitteln lässt, kann eine fahrplanbasierte Besonderheit im Voraus berücksichtigt werden (Beispiel: Ausschluss des Mannheim-Umstiegs).

Die Mehrheit der wichtigsten und sinnvollsten Verbindungen kann jedoch mit beiden Systemen gefunden werden. Während beide Systeme bei mehreren durchgeführten Vergleichen teilweise unterschiedliche Ergebnisse erzielen, ist ihre algorithmische Auswirkung in den überwiegenden Fällen nicht besonders groß oder entscheidend.

Ein Großteil der algorithmischen Unterschiede weist auf die Markierung in HAFAS hin. Sie hilft zwar bei der Suche zusätzliche überholte Verbindungen zu finden, hat aber gleichzeitig auch den Nebeneffekt, dass nicht besonders sinnvolle Verbindungen stets mit angezeigt werden. Diese sind aufgrund der geringen Nachfrage bei der PRIMA-Verbindungssuche genauso wenig relevant wie eine Verbindung mit absichtlich gebuchtem Umweg oder zusätzlich verlängertem Aufenthalt.

2.5.2 Einzelbewertungen

Fahrplanstand und Messesonderfahrten

Vom Vergleich von EMS->PRIMA lässt sich klar ableiten, dass die meisten sich unterscheidenden Verbindungen durch abweichende Fahrpläne verursacht werden.

Bei diesen Fahrplanunterschieden können zwei Varianten auftreten: **vergleichsmethodenabhängige Fahrplanunterschiede** und **datenquellenabhängige Fahrplanunterschiede**.

Die vergleichsmethodenabhängigen Fahrplanunterschiede wurden bereits beim Vergleich PRIMA->EMS minimiert. Dies bedeutet nicht, dass es sich in der dieser Vergleichsphase um eine bessere Vergleichsmethode handelt, da jeder der beiden Vergleiche unterschiedlich ausgerichtet ist. Eine Verwendung der Vergleichsmethode von PRIMA->EMS beim Vergleich von EMS->PRIMA kann dazu führen, dass an einem Tag zufällig angehäuften Buchungen einer Verbindung eine irrtümlich überhöhte Wichtigkeit bewirken.

Die datenquellenabhängigen Fahrplanunterschiede können dagegen ohne Nebenwirkungen minimiert werden, indem ein mit dem Jahr 2012 übereinstimmender Fahrplan der Nahverkehrs- und Nachtzüge in PRIMA importiert wird.

Messefahrten sind zugegebenermaßen spezielle Verbindungen an ganz bestimmten Tagen. Die wichtigen davon haben aber meist eine so große Auswirkung im ganzen Netz, dass sie fast gegenüber allen anderen Unterschieden überwiegen. Deshalb sollten sie auch in PRIMA eingeschlossen und auf eine spezielle Weise berücksichtigt werden.

Für eine bessere Qualität der Aussagen, sollten absolut identische Fahrplanstände vorliegen. Vor einem zuverlässigen Vergleich zwischen den beiden Systemen sollte dies in PRIMA überprüft werden.

Anbindung

Des Weiteren kann den Vergleichen entnommen werden, dass die verschiedenen Anbindungsmöglichkeiten in Städten mit mehr als einem Fernverkehrsbahnhof auch häufig zu großen Unterschieden führen. PRIMA nutzt hierzu in den meisten Fällen nur eine Möglichkeit, wohingegen in EMS

die Alternativen (Ein- oder Ausstieg an anderen Fernverkehrsbahnhöfen) in einigen Großstädten auch eine relativ große Nachfrage aufweisen können.

Dieses Problem kann in PRIMA potenziell gelöst werden, indem die Bezirke in den Städten Berlin, Hamburg und München nach Fernverkehrsbahnhöfen in kleinere Blöcke verfeinert werden. Dies ist bei den Flughafenbezirken bereits implementiert.

Markierungen

Die Markierung ist, wie bereits beschrieben, eine wichtige Ergänzung der Standardsuche in HAFAS und hauptsächlich dafür verantwortlich, dass Verbindungen – trotz Übereinstimmung von Fahrplan, Einstiegs-/ Ausstiegsbahnhof (Anbindung) und Umsteigemöglichkeit – in dem anderen System nicht gefunden werden.

Allgemein gesehen können der Markierung als Grund Zugbesetzungssteuerung („01“-Markierung), Preisvorteil („00“-Markierung) und Buchbarkeit (Sonderzug-Markierung) zugeschrieben werden.

Die „00“-Markierung in HAFAS ist ein preisrelevantes Thema. Sie ermöglicht die Suche nach langsameren aber potenziell viel günstigeren Verbindungen ggf. auch mit Umstieg, die sonst – wie in PRIMA – nur durch eine näherungsweise Preisbildung erfasst werden können. Alternativen zum ICE-Sprinter und internationalem IC-Bus (Stand 2014) können dabei effektiv gesucht werden. Außerdem ist die „00“-Markierung eine erfolgreiche Suchmethode auf der Schnellfahrstrecke KRM, wo langsamere Rheinfahrten mit angezeigt werden. Da PRIMA einige ICE-(Direkt-)Verbindungen über die Rheinstrecke nicht findet, könnte man den Markierungsmechanismus in PRIMA einführen und eingeschränkt auf diesen Abschnitt ähnlich nachbilden. Die Markierung zwischen Frankfurt und Göttingen dagegen ist teilweise überflüssig, weil hier neben der Anzeige der gewünschten IC-Linie auch eine überholte ICE-Umsteigeverbindung gefunden wird, welche nicht von großer Bedeutung ist. Darüber hinaus ändert sich der Markierungsabschnitt auf dieser Strecke von Jahr zu Jahr, weshalb eine Voraussage nur schwer möglich ist. In solchen Situationen wäre es logischer und potenziell besser geeignet, den aktuellen Sparpreis in die Verbindungssuche mit einfließen zu lassen.

Die „01“-Markierung greift auf eine auslastungsgesteuerte Weise zu. Sie ermöglicht die Suche nach Alternativen einiger ausgewählter Verbindungen (in Zugfahrtstreckenabschnitten) mit voraussichtlich starker Nachfrage. Diesen Bereich kann PRIMA im Moment nicht berühren. Die „01“-Markierung ist auch

für PRIMA sinnvoll, da das neue *Spill & Recapture* Verfahren von PRIMA auch den Besetzungsgrad der Züge berücksichtigt. Die auslastungsstarken Verbindungen werden nach einer Umlegung neu bewertet und die davon abgezogene Nachfrage erneut auf vorhandene Verbindungen verteilt. Parallel fahrende, ursprünglich dominierte Verbindungen könnten mithilfe einer solchen Markierung auch einen Teil dieser Nachfrage einfangen, wenn eine neue Verbindungssuche stattfinden und diese dominierten Verbindungen aufnehmen würde.

Die Sonderzug-Markierung bezieht sich auf zwei Bereiche: bei Nichtbuchbarkeit eines DB-Fremdproduktes der gefundenen Verbindung wird eine Ausweichverbindung vorgeschlagen, bei einigen Überholungen auf Strecken, auf welchen beide Verbindungen markiert sind, wird die überholte – soweit sinnvoll – auch behalten. Dieser Effekt wirkt sich unter den bedeutsamsten Verbindungen nur auf ein einziges Verbindungspaar aus, bei welchem ausschließlich der zu erwartende Preisvorteil (höheres Sparpreiskontingent) der langsameren Verbindung für die Ausweichung der Buchung spricht. Da PRIMA für die Zukunft plant und möglichst allgemeine Zugfahrtenprofile anstatt speziellen Zugverhaltens nachbilden soll, müssen in PRIMA Markierungen auf gesonderten Zugnummern prinzipiell ausgeschlossen werden.

Preiswertsuche

In HAFAS können Verbindungen teurer Produktklassen schrittweise ausgeschaltet werden, wenn das Häkchen „schnelle Verbindungen bevorzugen“ nicht angewählt ist. PRIMA kann aus den Tarifpunkten bereits Verbindungen nach zu erwartender Preishöhe differenzieren und findet neben reinen IC-Verbindungen, die aus Sicht von HAFAS angeblich billiger seien, auch ggf. ebenso preiswerte gemischte IC/ICE-Verbindungen, die HAFAS systematisch übersieht. So gesehen kommt eine zusätzliche Preiswertsuche für PRIMA nicht in Frage.

Hinsichtlich des bedeutendsten Unterschieds in dieser Kategorie, des direkten IC/EC auf der Relation Frankfurt und München, müssen ggf. die Dominanzkriterien geringfügig angepasst werden, um die überholten Verbindungen in PRIMA zu finden.

Auswahlverfahren und Dominanzkriterium

HAFAS verwirft bei der Suche keine Verbindungen, die in der zeitlichen Ausgangslage nicht durch eine andere Verbindung überholt wird. PRIMA setzt im Auswahlverfahren eine an der besten Verbindung orientierte Suchwiderstandsgrenze, die jede gefundene und ggf. auch eine nicht überholte (d.h. eine nächste schnellste) Verbindung einhalten muss.

Berlin – Lüneburg ist ein Beispiel, bei dem die meisten nicht überholten ICE-Verbindungen die Auswahlkriterien von PRIMA nicht erfüllen und darum gelöscht werden.

Eine einfache Lösung wäre, in VISUM die grundsätzliche Option einzuführen, nicht überholte Verbindungen in keinem Fall zu dominieren. Alternativ kann die Auswahl in PRIMA auch so eingestellt werden, dass VISUM die Suchwiderstandsgrenze auf Basis eines Durchschnittswertes der besten 5-10 Verbindungen zu berechnen anstatt wie bisher nur die Verbindung mit dem geringsten Suchwiderstand widerzuspiegeln. In beiden Fällen können Probleme wie Berlin – Lüneburg behoben werden, da sonst z.B. ein einziger Direktverbindungszug ggf. die Suchschwelle auf der Relation wesentlich erhöhen könnte.

Da das Verändern des PRIMA-Dominanzkriteriums erfahrungsgemäß eine gravierende Auswirkung hat, muss damit vorsichtig umgegangen werden. Bevor dieses als letzter Schritt vorgenommen werden kann, müssen alle anderen möglichen Faktoren bereits optimal eingestellt worden sein. Eine Lösung in dieser Kategorie sollte außerdem nur dann zugelassen werden, wenn nach den ersten Optimierungen noch wichtige ungelöste Probleme vorliegen.

Quellennachweise

- [1] K. L. Cooke and E. Halsey, "The shortest route through a network with time-dependent internodal transit times," *Journal of Mathematical Analysis and Applications*, vol. 14, pp. 493-498, 1966.
- [2] M. Müller-Hannemann, F. Schulz, D. Wagner, and C. Zaroliagis, "Timetable Information: Models and Algorithms," in *Algorithmic Methods for Railway Optimization*. Springer Berlin Heidelberg, 2007, pp. 67-90.
- [3] E. W. Dijkstra, "A note on two problems in connexion with graphs," in *Numerische Mathematik 1*, 1959, pp. 269-271.
- [4] Institut für Informatik, Universität zu Köln. (2009) Kürzeste-Wege-Algorithmen und Datenstrukturen. [Online]. http://www.zaik.uni-koeln.de/AFS/teachings/ss09/ProgPrakt/Skript_Teil1.pdf
- [5] HaCon Ingenieurgesellschaft mbH. (2014, Aug.) Internet — HaCon, Fahrplanauskunft HAFAS, Fahrplankonstruktion TPS. [Online]. <http://www.hacon.de/hafas/internet>
- [6] Deutsche Bahn AG. (2014, Aug.) DB BAHN - Verbindungen - Ihre Auskunft. [Online]. <http://www.bahn.de>
- [7] F. Schulz, "Timetable Information and Shortest Paths," Fakultät für Informatik, Universität Karlsruhe (TH), Dissertation, 2005.
- [8] Verkehrs- und Tarifverbund Stuttgart GmbH. (2014, Aug.) VVS - Elektronische Fahrplanauskunft EFA. [Online]. <http://www.vvs.de>
- [9] PTV GROUP, "Verbindungssuche," in *VISUM 12.5 Grundlagen*, ch. 6.10.2.
- [10] I. H. S. W. Markus Friedrich, "Timetable-based Transit Assignment Using Branch & Bound Techniques," *Transportation Research Records*, no. 1752, pp. 100-107, 2001.
- [11] HaCon Ingenieurgesellschaft mbH. (2010, Sep.) HAFAS Rohdatenformat - Eingabedateien der Datenaufbereitung. Dokumentation.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben.

Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet.

Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens.

Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht.

Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.