

Dipl.-Ing. Clemens Strauß

ÜBER DAS FINDEN KÜRZESTER WEGE
DURCH RAUM UND ZEIT

D I S S E R T A T I O N

zur Erlangung des akademischen Grades
Doktor der technischen Wissenschaften

Doktoratsstudium der technischen Wissenschaften
Doctoral School Geosciences – Class of Geodesy



Technische Universität Graz

Betreuer:

Univ.-Prof. Dr.phil. Norbert Bartelme
Institut für Geoinformation

Graz, Mai 2011

The most exciting phrase to hear in science, the one
that heralds new discoveries, is not "*Eureka!*" but
"*That's funny ...*".

Isaac Asimov (1920–1992)

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

A handwritten signature in black ink, appearing to read 'Hannes Schmid', written in a cursive style.

Graz, am 4. Mai 2011

.....
(Unterschrift)

Kurzfassung

Den kürzesten Weg zu kennen, ist für einen Reisenden von großem Wert. Die Herausforderungen, denen er dabei zu begegnen hat, haben sich im Lauf der Geschichte grundlegend geändert: War es früher die Navigation selbst, wie beispielsweise das intensive Kartenstudium, das dem eingeschlagenen Weg zugrunde lag, so muss das korrekte Bedienen eines Autonavigationssystems als die navigatorische Herausforderung für den Reisenden von heute gesehen werden. Was sich aber im Inneren eines Autonavigationssystems verbirgt, ist nicht minder interessant als die *Tabula Peutingeriana*.

Beginnt man die Algorithmen, die nicht nur im Inneren eines Autonavigationssystems anzutreffen sind, genauer zu betrachten, so finden sich grundsätzlich zwei Herangehensweisen zur Berechnung kürzester Wege: Einerseits wird die Datenbasis so weit an die Fragestellung angepasst, dass ein Standardalgorithmus darauf angewendet werden kann. Andererseits wird bei allgemein formulierten Basisdaten der Funktionsumfang des Algorithmus speziell auf die Fragestellung abgestimmt. In der vorliegenden Arbeit sind beide Herangehensweisen vertreten und werden vor dem Hintergrund einer funktionellen Erweiterung der Suchbaumausbauung und einer Erschließung zusätzlicher Anwendungsfelder für die Routenberechnung eingesetzt.

Nimmt man für die Datenmodellierung ein Knoten-Kanten Modell, eine in der Navigation geläufige Form der Datengrundlage, als Ausgangspunkt, so lassen sich durch Steigerung der Dimensionalität des Basiselementes drei weitere Modelle herleiten: Kanten-Traversen Modell, polygonales Modell und polyhedrales Modell. Diese Modelle unterscheiden sich nicht nur durch die Dimension ihres jeweiligen Basiselementes, sondern auch durch die Beschreibungsform der Nachbarschaften zwischen den einzelnen Basiselementen. Werden diese Nachbarschaften bei Knoten-Kanten und Kanten-Traversen Modellen explizit durch ein Verbindungselement beschrieben, so wird die Berührung der Basiselemente bei den polygonalen und polyhedralen Modellen zur impliziten Nachbarschaftsbeschreibung genutzt. Zwei zusätzliche Formen der Datengrundlage entstehen bei der Mischung der einzelnen Modelle. Hierbei werden gemischt-einfache und gemischt-komplexe Strukturen gebildet, die sich durch die vorherrschenden Nachbarschaftsbeschreibungen der einzelnen Modelle einordnen lassen.

Eine Sonderstellung in der Datenmodellierung nimmt die Beschreibung zeitabhängiger Vorgänge vor dem Hintergrund des öffentlichen Fahrplanverkehrs ein. Diese werden in einem 3-dimensionalen Kanten-Traversen Modell abgebildet und bestehen aus einer räumlichen Komponente in der XY-Ebene und einer zeitlichen Komponente entlang der Z-Achse. Durch die geometrische Form des Basiselementes lassen sich Prozesse, wie beispielsweise das Warten an einer Haltestelle oder das Fahren mit einem Verkehrsmittel, beschreiben. Mit Hilfe des Verbindungselementes werden die einzelnen Prozesse miteinander in Beziehung gebracht, wodurch Vorgänge, wie beispielsweise das Sitzenbleiben im Verkehrsmittel oder das Aussteigen an einer Haltestelle, im Zuge einer Routenberechnung Berücksichtigung finden.

Innerhalb der formulierten Datenmodelle können Routen auf unterschiedlicher Weise berechnet werden. Dabei stellt die Ausbreitung von Suchbäumen innerhalb der Datengrundlage das funktionelle Zentrum der Routenberechnung dar. Ausgehend von einem Algorithmus nach *Dijkstra* sind bidirektionale und hierarchische Suchbaumausbreitungsverfahren ableitbar, die eine Minimierung der Menge an Basisdaten und eine damit verbundene Verkürzung der Berechnungszeit einer Route anstreben. Unter der Bezeichnung *Multi Level Hopping* ist eine Sonderform eines hierarchischen Routing-Algorithmus zu verstehen, der nicht nur bestrebt ist, den Suchbaum in der höchsten Hierarchieebene auszubreiten, sondern auch bei

entsprechenden Umständen eine Ausbreitung in niedrigeren Hierarchieebenen zulässt.

In drei Anwendungsentwürfen werden auf konzeptioneller Ebene verschiedene Datenmodelle und Suchbaumausbreitungsmethoden zusammengeführt. Im ersten Konzept über multimodale Routenplanung werden Daten eines Wegenetzes für Fußgänger (Knoten-Kanten Modell) mit dem zeitabhängigen Liniennetz des öffentlichen Verkehrs (Kanten-Traversen Modell) kombiniert. Dieses Konzept dient als Beispiel für eine gemischt-einfache Struktur, da in der gemischten Datenbasis ausschließlich explizite Nachbarschaftsbeschreibungen vorkommen. Der zweite Anwendungsentwurf, *Short-Cut Routing*, behandelt die Routenberechnung in einer gemischt-komplexen Struktur. Hier werden Basisdaten mit explizit und implizit beschriebenen Nachbarschaften eingesetzt. Dieses Konzept ist thematisch der Fußgänger-navigation zuzuordnen und basiert auf einem Wegenetz für Fußgänger (Kanten-Traversen Modell) und auf freiem Gelände (polygonales Modell). Durch die verwendete Datengrundlage wird der Bewegungsfreiheit eines Fußgängers Rechnung getragen. Das dritte und letzte Konzept beschreibt unter dem Namen *Cube Routing* die kürzeste Wegesuche durch eine 3-dimensionale Datengrundlage (polyhedrales Modell). Dieses Verfahren der Routenberechnung ist funktionell im Bereich der traditionellen Navigation verwurzelt, eröffnet jedoch Anwendungsfelder, die sich davon fernab befinden.

Eine Weiterentwicklung, aufbauend auf den gewonnenen Erkenntnissen, kann auf konzeptioneller, technischer und anwendungsbezogener Ebene erfolgen. Dabei sind konzeptionelle und technische Erweiterungen mit Weiterentwicklungen im Bereich der Geodateninfrastruktur gekoppelt und von der Definition geodatenspezifischer Standards abhängig. Eine anwendungsbezogene Weiterentwicklung wird vor allem von der Kreativität, neue Anwendungsfelder zu identifizieren, getragen, wobei das *Cube Routing* das größte Potential dazu vorweisen kann.

Abstract

Every traveller is looking for a perfect route. Depending on the traveller's preferences the shortest, the cheapest or the most beautiful route is chosen as the perfect route. Over the years this core task of navigation – the process of planning a route – has changed: It has moved from the navigation itself, like reading a map, to the handling of the in-car navigation system. Nevertheless it is still challenging. Hence, in the past the quality of the decided route was connected to the experiences of a traveller. Nowadays optimal or heuristic routing algorithms exculpate the traveller from deciding which way to go.

Generally two main strategies for these algorithms exist to face the shortest path problem: The first strategy focuses on an intense data base modelling process for an adequate reflection of the problem. Afterwards standard routing algorithms can be applied to this data base without changes. The second strategy is based on a rather broadly described data base, but the algorithm is specially adapted to the given routing problem. Within this thesis both strategies are applied to extend the functionality of search tree spreading methods – the core task of route computation – and to identify additional fields of application for route calculation processes.

The starting point of data modelling is represented by a very common structure in a navigational environment and consists of vertices and edges. Extending the structures by increasing the dimension of base elements, edge-traverse models, polygonal and polyhedral structures arise. Beside a dimensional-based distinction data structures can be classified by the description of neighbourhood relationships between single base elements. Vertex-edge and edge-traverse models are assigned to the class of explicitly entitled neighbourhood relationships – base elements are connected by additional linking elements. In contrast to this, polygonal and polyhedral structures are assigned to the class of implicitly entitled neighbourhood relationships – neighbouring base elements are identified by spatially touching each other. Based on this uniform classification of data base structures two further classes can be defined: mixed structures of low and high complexity which are ranked by the dominating neighbourhood declaration.

Temporal processes which are situated in the context of public transport are holding an exceptional position within the whole data modelling procedure. Therefore a 3-dimensional edge-traverse structure is used to describe these processes and consists of spatial information within the XY-plane and temporal information along the Z-axis. In detail edges are used to describe elementary temporal processes, like waiting at the station or going by bus from one station to another. Traverses connect these processes and sequences, like continuing a trip or entering the bus at the station, can be defined.

Within these defined data structures routes can be calculated in different ways. Coming from a well known algorithm by *Dijkstra* bidirectional and hierarchical strategies improve the search tree spreading. The aim of these improvements can mainly be identified in limiting the quantity of elements within the data base for a route computation which is linked to a shortening of its calculation period. *Multi Level Hopping* is the name of a specialised hierarchical routing algorithm which does not necessarily aim to reach the highest hierarchical level during a route computation. In certain circumstances the search tree is spread to lower hierarchical levels to guarantee a connection between start and destination.

The combination of different data structures and routing algorithms leads to three conceptual use-cases. The first concept focuses on a multimodal transportation scenario. Therefore a path network for pedestrians (vertex-edge model) and a temporal network of public

transport (edge-traverse model) are jointly used. This use-case exemplifies a route computation within a mixed data structure of low complexity. Within the second concept, *Short-Cut Routing*, a mixed data structure of high complexity is used to satisfy a pedestrian navigation application. This structure consists of a path network for pedestrians (edge-traverse model) and open space elements (polygonal model) which considers the mobility of a pedestrian best. The third and last concept is called *Cube Routing* reasoned by the cubic shape of the used base elements (polyhedral model). This data base permits a 3-dimensional and voluminous description of spatial phenomena. Although from a methodical point of view calculating a route through this data base relates to the discipline of navigation a wide range of applications can be served which is sited far-off.

The range of enhancement divides into conceptual, technical and application-based fields. Especially conceptual and technical enhancements are strongly linked to the design and development of geo-data infrastructures respectively to the definition of new geo-specific standards. According to an application-based enhancement, creativity sustains the identification of new fields of application at which *Cube Routing* seems to have the highest potential.

Danksagung

Diese Dissertation mag zwar nur meinen Namen als Verfasser anführen, jedoch ist das Entstehen einer solchen Arbeit von vielen Menschen beeinflusst! *Norbert Bartelme*, meinem Doktorvater, ist es zu verdanken, dass ich diese Arbeit überhaupt in Angriff nehmen konnte. Es ist nicht selbstverständlich als Dissertant ein Thema der persönlichen Wahl bearbeiten zu dürfen. Auch für das äußerst angenehme Arbeitsumfeld und für die immer herrschende konstruktive Atmosphäre möchte ich mich sehr herzlich bei ihm bedanken!

Da sich diese Dissertation in der Schnittmenge von Geoinformatik und Navigation positionieren lässt, war *Manfred Wieser* ein wichtiger Ansprechpartner für navigatorische Anliegen. Hiermit möchte ich mich für die gemeinsamen und intensiv geführten Diskussionen bedanken, die so manche Lösung eines meiner Probleme nach sich zogen.

Nicht selten kam es vor, dass durch kurze, vermeintlich belanglose Gespräche am Gang, bei Tee oder Kaffee doch wesentliche Ideen geboren wurden. Diese Ideen fanden nicht nur konzeptionell in die Arbeit Eingang sondern boten auch konkrete Lösungen auf technischer Ebene. Deshalb mein ganz besonderer Dank an *Gernot Hollinger*, *Rainer Prüller*, *Konrad Rautz* und *Johannes Scholz*!

Weiters möchte ich mich bei den *Veranstaltern* und bei den *Mitstudierenden* der Doktoral School bedanken. Die Möglichkeit, wesentliche Abschnitte der Forschung in einer angenehmen Atmosphäre präsentieren und untereinander konstruktiv diskutieren zu können, wirkte sich stets motivierend auf die Fortführung der Arbeit aus.

Wer auch hier nicht unerwähnt bleiben soll, sind *Viktoria Galler* und *Elke Riedlberger*. Fachlicher Diskurs ist für eine Dissertation unerlässlich, aber auch ein zeitweises bewusstes Distanzieren von der Arbeit ist notwendig. Für all die Momente des kurzzeitigen Loskommens von lösenzumüssenden Problemstellungen und des Krafttankens möchte ich mich sehr herzlich bei beiden bedanken!

Abschließend möchte ich mich bei meiner *Familie* bedanken. Nicht nur dass sie mir immer sicheren Rückhalt bietet und mich in meinem bisherigen Ausbildungsweg stets unterstützt, sondern auch dass sie meine Dissertation gelesen hat und dabei von so manch wirrer Formulierung befreien konnte – Danke!

Inhaltsverzeichnis

1	Einführung	1
1.1	Ziel der Arbeit	1
1.2	Motivation	1
1.3	Innovationen	2
1.4	Organisation der Arbeit	3
2	Konzepte der Routenberechnung	5
2.1	Ein Überblick	5
2.1.1	Optimale kürzeste Wege	6
2.1.2	Heuristische kürzeste Wege	6
2.2	Algorithmus nach Dijkstra	8
2.2.1	Das Basiselement	8
2.2.2	Das Verbindungselement	9
2.2.3	Ausbreitung des Suchbaumes	10
2.2.4	Aufbereitung des Ergebnisses	13
3	Datenmodellierung	15
3.1	Knoten-Kanten Modelle	15
3.2	Kanten-Traversen Modelle	17
3.3	Gemischt-einfache Strukturen	19
3.4	Polygonale Modelle	20
3.5	Polyhedrale Modelle	24
3.6	Gemischt-komplexe Strukturen	29
4	Beschreibung zeitabhängiger Vorgänge	31
4.1	Beziehung zwischen zeitabhängigen Vorgängen	32
4.1.1	Definitionen	32
4.1.2	Beziehungen zwischen zwei Zeitpunkten	33
4.1.3	Beziehung zwischen Zeitpunkt und Zeitspanne	33
4.1.4	Beziehung zwischen zwei Zeitspannen	34
4.2	Abbildung zeitabhängiger Vorgänge	37
4.2.1	Thema/Zeit Diagramme	37
4.2.2	Raum/Zeit Diagramme	38
4.3	Modellierung zeitabhängiger Vorgänge	39
4.3.1	Einfache Haltestellen	41
4.3.2	Komplexe Haltestellen	45

4.3.3	Wechsel zwischen Individualverkehr und Fahrplanverkehr	47
5	Methoden der Suchbaumausbreitung	51
5.1	Monodirektionale Suchbaumausbreitung	51
5.2	Bidirektionale Suchbaumausbreitung	54
5.3	Hierarchische Suchbaumausbreitung	57
5.4	Multi Level Hopping	64
6	Anwendungskonzepte	71
6.1	Multimodale Routenplanung	72
6.1.1	Datengrundlage	72
6.1.2	Anpassung des Algorithmus an die gegebene Problemstellung	74
6.1.3	Ergebnisse	75
6.2	Short-Cut Routing	79
6.2.1	Datengrundlage	79
6.2.2	Anpassung des Algorithmus an die gegebene Problemstellung	80
6.2.3	Ergebnisse	82
6.3	Cube Routing	85
6.3.1	Datengrundlage	85
6.3.2	Anpassung des Algorithmus an die gegebene Problemstellung	86
6.3.3	Ergebnisse	88
7	Zusammenfassung, Diskussion und Ausblick	93
7.1	Zusammenfassung	93
7.1.1	Definition gemischter Datenstrukturen	93
7.1.2	Modellierung zeitabhängiger Prozesse	94
7.1.3	Multi Level Hopping	95
7.1.4	Anwendungskonzepte für gemischte Datenstrukturen	95
7.1.5	Routenberechnung im Raum	96
7.2	Diskussion	96
7.2.1	Einsatz gemischt-komplexer Datenstrukturen	97
7.2.2	Multi Level Hopping	97
7.2.3	Einsatz 3-dimensionaler Datenbasen zur Routenberechnung	98
7.3	Ausblick	98
7.3.1	Konzeptionelle Weiterentwicklung	98
7.3.2	Technische Weiterentwicklung	99
7.3.3	Anwendungsbezogene Weiterentwicklung	99

Abkürzungen und Akronyme

BHFFA Bidirectional Heuristic Front-to-Front Algorithm

BHHS Bidirectional Heuristic-based Hierarchical Search

BHPA Bidirectional Heuristic Path Algorithm

BIDA* Bidirectional Iterative-Deepening A*

CAD Computer Aided Design

EPSG European Petrol Survey Group

EWKB Extended Well Known Binary

EWKT Extended Well Known Text

FRC Functional Road Class

FS Fremdschlüssel

GES Gemischt-Einfache Struktur

GIS Geographisches Informationssystem

GPSE Global Positioning System Standard Epoch

GPST Global Positioning System Time

HHPS Heuristic Hierarchical Path Search

HSR Hierarchical Spatial Reasoning

HWA Hierarchical Wayfinding Algorithm

IAT International Atomic Time

IC Intercity

IDPS* Iterative-Deepening Perimeter Search

ITS Intelligent Transportation System

JD Julianisches Datum

LEA Level Exit Arc

LGS Level Graph Search

MJD Modifiziertes Julianisches Datum

NHWA Non-Hierarchical Wayfinding Algorithm

MLH Multi Level Hopping

OeBB Österreichische Bundesbahn

OGC Open Geospatial Consortium
OGP International Association of Oil and Gas Producers
PS Primärschlüssel
RTE Road Transition Element
SP Shortest Path
spatialSQL spatial Structured Query Language
SPP Shortest Path Problem
SRID Spatial Reference Identifier
SSMD Single Source Multiple Destination
SSMDP Single Source Multiple Destination Problem
SSSD Single Source Single Destination
SSSDP Single Source Single Destination Problem
TMC Traffic Message Channel
UT Universal Time
UTC Universal Time Coordinated
UTM Universal Transverse Mercator
WGS84 World Geodetic System 1984
WKB Well Known Binary
WKT Well Known Text
X3D extensible 3D

Kapitel 1

Einführung

„*Der Weg ist das Ziel*“ (Konfuzius). Erfasst man dieses Zitat wortwörtlich, so trifft es den Kern der vorliegenden Arbeit sehr genau. Konkret betrachtet, bereitet der Titel dieser Arbeit – „*Über das Finden kürzester Wege durch Raum und Zeit*“ – den Leser darauf vor, sich allgemein mit der Thematik der Routenberechnung zu konfrontieren. Die Einschränkung, „[...] *durch Raum und Zeit*“, spezifiziert den Kontext, vor dem diese Routenberechnung stattfindet. Die Arbeit ist zwei Disziplinen zuzuordnen und bewegt sich dabei in der Schnittmenge von *Navigation* und *Geoinformatik*. Die erste wird zur allgemeinen Behandlung der Routenberechnung, die zweite zur Modellierung von Raum und Zeit eingesetzt.

1.1 Ziel der Arbeit

Die *funktionelle Erweiterung der Suchbaumausbreitung* und die *Erschließung zusätzlicher Anwendungsfelder für die Routenberechnung* sind als Ziele dieser Arbeit zu benennen. Dabei bezieht sich die funktionelle Erweiterung nicht nur auf die Bewältigung 2- und 3-dimensionaler Basisdaten für die Suche eines kürzesten Weges, sondern auch auf eine flexiblere Gestaltung der Suchbaumausbreitung vor dem Hintergrund einer hierarchisch strukturierten Datenbasis.

Die Erschließung zusätzlicher Anwendungsfelder umfasst verschiedene geometrische Modellierungsverfahren, mit denen Basisdaten unterschiedlichster Dimensionen beschreibbar werden, sowie die Kombination dieser Basisdaten zur Berechnung einer Route aus einem Guss. Die Begegnung dieser formulierten Ziele erfolgt auf konzeptioneller Basis. Eine konkrete Ausarbeitung von Anwendungsfällen wird jedoch innerhalb dieser Arbeit *nicht* angestrebt.

1.2 Motivation

Die Motivation zur Verfassung dieser Arbeit begründet sich im persönlichen Interesse des Autors für die Geoinformatik und die Navigation. Das Thema der Routenplanung bietet dafür eine dankbare Plattform, sich mit diesen beiden Disziplinen vertiefend auseinanderzusetzen zu können. Dem ergänzend ist anzumerken, dass weder zu Beginn noch im Verlauf der Forschungstätigkeit die Notwendigkeit bestand, einen Auftrag mit den entstehenden Ergeb-

nissen aus dieser Arbeit bedienen zu müssen. Somit konnte diesem Thema frei von jeglichen äußeren Zwängen begegnet werden, weshalb ausschließlich der Neugier Rechnung zu tragen war.

1.3 Innovationen

Im Verlauf dieser Arbeit werden aufbauend auf anerkannten Algorithmen und etablierten Modellierungsverfahren fünf innovative Erweiterungen vorgestellt. Durch den Aufbau der Arbeit bedingt (vgl. Abschnitt 1.4), werden diese Erweiterungen meist nicht gesammelt beschrieben, sondern sind thematisch den zugehörigen Kapiteln mit entsprechenden Grundlagenabschnitten angegliedert. Im Detail beziehen sich diese fünf Erweiterungen auf folgende Themen:

1. Definition gemischter Datenstrukturen, Abschnitte 3.3 und 3.6: Für viele Fragestellungen der Routenberechnung bilden einheitlich strukturierte Basisdaten eine hinreichende Grundlage. Ein Knoten-Kanten Modell ist dabei als eine der geläufigsten Strukturen zu sehen. Beginnt man nun Basisdaten unterschiedlicher geometrischer Form miteinander zu kombinieren, so eröffnen sich zusätzliche Anwendungsfelder für die Routenberechnung. Bei einer derartigen Mischung von Datenstrukturen ist besonders auf die Beschreibung der nachbarschaftlichen Beziehungen zwischen den einzelnen Basiselementen zu achten. Hierbei wird zwischen expliziten und impliziten Beschreibungen unterschieden, die das Ausbreitungsverfahren des Suchbaumes während der Routenberechnung beeinflussen.
2. Modellierung zeitabhängiger Prozesse, Abschnitt 4.3: Im Kontext des öffentlichen Fahrplanverkehrs wird ein Kanten-Traversen Modell zur Beschreibung der für den Fahrgast relevanten Prozesse entwickelt. Das Datenmodell ist dabei einem Bildfahrplan nachempfunden und besitzt eine 3-dimensionale Ausprägung: Zwei Dimensionen stehen dabei der räumlichen und eine Dimension der zeitlichen Beschreibung von Prozessen zur Verfügung. Durch den Einsatz eines Kanten-Traversen Modelles können Vorgänge durch Kanten (z.B.: Warten an einer Haltestelle oder Fahren mit einem Verkehrsmittel) und die Verknüpfung dieser Vorgänge durch Traversen (z.B.: Sitzenbleiben im Verkehrsmittel oder Aussteigen an einer Haltestelle) definiert werden.
3. Multi Level Hopping, Abschnitt 5.4: Unter dieser Bezeichnung ist eine funktionelle Erweiterung eines hierarchischen Routing-Algorithmus zu verstehen. Dieser Algorithmus ist so ausgelegt, dass sich der Suchbaum nicht ausschließlich entlang der höchsten Hierarchieebene ausbreitet, sondern bei entsprechenden Umständen einen Abstieg in eine niedrigere Hierarchieebene bewusst ausführt. Die Voraussetzung für eine Anwendung dieses Algorithmus liegt dabei in der Struktur der Datenbasis, die einem Kanten-Traversen Modell zu entsprechen hat.
4. Anwendungskonzepte für gemischte Datenstrukturen, Abschnitte 6.1 und 6.2: Aufbauend auf die Definition gemischter Datenstrukturen werden zwei konzeptionelle Anwendungsentwürfe vorgestellt. Im ersten Anwendungskonzept wird eine multimodale Fragestellung behandelt, in der ein Wegenetz für Fußgänger (Knoten-Kanten Modell) und ein zeitabhängiges Streckennetz des öffentlichen Verkehrs (Kanten-Traversen Modell)

die gemischte Datenbasis bilden. In diesem Anwendungsfall beruhen beide Modelle auf einer expliziten Nachbarschaftsbeschreibung.

Im Gegensatz dazu werden im zweiten Anwendungskonzept unter der Bezeichnung *Short-Cut Routing* Modelle mit expliziter und impliziter Nachbarschaftsbeschreibung gemeinsam genutzt. Ziel dieses Anwendungskonzeptes ist die Berechnung einer Route, die auf die Bewegungsfreiheit eines Fußgängers Rücksicht nimmt. Hierzu wird ein Wegenetz für Fußgänger (Kanten-Traversen Modell) mit einer flächenhaften Beschreibung des freien Geländes (polygonales Modell) kombiniert eingesetzt.

5. Routenberechnung im Raum, Abschnitt 6.3: Ein zusätzliches Anwendungskonzept beschreibt die Suche nach kürzesten Wegen in einer Menge 3-dimensionaler Basisdaten. Wegen der würfelartigen Form der einzelnen Basiselemente wird diese Routenberechnung in weiterer Folge auch als *³routing* (sprich: *Cube Routing*) bezeichnet. Trotz einer datenmodellbedingten impliziten Nachbarschaftsbeschreibung zwischen den einzelnen Würfeln, können dennoch Nachbarschaften in vergleichbarer Weise zu expliziten Nachbarschaftsbeschreibungen (z.B.: gerichtete Kante) orientiert werden. Bei einer anwendungsbezogenen Weiterentwicklung dieses konzeptionellen Ansatzes eröffnen sich neue Anwendungsfelder jenseits der traditionellen Navigation.

1.4 Organisation der Arbeit

Die vorliegende Arbeit gliedert sich in drei Teile. Der erste Teil, der Einstieg in die Thematik, bereitet den Leser auf den Hauptteil vor. Dabei wird in *Kapitel 1 – Einführung* eine allgemeine Einführung gegeben und der Kontext der Arbeit umrissen. Daran anschließend wird in *Kapitel 2 – Konzepte der Routenberechnung* ein grober Überblick über verschiedene Strategien der Routenberechnung gegeben. Das Verfahren nach *Dijkstra* wird dabei herausgegriffen und im Detail behandelt. Zusätzlich wird in diesem Kapitel die Nomenklatur festgelegt, so wie sie in den nachfolgenden Kapiteln vorkommt.

Der Hauptteil der Arbeit, der sich über vier Kapitel erstreckt, beginnt mit der Aufbereitung geometrischer Modellierungsverfahren (*Kapitel 3 – Datenmodellierung*). Dabei werden die Modelle in Abhängigkeit der Dimension des jeweiligen Basiselementes von 0-dimensional bis 3-dimensional getrennt von einander beschrieben. Dem ergänzend werden zwei Mischformen von Basiselementen vorgestellt.

In *Kapitel 4 – Beschreibung zeitabhängiger Vorgänge* wird die Abbildung von Zeitinformation in einem Basiselement als ein Spezialfall der Modellierung aufbereitet. Dafür wird ein graphisch orientierter Lösungsansatz basierend auf einem Bildfahrplan des öffentlichen Verkehrs gewählt, der schlussendlich zu einem 3-dimensionalen Kanten-Traversen Modell führt.

Aufbauend auf Kapitel 2 werden in *Kapitel 5 – Methoden der Suchbaumausbreitung* verschiedene Varianten der Suchbaumausbreitung, das funktionelle Kernelement der Routenberechnung, diskutiert. Ausgehend von einer monodirektionalen Methode über ein bidirektionales Verfahren mündet die Suchbaumausbreitung in einem hierarchischen Algorithmus mit situationsangepasster Nutzung der Hierarchieebenen.

Die entwickelten Datenmodelle aus den Kapiteln 3 und 4 sowie die Verfahren zu Suchbaumausbreitung aus Kapitel 5 bilden die Grundlage der in *Kapitel 6 – Anwendungskonzepte* beschriebenen Anwendungsentwürfe. Dabei werden Konzepte beschrieben, die einerseits auf gemischte Datenstrukturen aufbauen (*Multimodale Routenplanung* und *Short-Cut Routing*)

und andererseits eine 3-dimensionale Datengrundlage zur Suchbaumausbreitung nutzen (*Cube Routing*).

Den dritten und letzten Teil dieser Arbeit bildet das *Kapitel 7 – Zusammenfassung, Diskussion und Ausblick*. Hier werden die Erkenntnisse aus der Datenmodellierung, der funktionellen Erweiterung im Bereich der hierarchischen Suchbaumausbreitung und der daran anschließenden Anwendungskonzepte reflektiert und diskutiert. Als Abschluss dieser Arbeit wird ein Ausblick gegeben, der zukünftige Herausforderungen im Bereich der Routenberechnung benennt und der für anschließende Arbeiten als Anknüpfungspunkt dienen kann.

Kapitel 2

Konzepte der Routenberechnung

Das Kapitel *Konzepte der Routenberechnung* beschreibt die funktionelle Grundlage für das Kapitel 5, in dem verschiedene Varianten der Routenberechnung aufgezeigt werden. In all diesen Facetten wird von dem hier beschriebenen Algorithmus ausgegangen und dieser hinsichtlich spezieller Situationen adaptiert oder für weitere Funktionalitäten ergänzt. Im Kapitel 6 wird vor dem Hintergrund diverser Anwendungsszenarien ebenfalls auf die Konzepte der Routenberechnung Bezug genommen.

Einen groben Überblick über Routing-Algorithmen bietet der erste Abschnitt des Kapitels. Hier wird in Anlehnung an [FSR06] eine kategorisierte Kurzbeschreibung von ausgewählten Algorithmen geboten, die einen ersten Einstieg in diese Thematik darstellen. Für die weitere Arbeit wird einer dieser Algorithmen – ein Algorithmus nach *Dijkstra*, [Dij59] – im darauf folgenden Abschnitt detaillierter vorgestellt.

2.1 Ein Überblick

Die Suche nach der kürzesten Verbindung zweier Orte ist keine spezifische Problemstellung, die es seit dem 20. Jahrhundert und der Globalisierung zu lösen gilt. Historische Dokumente, wie die *Tabula Peutingeriana* [Oes10] aus dem 4. bis 5. Jahrhundert oder die *Post Itinera-rien* [Beh06] aus der Zeit um 1560 bis in die Mitte des 17. Jahrhunderts, zeugen von der Notwendigkeit, Fragestellungen bezogen auf kürzeste Verbindungen beantworten zu wollen. Im Fall der *Tabula Peutingeriana* lassen sich sogar Vergleiche mit modernen Netzplänen öffentlicher Verkehrsmittel – wie sie in Bussen, Straßenbahnen oder U-Bahnwagen zu finden sind – anstellen; die Richtigkeit der topologischen Zusammenhänge steht im Vordergrund.

Heutzutage ist das Finden der kürzesten, der schnellsten oder der schönsten Verbindung zweier Orte nicht mehr mit der Kunst des Kartenlesens so eng verwoben wie einst. Das Bedienen von Routenplanungssoftware ist die zeitgemäße Variante an Herausforderung, um diese Aufgabe zu meistern. Systematisierte Datenanalysen, ausgeführt von Algorithmen, versehen im Hintergrund ihren Dienst in Abhängigkeit von eingegebenem Start, Ziel und zusätzlichen Restriktionen, die der Benutzer berücksichtigt wissen will.

Um die kürzeste Verbindung zweier Orte – eine Route – ermitteln zu können, sind die

Nachbarschaftsverhältnisse zwischen den einzelnen Orten zu analysieren. Die Frage, die es im Verlauf eines sogenannten Shortest Path Problem (SPP) zu beantworten gilt, kann allgemein folgendermaßen formuliert sein: *Ist der direkte Weg von Ort A nach Ort B besser als jener über Ort C?*

Das SPP kann zum einen so formuliert sein, dass die Verbindung zwischen einem Start (*source*) und einem Ziel (*destination*) gesucht wird, Single Source Single Destination Problem (SSSDP), und zum anderen die Erreichbarkeit, ausgehend von einem Start zu allen übrigen Orten, zum Inhalt hat, Single Source Multiple Destination Problem (SSMDP).

Einen ausführlichen Überblick zu den Algorithmen, die Single Source Single Destination (SSSD) und Single Source Multiple Destination (SSMD) Probleme lösen, bieten unter anderem [Dom07], [FSR06], [ZN98], [HTB95] und [DP84]. [FSR06] unterteilt hierbei die angeführten Algorithmen in *Optimale kürzeste Wege Algorithmen* und *Heuristische kürzeste Wege Algorithmen*. Eine diffizilere Klassifizierung erfolgt bei [DP84]. Hier wird nach *Art des SPP*, *Charakteristik der zugrunde liegenden Netzwerkdaten* und *angewandtes Lösungsverfahren* unterschieden.

Die nachfolgende Einteilung erfolgt nach [FSR06], wobei sich die Nomenklatur der Datenbasis auf eine Knoten-Kanten Struktur (vgl. Abschnitt 3.1) bezieht.

2.1.1 Optimale kürzeste Wege

Die Bestimmung optimaler kürzester Wege teilt sich in *Label-setting* und *Label-correcting* Verfahren. Dabei werden die Erreichungskosten (*Label*) des jeweiligen Knotens von einem ausgesuchten Startknoten bestimmt und im Zuge der fortschreitenden Ausbreitung des Suchbaumes angepasst.

Hat der Suchbaum im Verlauf seiner Ausbreitung einen Knoten erreicht, so ist dessen Erreichbarkeit vom Startknoten aus inklusive der hierfür benötigten Kosten im Falle eines Label-setting Verfahrens als bekannt und als unveränderlich anzusehen. Diese Verfahren sind unter Berücksichtigung der zu analysierenden Datenstruktur und deren Zugriff bei [Dij59] (*sortierte Listen*), [Dia69] (*bucket sort*) und [Wil64] (*heap sort*) ausführlich beschrieben. Das Hauptanwendungsgebiet von Label-setting Algorithmen ist laut [FSR06] in der Lösung von SSSD Problemen zu sehen.

Bei Label-correcting Verfahren, wie sie beispielsweise bei [Dom07] beschrieben sind, können sich die Erreichungskosten einzelner Knoten im Zuge der Suchbaumausbreitung immer wieder ändern. Somit steht die Kenntnis des optimalen kürzesten Weges erst nach abgeschlossener Analyse der kompletten Knoten-Kanten Struktur zur Verfügung, wodurch der bevorzugte Einsatz bei SSMD Problemen zu sehen ist. Eine Unterteilung kann anhand ihrer Datenverarbeitung angestellt werden: *queue* Algorithmen, [Moo57] bzw. FIFO-Verfahren, *double ended queue* Algorithmen, [Pap74] und *threshold lists* Algorithmen, [GKP85].

2.1.2 Heuristische kürzeste Wege

Das grundsätzliche Wesen heuristischer Ansätze ist in der Nutzung zusätzlichen Wissens und statistischer Vermutungen zu sehen, mit der Einschränkungen in der Datenbasis getroffen oder Tendenzen der Ausbreitung des Suchbaumes gesteuert werden. Durch diese Eingriffe in ein optimal lösbares System versucht man Vorteile in der Berechnungsgeschwindigkeit einer Route auf Kosten streng optimaler Lösungen zu erhalten.

Die Einteilung der Algorithmen zur heuristischen Bestimmung kürzester Wege nach [FSR06] erfolgt nach räumlicher Einschränkung des Suchbereiches (*limit the search area*), Unterteilen des Suchproblems (*decompose the search problem*) und expliziter Beschränkung von Nachbarschaften (*limit the search links*). Kombinationen aus Algorithmen, die diesen drei Gruppen zugeordnet werden können, sind ebenfalls erwähnt.

Bei räumlichen Einschränkungen wird hierbei zwischen einer Vorabauswahl der Basisdaten und einer Validierung während der Suchbaumausbreitung unterschieden. Werden im Vorfeld der Routenberechnung die potentiellen Daten durch dynamisch statistische Auswahlmechanismen festgelegt, stehen nur diese bei der Suchbaumausbreitung zur Verfügung. Diese Methode wird als *branch pruning* bezeichnet und unter anderem bei [Fu96], [Kar96] bzw. [Kor85] beschrieben. Im Gegensatz dazu kann bei A^* Algorithmen (vgl. [HNR68], [Nil71], [Poh71] und [Pea84]) auf alle Basisdaten zugegriffen werden, jedoch werden jene Elemente bevorzugt für die Suchbaumausbreitung herangezogen, deren Beteiligung an der gesuchten Route als wahrscheinlich zu bewerten ist. Eine Kombination aus *branch pruning* Verfahren und A^* Algorithmen ist bei [Lys95] dokumentiert.

Für die Unterteilung des gesamten Shortest Path (SP) Problems in mehrere SP Probleme führt [FSR06] bidirektionale Methoden bzw. die Definition von Teilzielen (*subgoal method*), vergleichbar mit vordefinierten Wegpunkten, die es zu durchlaufen gilt, an. Bidirektionale Verfahren sind etwa bei [Nic66] beschrieben. Hier werden zwei Suchbäume ausgebreitet: einer vom Start und der andere vom Ziel aus. Überlappen sich die beiden Suchbäume in einer vordefinierten Weise, so kann die Ausbreitung beendet und die Route extrahiert werden. Bei der Definition von Teilzielen, wie sie bei [YWH06] oder etwa als *island search* Algorithmen bei [DN95] beschrieben werden, zielt man auch auf eine geringere Anzahl an zu analysierenden Elementen und somit einer Zeitersparnis ab. Dies erreicht man durch die Definitionen von Elementen, durch welche die gesuchte Route zu führen hat und die zugleich als Start- und Zielelemente mehrerer Suchbäume dienen.

Die letzte eigenständige Gruppe der heuristischen Verfahren zur Bestimmung kürzester Wege umfasst nach [FSR06] jene Methoden, die sich auf die Einschränkung von Nachbarschaften verstehen. Obwohl ein Knoten durch seine Kanten mit einer bestimmten Menge an Nachbarknoten verbunden ist, werden hierbei nicht alle zur Verfügung stehenden Kanten genutzt, sondern auf eine Untermenge reduziert. Eine geringe Wahrscheinlichkeit an der gesuchten Route beteiligt zu sein, beziehungsweise Erfahrungswerte aus der Praxis sind Kriterien, auf die in einer solchen Reduktion Rücksicht genommen wird. Umgesetzt werden solche Strategien in hierarchischen Suchbaumausbreitungen, wobei Vergleiche mit der Thematik *abstraction problem solving strategy* aus der künstlichen Intelligenz durchaus zulässig sind.

Die Nutzung hierarchischer Strukturen, speziell ausgerichtet auf das Verkehrsnetz, wird bereits 1945 in einer Publikation von Plolya erwähnt (vgl. [FSR06, S. 3336]) und in weiterer Folge unter anderem von [Sac74] und [Kor87] beschrieben. Bei hierarchischen Verfahren sind im Vergleich zu den zuvor angeführten Algorithmen zwei wesentliche Fragen zu beantworten: Eine bezogen auf die Datenmodellierung (*Wie unterteilt man ein vorhandenes Netzwerk in verschiedene Hierarchien und wie sind die Übergänge zwischen diesen zu modellieren?*) und eine hinsichtlich des Analyseprozesses (*Wann wechselt der Suchbaum im Zuge der Ausbreitung in eine andere hierarchische Ebene?*). Diese Fragestellungen wurde bei [CF94] unter Zuhilfenahme von bidirektionalen Ansätzen beschrieben. Als herausfordernd für hierarchische Algorithmen sind laut [Liu97] Fragestellungen bei kurzen und mittellangen Routen oder etwa die Nutzung von Abkürzungen innerhalb hierarchischer Datenstrukturen zu sehen.

2.2 Algorithmus nach Dijkstra

Nach dem vorangegangenen Überblick über optimale und heuristische Ansätze zur Ermittlung kürzester Wege folgt in diesem Abschnitt eine detaillierte Beschreibung eines Algorithmus nach *Dijkstra*, [Dij59], ein Label-setting Algorithmus zur Bestimmung eines optimalen kürzesten Weges.

In der Literatur finden sich Leistungsvergleiche zwischen den unterschiedlichsten Routing-Algorithmus. [CGR93] kommt bei einem solchen Vergleich zum Schluss, dass ein *Dijkstra*-Algorithmus, angewandt auf synthetische Netze, das schnellste Ergebnis liefert. Bei einer Untersuchung von [ZN98] werden die Leistungen der Algorithmen an realen Daten des US-Bundesstaates Missouri getestet. Dabei wird festgestellt, dass für SSMD Fragestellungen der PAPE Algorithmus, [Pap74] bzw. TWO_Q von [Pal84] und für SSSD Fragestellungen ein *Dijkstra*-Algorithmus das schnellste Ergebnis liefert. Aus diesen Untersuchungsergebnissen heraus begründet sich die getroffene Wahl des Dijkstra-Algorithmus für diese Arbeit.

Nachfolgend werden Eigenschaften von Basis- und Verbindungselementen beschrieben, die dem Algorithmus als Datenbasis dienen. Die eigentliche Funktion des Algorithmus wird unter der Überschrift *Ausbreitung des Suchbaumes* erläutert. Wie man das Ergebnis des Algorithmus – die Route – aus den Daten extrahieren kann, hat der letzte Abschnitt zum Inhalt.

2.2.1 Das Basiselement

Das Basiselement $\beta \in B\{\beta_1, \beta_2, \dots, \beta_n\}$ bildet die Grundlage eines zu analysierenden Datenbestandes. Liegt beispielsweise eine Knoten-Kanten Struktur vor, so repräsentiert der Knoten das Basiselement. In [Dij59] wird dieses Element ebenfalls als *point* bzw. als *node* bezeichnet. Verallgemeinernd stellt ein Basiselement jenen Ort in einem Netzwerk dar, an dem Entscheidungen getroffen werden müssen: *Welches Basiselement soll als nächstes genutzt werden, um einen optimalen kürzesten Weg zu finden?* Auch Start β_S und Ziel β_D einer gesuchten Route gehören der Gruppe der Basiselemente an. Somit kann eine Route als eine Aneinanderreihung von Basiselementen gesehen werden:

$$route = \{\beta_S, \beta_{S+1}, \dots, \beta_i, \beta_j, \dots, \beta_{D-1}, \beta_D\}. \quad (2.1)$$

Für die Ausbreitung des Suchbaumes sind dem Basiselement verschiedene Eigenschaften zuzuweisen, die zum einen ihren Wert im Verlauf der Nachbarschaftsanalyse unverändert beibehalten (${}^c\beta$), und zum anderen je nach Fortschritt der Analyse verändern (${}^l\beta$ und ${}^p\beta$):

- Kosten (*cost*) ${}^c\beta$: Nutzungskosten des jeweiligen Basiselementes.
- Labelwert (*label*) ${}^l\beta$: Summe aller Kosten von Basiselementen und Verbindungselementen, die mit dem Start beginnend entlang des Astes eines Suchbaumes bis hin zum jeweiligen Basiselement durchlaufen werden.
- Vorgänger (*predecessor*) ${}^p\beta$: Jenes Basiselement, das sich im Ast des Suchbaumes vom Startelement gesehen direkt vor dem jeweiligen Basiselement befindet (${}^p\beta_j \equiv \beta_i$ bezogen auf Formel 2.1).

Im Gegensatz zu den in Abschnitt 2.1.1 angeführten Varianten, in denen die Basisdaten jeweils vorgehalten werden, wird in der Arbeit für die konzeptionelle Auseinandersetzung mit der Thematik ein relationales Datenmodell eingesetzt. In der Tabelle 2.1 ist der

grundsätzliche Aufbau der Basiselementerelation dargestellt. Die Zusätze PS bzw. FS stehen für Primärschlüssel und Fremdschlüssel, die für eine normalformkonforme Erstellung der Relationen benötigt werden.

Tabelle 2.1: Relationale Darstellung der Eigenschaften von Basiselementen.

Id [PS]	Kosten	Labelwert	Vorgänger [FS]
β_1	${}^c\beta_1$	${}^l\beta_1$	${}^p\beta_1$
β_2	${}^c\beta_2$	${}^l\beta_2$	${}^p\beta_2$
\dots	\dots	\dots	\dots
β_n	${}^c\beta_n$	${}^l\beta_n$	${}^p\beta_n$

2.2.2 Das Verbindungselement

Die Beschreibung der einzelnen Nachbarschaftsverhältnisse innerhalb einer Datenstruktur ist durch das Verbindungs- bzw. Adjazenzelement α realisiert. In der Literatur findet sich unter anderem folgende Benennungen hierfür: *branch* bei [Dij59], *edge* (ungerichteter Graph) und *arc* (gerichteter Graph) bei [HLW03], *link* bei [FSR06] und *Kante* (ungerichteter Graph) und *Pfeil* (gerichteter Graph) bei [Dom07].

Ist die Reihenfolge einer Nachbarschaftsdefinition strikt einzuhalten, also kann β_j zwar von β_i aus erreicht werden, aber die umgekehrte Reihenfolge ist nicht zulässig, so liegt ein gerichteter Graph vor, dessen Verbindungselement in folgender Weise definiert ist: $\alpha = (\beta_i, \beta_j) \in A\{\alpha_1, \alpha_2, \dots, \alpha_n\}$. Bei ungerichteten Graphen, in denen die Richtung einer Nachbarschaft nicht definiert wird, ist das Verbindungselement als $\alpha = [\beta_i, \beta_j] \in A\{\alpha_1, \alpha_2, \dots, \alpha_n\}$ beschrieben. Hierbei lehnen sich beide Definitionen an der Nomenklatur von [Dom07] an.

Ein sehr repräsentatives Beispiel für die Notwendigkeit gerichteter Graphen stellt eine Einbahnstraße innerhalb eines Straßennetzes dar. Hier ist die Fahrtrichtung, in der die Straße zu befahren ist bzw. von welcher Kreuzung aus kommend man zu welcher Kreuzung fahren darf, fix vorgegeben. Ein primärer Anwender für Routenberechnungen basierend auf ungerichteten Graphen ist in einem Fußgänger zu sehen. Seine Fortbewegungsmöglichkeiten sind größten Teils keinen Restriktionen unterworfen, sieht man von einer Rolltreppe ab.

Nachfolgend sind jene Eigenschaften angeführt, die für eine Suchbaumausbreitung benötigt werden. Hinsichtlich der weiteren Arbeit wird eine Einschränkung auf Verbindungselemente eines gerichteten Graphen vorgenommen. Verglichen mit den Eigenschaften des Basiselementes, sind alle Eigenschaften zeitinvariant und verändern sich im Laufe des Nachbarschaftsanalyseprozesses nicht:

- Kosten (*cost*) ${}^c\alpha$: Nutzungskosten des jeweiligen Verbindungselementes.
- Vorgänger (*predecessor*) ${}^p\alpha$: Basiselement, das in der definierten Nachbarschaft α als erstes zu durchlaufen ist ($\beta_i \equiv {}^p\alpha$ bei $\alpha = (\beta_i, \beta_j)$).
- Nachfolger (*successor*) ${}^s\alpha$: Basiselement, das in der definierten Nachbarschaft α als zweites zu durchlaufen ist ($\beta_j \equiv {}^s\alpha$ bei $\alpha = (\beta_i, \beta_j)$).

In Tabelle 2.2 ist der Aufbau der Verbindungselemente in relationaler Darstellung, unter der ergänzenden Kennzeichnung von Primär- und Fremdschlüsselementen, ersichtlich.

Tabelle 2.2: Relationale Darstellung der Eigenschaften von Verbindungselementen.

Id [PS]	Kosten	Vorgänger [FS]	Nachfolger [FS]
α_1	${}^c\alpha_1$	${}^p\alpha_1$	${}^s\alpha_1$
α_2	${}^c\alpha_2$	${}^p\alpha_2$	${}^s\alpha_2$
\dots	\dots	\dots	\dots
α_n	${}^c\alpha_n$	${}^p\alpha_n$	${}^s\alpha_n$

2.2.3 Ausbreitung des Suchbaumes

Die Menge an Basis- und Verbindungselementen aus den Abschnitten 2.2.1 und 2.2.2 bildet die Datengrundlage, den gerichteten und bewerteten Graph $G = (B, A, {}^cB, {}^cA)$, der im Verlauf der Suchbaumausbreitung hinsichtlich seiner Nachbarschaftsverhältnisse analysiert wird, um eine optimale kürzeste Verbindung zwischen einem Start- und einem Zielelement zu finden.

Je nachdem wie groß der Suchbaum ausgebreitet werden soll, ist einerseits ein Startelement β_S und ein Zielelement β_D , oder andererseits nur ein Startelement β_S zu definieren. Im erste Fall wird der Algorithmus bei Erreichen des definierten Zielelementes die Suchbaumausbreitung beenden und man erhält eine Lösung des SSSD Problems. Im zweite Fall wird der Suchbaum so lange ausgebreitet, bis alle verfügbaren Elemente darin vorkommen und dadurch eine Lösung des SSMD Problems vorliegt.

Das Kriterium, den optimalen kürzesten Weg zu erhalten, kann in Abhängigkeit der Datengrundlage – positive bzw. negative Graphenbewertung – formuliert werden. [HLW03] unterscheidet hierbei zwischen einer Maximierung der Vorteile bei einer positiven Bewertung – je höher ein Element bewertet ist, desto vorteilhafter ist der Gebrauch für die gesuchte Route zu sehen – mit $\prod_{path} p_{ij} \stackrel{!}{=} max.$ und einer Minimierung der Nachteile bei negativer Bewertung – je höher ein Element bewertet ist, umso geringer ist der Nutzen für die gesuchte Route zu sehen (vgl. ${}^c\alpha$) – mit $\sum_{path} c_{ij} \stackrel{!}{=} min..$ In beiden Fällen beschränkt sich die Bewertung auf das Verbindungselement α des Graphen. In dieser Arbeit wird jenes Kriterium für negativ bewertete Graphen herangezogen und um die Nutzungskosten der Basiselemente ${}^c\beta$ erweitert. Somit ergibt sich für die Ausbreitung des Suchbaumes innerhalb des Graphen folgende Bedingung:

$$\sum_{route} {}^c\beta + \sum_{route} {}^c\alpha \stackrel{!}{=} min.. \quad (2.2)$$

Die eigentliche Ausbreitung des Suchbaumes entsteht durch die Berechnung und Zuweisung von Labelwerten und die damit verbundene Einteilung der Basiselemente β zu drei verschiedenen Mengengruppen, wobei eine zeitgleiche Zuweisung eines Basiselementes zu mehreren Gruppen nicht zulässig ist:

- Basiselement wurde vom Suchbaum bisher noch nicht erreicht; kein Labelwert zugewiesen (*unlabelled*): $U = \{\beta_{U_1}, \beta_{U_2}, \dots, \beta_{U_n}\} \subseteq B$.
- Basiselement wurde vom Suchbaum bereits erreicht. Der kürzeste Weg vom Startelement zu diesem konnte aber noch nicht bestimmt werden; vorübergehender Labelwert zugewiesen (*temporally labelled*): $T = \{\beta_{T_1}, \beta_{T_2}, \dots, \beta_{T_n}\} \subseteq B$.

- Basiselement ist Bestandteil des Suchbaumes. Der kürzeste Weg vom Startelement zu diesem Element wurde bestimmt; bestehender Labelwert fixiert (*permanently labelled*):
 $P = \{\beta_{P_1}, \beta_{P_2}, \dots, \beta_{P_n}\} \subseteq B$.

Algorithmus 1 Allgemeine Suchbaumausbreitung nach Dijkstra.

```

1:  $U \leftarrow B$ 
2:  ${}^l\beta := \infty$ 
3:  $T \leftarrow \{\}$ 
4:  $P \leftarrow \{\}$ 
5:  $T \leftarrow T + \{\beta_S\}$ 
6:  $U \leftarrow U \setminus \{\beta_S\}$ 
7:  ${}^l\beta_S := {}^c\beta_S$ 
8: repeat
9:    $\beta_i := \{\beta_j \in T \mid {}^l\beta_j = \min({}^l\beta)\}$ 
10:   $P \leftarrow P + \{\beta_i\}$ 
11:   $T \leftarrow T \setminus \{\beta_i\}$ 
12:   $A_k \leftarrow \{A_l \in A \mid {}^p\alpha_l = \beta_i\}$ 
13:  for all  $\alpha_r \in A_k$  do
14:    if  $(\beta_{s_{\alpha_r}} \notin P) \wedge ({}^l\beta_{s_{\alpha_r}} > {}^l\beta_i + {}^c\alpha_r + {}^c\beta_{s_{\alpha_r}})$  then
15:       ${}^l\beta_{s_{\alpha_r}} := {}^l\beta_i + {}^c\alpha_r + {}^c\beta_{s_{\alpha_r}}$ 
16:       ${}^p\beta_{s_{\alpha_r}} := \beta_i$ 
17:      if  $\beta_{s_{\alpha_r}} \in U$  then
18:         $T \leftarrow T + \{\beta_{s_{\alpha_r}}\}$ 
19:         $U \leftarrow U \setminus \{\beta_{s_{\alpha_r}}\}$ 
20:      end if
21:    end if
22:  end for
23: until  $\beta_D \in P$ 

```

In Algorithmus 1 ist der funktionelle Ablauf der Suchbaumausbreitung hinsichtlich des Optimierungskriteriums aus Formel 2.2 dargestellt. In den ersten Zeilen (1 bis 4) werden Wert- und Mengenzuweisungen beschrieben, die eine Ausgangssituation unabhängig von Start und Ziel bieten. Die Menge aller Basiselemente B wird der Menge U zugewiesen und jedes Basiselement β erhält den Labelwert ∞ . In den Zeilen 5 und 6 wird das Startelement definiert. Nur dieses Basiselement wird der Menge T zugewiesen und der Menge U entnommen. Gleichzeitig wird der Labelwert des Startelementes seinem Kostenwert gleichgesetzt (Zeile 7). Die durch die Schleifenkonstruktion (Zeile 8 und Zeile 23) zu wiederholenden Anweisungen enden entweder nach Erreichen des Zielelementes (Zeile 23) oder bei SSMD Problemen nach der Ausbreitung des Suchbaumes über den gesamten Graphen. Hierfür ist die Zeile 23 in **until** $G \in P$ abzuändern.

Innerhalb dieser Schleife wird anfangs jenes Basiselement gesucht, das der Menge T zugewiesen ist und den kleinsten Labelwert aufweist. Anm.: Im Fall des ersten Schleifendurchlaufes trifft dieses Auswahlkriterium auf das Startelement zu. Dieses ausgewählte Basiselement wird der Menge P zugewiesen (Zeile 10) und aus der Menge T entnommen (Zeile 11). Dieser Prozess erweitert den Suchbaum um ein zusätzliches Element bzw. Ast.

Die Nachbarschaftsanalyse beginnt in Zeile 12, in der all jene Verbindungselemente gesucht werden, deren Vorgängereigenschaft auf das aktuelle Basiselement verweist. Die ge-

fundenen Verbindungselemente werden in einer weiteren Schleife analysiert (Zeile 13 bis Zeile 22). Das Optimierungskriterium aus Formel 2.2 wird für die Formulierung eines Entscheidungsmaßes für die Prozesse der Zeilen 15 bis 20 in Zeile 14 eingeführt. Diese Prozesse werden jedoch nur bei Nachfolgeelementen ausgeführt, die nicht der Menge P angehören und einen Labelwert besitzen, der größer als der aktuell berechenbare ist und somit auch verbessert werden kann. Treffen die zwei Kriterien zu, so wird der aktuell berechenbare Labelwert dem Nachfolgeelement zugewiesen (Zeile 15) und als Vorgänger des Nachfolgeelementes der Vorgänger des Verbindungselementes eingetragen (Zeile 16). Sollte das untersuchte Nachfolgeelement der Menge U angehören (Zeile 17), so wird dieses der Menge T zugewiesen und aus der Menge U entnommen (Zeilen 18 und 19).

Sobald das Abbruchkriterium zutrifft, ist der optimal kürzeste Weg zwischen Start und Ziel oder Start und allen anderen Elementen, je nach Problemstellung (SSSD oder SSMD) und dem darauf abgestimmten Abbruchkriterium, bekannt. Die Rekonstruktion der ermittelten Route ist anschließend in Abschnitt 2.2.4 erläutert.

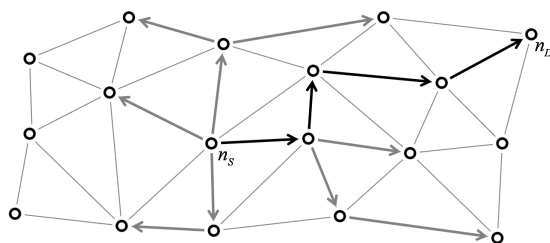


Abbildung 2.1: Knoten-Kanten Struktur mit darin ausgebreitetem Suchbaum zur Ermittlung einer Route zwischen dem Startelement n_S und dem Zielelement n_D .

Eine beispielhafte Darstellung einer Suchbaumausbreitung im Zuge einer SSSD Aufgabenstellung ist in Abbildung 2.1 aufbereitet. Den Hintergrund dieser Abbildung stellt eine Knoten-Kanten Struktur gemäß Abschnitt 3.1 dar. Die Knoten sind als schwarze Kreise und die Kanten als dünne graue Linien ausgewiesen. Mittels grauer und schwarzer dicker Pfeile wird der Suchbaum, ausgehend vom Startelement der Ausbreitung n_S , zu jenem Zeitpunkt dargestellt, an dem das Abbruchkriterium der Ausbreitung erfüllt wurde: Das Zielelement der gesuchten Route n_D wurde Bestandteil der Menge P . Die Kanten der ermittelten Route sind hierbei als dicke schwarze Pfeile gekennzeichnet.

Zusätzlich können die Eingangs- (*indegree*) $d_{in}(n_i)$ und Ausgangsgrade (*outdegree*) $d_{out}(n_i)$ der einzelnen Basiselemente des Baumes erkannt werden (vgl. $d_{n_i}^-$ und $d_{n_i}^+$ bei [Dom07], $d_{in}(n_i)$ und $d_{out}(n_i)$ bei [Jun05] oder $id(n_i)$ und $od(n_i)$ bei [CH91]): Abgesehen vom Startelement n_S , das einen Eingangsgrad $d_{in}(n_S) = 0$ besitzt, führt exakt *ein* Pfeil zu einem Knoten ($d_{in}(n_i) = 1$). Die Ausgangsgrade der Knoten – die Anzahl der abgehenden Pfeile – sind unterschiedlich: Befindet sich der Knoten nicht am Ende eines Suchbaumastes, so weist er einen Ausgangsgrad von $d_{out}(n_i) > 0$ auf. Nur jene Knoten am Ende eines Astes, wie in diesem Beispiel u.a. n_D , besitzen einen exakten Wert von Null ($d_{out}(n_D) = 0$). In Abschnitt 5.1 wird in weiterer Folge speziell auf die Eingangsgrade der Basiselemente eines Suchbaumes Bezug genommen.

2.2.4 Aufbereitung des Ergebnisses

Durch das in Abschnitt 2.2.3 beschriebene Verfahren, ist zu jedem untersuchten Basiselement β jenes Vorgängerelement ${}^p\beta$ ermittelt worden, das im Verlauf der gesuchten Route zuvor zu durchlaufen ist. Somit können die an der optimal kürzesten Route beteiligten Basiselemente nur vom Ziel β_D hin zum Start β_S identifiziert werden. Diese *route'* ist in Formel 2.3 als Aneinanderreihung von Basiselementen und den Vorgängerinformationen dargestellt. Durch die Umkehrung der Reihenfolge erhält man die gesuchte *route* nach Formel 2.1.

$$route' = \{\beta_D, {}^p\beta_D \equiv \beta_{D-1}, \dots, {}^p\beta_k \equiv \beta_j, {}^p\beta_j \equiv \beta_i, \dots, {}^p\beta_{S+2} \equiv \beta_{S+1}, {}^p\beta_{S+1} \equiv \beta_S\}. \quad (2.3)$$

Die Identifikation der beteiligten Basiselemente ist in Algorithmus 2 beschrieben. Initialisiert wird dieser Vorgang mit der Zuweisung des Zielelementes zur Menge *route'* (Zeilen 1 und 2). Zusätzlich wird das Vorgängerelement des Zieles in Zeile 3 identifiziert. Nachfolgend werden in einer Schleife (Zeile 4 bis Zeile 7) solange Vorgängerelemente der Menge *route'* zugewiesen (Zeile 5) und die dazugehörigen Vorgängerelemente identifiziert (Zeile 6), bis das Startelement Teil der Menge *route'* geworden ist.

Algorithmus 2 Extraktion der Route.

```

1: route'  $\leftarrow \{\}$ 
2: route'  $\leftarrow \beta_D$ 
3:  $\beta_i := {}^p\beta_D$ 
4: repeat
5:   route'  $\leftarrow \beta_i$ 
6:    $\beta_i := {}^p\beta_i$ 
7: until  $\beta_S \in route'$ 

```

Kapitel 3

Datenmodellierung

Die Modellierung der realen Welt vor dem Hintergrund der Routenberechnung aus Kapitel 2 hat das Kapitel *Datenmodellierung* zum Inhalt. Hier werden verschiedene Varianten aufgezeigt, mit denen eine Abstraktion des Umfeldes so erfolgen kann, dass die Ausbreitung von Suchbäumen darin möglich wird. Dieses Kapitel steht somit direkt mit den Inhalten des Kapitels 6 in Verbindung, in dem diese Modelle praktische Anwendung finden.

Die Gliederung dieses Kapitels erfolgte nach geometrischen Aspekten. Je nach Repräsentationsform des Basiselementes β gemäß dem Abschnitt 2.2.1 wird in 0- (Abschnitt 3.1), 1- (Abschnitt 3.2), 2- (Abschnitt 3.4) und 3-dimensionale (Abschnitt 3.5) Formen unterschieden. Zusätzlich zu dieser streng dimensionalen Trennung werden zwei Mischformen in Abschnitt 3.3 mit 0- und 1-dimensionalen Basiselementen beziehungsweise in Abschnitt 3.6 mit allen Arten von Basiselementen vorgestellt.

3.1 Knoten-Kanten Modelle

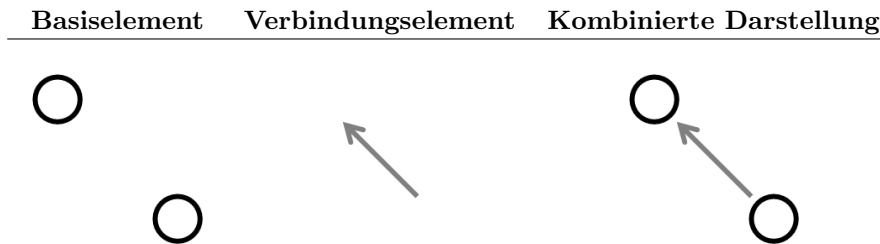
Jene Form der Modellierung, die in vielen Bereichen der Wissenschaft Anwendung findet, wird durch eine Knoten-Kanten Struktur dargestellt. In der bereits erwähnten *Tabula Peutingeriana*, [Oes10] können diese Strukturen erkannt werden: Die eingezeichneten Orte können als Knoten und die Verbindungslinien dazwischen als Kanten interpretiert werden.

In der ersten Hälfte des 18. Jahrhunderts wurde von *Leonhard Euler* (1707–1783) das *Königsberger Brückenproblem* formuliert (u.a. bei [GB05] beschrieben), das als Nukleus der Graphentheorie gesehen werden kann. Hier werden Landmassen und Brücken zu Knoten und Kanten abstrahiert. Weitere Persönlichkeiten, die mit der historischen Entwicklung der Graphentheorie in Verbindung gebracht werden können, sind *James Joseph Sylvester* (1814–1897), [Hal97] und *Arthur Cayley* (1821–1885), [Mac95]. Damals wie heute (z.B.: [Dom07], [Bar05], [Jun05], [HLW03] oder [CH91]) werden vorrangig Knoten und Kanten zur Modellierung von Netzwerken herangezogen, in denen Suchbäume ausgebreitet werden.

Bei dieser Modellierungsstrategie wird das Basiselement β durch den Knoten (*node*) $n \in V\{n_1, n_2, \dots, n_n\}$ – ein 0-dimensionales geometrisches Objekt – repräsentiert. Dieser Knoten kann je nach Problemstellung in einem \mathfrak{R}^1 , \mathfrak{R}^2 oder \mathfrak{R}^3 platziert werden. Das Verbindungselement α , das hier die Nachbarschaften der einzelnen Knoten n untereinander realisiert, wird durch die gerichtete Kante bzw. Pfeil (*arc*) $a = (n_i, n_j) \in E\{a_1, a_2, \dots, a_n\}$ dargestellt. Eine geometrische Repräsentationsform ist hierbei optional und kann durch

ein Linienelement mit verschiedenen vielen Zwischenpunkten erfolgen (vgl. *polygon points* bei [HLW03]). Zusammen bildet die Menge an Knoten (*vertices*) V und Kanten bzw. Pfeilen (*edges*) E den gerichteten und bewerteten Graph $G = (V, E, {}^cV, {}^cE)$, der als Datengrundlage einer Suchbaumausbreitung dient.

Tabelle 3.1: Basis- und Verbindungselement in einer Knoten-Kanten Struktur.



In Tabelle 3.1 werden die beiden Elementtypen dargestellt, wobei auf eine geometrische Repräsentanz des Verbindungselementes verzichtet wird. Die topologische Funktion, das Verbinden zweier Knoten, ist für die Suchbaumausbreitung wesentlich, im Gegensatz zur geometrischen Beschreibung, die hauptsächlich einer graphischen Darstellung dient.

Basierend auf der relationalen Darstellung der Eigenschaften von Basiselementen (Tabelle 2.1 aus Abschnitt 2.2.1) beinhaltet die Tabelle 3.2 die Eigenschaften von Knotenelementen n_i . Kosten von Knoten cn_i werden in der Literatur, beispielsweise bei [Dom07] oder [HLW03], nicht angeführt, werden hier jedoch im Rückblick auf Abschnitt 2.2 für die Knoten beibehalten. Ergänzt wird die Relation um eine geometrische Eigenschaft: Die Open Geospatial Consortium (OGC) konforme Beschreibung im Well Known Text (WKT) Format einer 0-dimensionalen Geometrie in einem \mathfrak{R}^2 mit dem Rechtswert xn_i und dem Hochwert yn_i (vgl. [Ope06a]).

Tabelle 3.2: Relationale Darstellung der Eigenschaften von Basiselementen in einer Knoten-Kanten Struktur.

Id	[PS]	n_i
Kosten		cn_i
Labelwert		ln_i
Vorgänger	[FS]	pn_i
Geometrie		POINT(xn_i yn_i)

Die Adaption der relationalen Darstellung der Eigenschaften von Verbindungselementen der Tabelle 2.2 aus Abschnitt 2.2.2 für eine passende Strukturierung von Kanten a_j ist in der optionalen Erweiterung um ein Geometrieattribut zu sehen (Tabelle 3.3). Hier wird der Notwendigkeit, Zwischenpunkte setzen zu können, mit der Verwendung einer OGC konformen *LINESTRING* Geometrie entsprochen. Diese ist bei [Ope06a] so definiert, dass sie beliebig viele Zwischenpunkte (${}^{xi}a_j$ ${}^{yi}a_j$) zwischen Start- und Endpunkt der Linie (${}^{x1}a_j$ ${}^{y1}a_j$ und ${}^{xn}a_j$ ${}^{yn}a_j$) beinhalten kann.

Tabelle 3.3: Relationale Darstellung der Eigenschaften von Verbindungselementen in einer Knoten-Kanten Struktur.

Id	[PS]	a_j
Kosten		$^c a_j$
Vorgänger	[FS]	$^p a_j$
Nachfolger	[FS]	$^s a_j$
Geometrie	(optional)	$\text{LINESTRING}(^{x_1} a_j \ ^{y_1} a_j, \ ^{x_2} a_j \ ^{y_2} a_j, \dots, \ ^{x_n} a_j \ ^{y_n} a_j)$

Die bei [Dij59] als Entfernung (*distance* bzw. *length*) beschriebenen Kosteninformationen, werden in Tabelle 3.3 allgemein als Kosten $^c a_j$ bezeichnet. Diese sind je nach Art der Problemstellung zu definieren. Steht eine geometrische Information für eine Kante zur Verfügung, so können Kostenwerte mit Hilfe von spatial Structured Query Language (spatialSQL) Anweisungen gemäß [Ope06b] aus dieser abgeleitet werden. Die Analysefunktion `st_length(<geometry>)`, angewandt auf eine *LINESTRING* Geometrie, liefert die Länge dieser Geometrie. Sollte keine Geometrie des Verbindungselementes vorhanden sein, so kann beispielsweise die Euklidische Entfernung mittels `st_distance(<geometry>, <geometry>)` zwischen den beiden zu verbindenden Knoten ermittelt werden und in den Kostenwert der Kante einfließen.

3.2 Kanten-Traversen Modelle

Die Modellierung von Netzwerken als Kanten-Traversen Strukturen ist thematisch eng mit der Berücksichtigung von Abbiegeverboten oder der Bewertung von Abbiegemanövern in Straßennetzen im Zuge der Suchbaumausbreitung verbunden. Bei [Cal61], [Win02] oder [GM07] finden sich Modellierungsverfahren, die hierfür *pseudo*, *pseudo-duale* bzw. *duale* Graphen einsetzen. Das darin beschriebene Prinzip beruht auf der Verwendung des Kanten-Elementes a als Basiselement β und die Aufspaltung des Knoten-Elementes n zu mehreren Verbindungselementen α . Dieser Weg der Modellierung, das vermeintlich bloße Vertauschen von Knoten und Kanten, ist jedoch als eine evolutionäre Konsequenz der Knoten-Kanten Struktur zu sehen: Es werden Basiselemente durch Verbindungselemente in Bezug gebracht. Waren es in Abschnitt 3.1 die Kanten, die die Nachbarschaft eines Knoten realisierten, so werden durch ein neues Element die Nachbarschaften von Kanten definiert. Dieses neue Element wird bei [Bar91] bzw. bei [Wie91] als *Traverse* bezeichnet.

In Tabelle 3.4 sind die Elemente eines Netzwerkes bei Kanten-Traversen Modellierung abgebildet. Das Basiselement wird hierbei von einer Kante (*arc*) $a \in E\{a_1, a_2, \dots, a_n\}$ dargestellt, die die Information, in welche Richtung sie durchlaufen werden darf, nicht zu beinhalten braucht. Die richtungsabhängige Verbindung zwischen zwei Basiselementen wird durch eine Traverse (*traverse*) $t = (a_i, a_j) \in T\{t_1, t_2, \dots, t_n\}$ beschrieben. Daraus ergibt sich der gerichtete und bewertete Graph $G = (E, T, ^c E, ^c T)$ als Datengrundlage für eine Suchbaumausbreitung.

Die Modellierung einer Einbahnstraße, die in Knoten-Kanten Strukturen durch die explizite Angabe von Ausgangskreuzung $^p a_i$ und Zielkreuzung $^s a_i$ des Verbindungselementes a_i erfolgt, kann hier mit den Beziehungen zwischen Ausgangsstraße $^p t_r$ und Einbahnstraße $^s t_r$ der Traverse t_r bzw. Einbahnstraße $^p t_s$ und Zielstraße $^s t_s$ der Traverse t_s umgesetzt werden.

Tabelle 3.4: Basis- und Verbindungselement in einer Kanten-Traversen Struktur.



Eine Auflistung der für die Suchbaumausbreitung notwendigen Eigenschaften des Basiselementes findet sich in Tabelle 3.5. Hier ist die Angabe der Kosteninformation $^c a_i$ eines Basiselementes β , verglichen mit jener Kosteninformation eines Knotens $^c n_j$ aus Abschnitt 3.1, das Basiselement einer Knoten-Kanten Struktur, jedoch ein wesentlicher Bestandteil für die Modellierung der realen Welt. Eng verknüpft mit der Definition von Kostenwerten ist eine geometrische Modellierung dieses Basiselementes als ein 1-dimensionales Objekt in einem \mathfrak{R}^1 , \mathfrak{R}^2 oder \mathfrak{R}^3 zu sehen (vgl. *LINESTRING* Geometrie gemäß [Ope06a]). Wie im vorangegangenen Abschnitt bereits beschrieben, kann über die Anwendung von diversen Geometrieanalysefunktionen nach [Ope06b], wie etwa `st_length(<geometry>)`, relevante Information über diese Geometrie extrahiert werden und bei der Definition eines Kostenwertes mit einfließen.

Tabelle 3.5: Relationale Darstellung der Eigenschaften von Basiselementen in einer Kanten-Traversen Struktur.

Id	[PS]	a_i
Kosten		$^c a_i$
Labelwert		$^l a_i$
Vorgänger	[FS]	$^p a_i$
Geometrie		$\text{LINESTRING}(^{x_1} a_i \ ^{y_1} a_i, \ ^{x_2} a_i \ ^{y_2} a_i, \dots, \ ^{x_n} a_i \ ^{y_n} a_i)$

Bei den Eigenschaften von Traversen innerhalb einer Kanten-Traversen Struktur (Tabelle 3.6) bilden Vorgänger und Nachfolger ($^p t_j$ und $^s t_j$) die wesentlichen Bestandteile, die bei der Suchbaumausbreitung benötigt werden. Mit Hilfe der Kosteninformationen $^c t_j$ kann der Wechsel von einer Kante zu einer anderen Kante bewertet werden. Kreuzungssituationen können dahingehend modelliert werden, dass die Richtung eines Abbiegemanövers (z.B.: Links-Abbiegen) in den Kosten der Traverse Berücksichtigung findet. Eine (optionale) Angabe einer geometrischen Repräsentanz des Verbindungselementes, wie sie bei der Knoten-Kanten Struktur beschrieben werden kann, findet bei der Kanten-Traversen Struktur nicht statt. Da es sich hierbei um ein rein topologisches Element handelt, das die Nachbarschaft zweier Kanten definiert, ist eine geometrische Beschreibung nicht zweckmäßig.

Im Hinblick auf die Abschnitte 3.4 und 3.5 und die darin beschriebene Definition der Nachbarschaften – implizit durch die Berührung von Basiselementen – kann bei vorliegenden

Tabelle 3.6: Relationale Darstellung der Eigenschaften von Verbindungselementen in einer Kanten-Traversen Struktur.

Id	[PS]	t_j
Kosten		ct_j
Vorgänger	[FS]	pt_j
Nachfolger	[FS]	st_j

Kanten, bei denen der Endknoten einer Kante (Formel 3.1) mit dem Anfangsknoten einer Nachbarkante (Formel 3.2) exakt koordinativ übereinstimmen ($^{x_n}a_i \equiv ^{x_1}a_j \wedge ^{y_n}a_i \equiv ^{y_1}a_j$), ebenfalls eine implizite Nachbarschaftsdefinition genutzt werden.

$$a_i : \text{LINESTRING}(^{x_1}a_i \ ^{y_1}a_i, ^{x_2}a_i \ ^{y_2}a_i, \dots, ^{x_n}a_i \ ^{y_n}a_i). \tag{3.1}$$

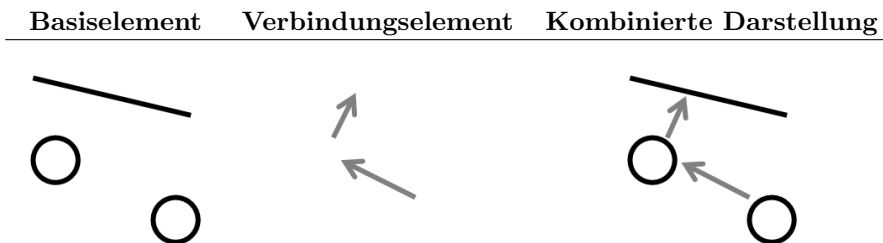
$$a_j : \text{LINESTRING}(^{x_1}a_j \ ^{y_1}a_j, ^{x_2}a_j \ ^{y_2}a_j, \dots, ^{x_n}a_j \ ^{y_n}a_j). \tag{3.2}$$

3.3 Gemischt-einfache Strukturen

Jene Aufgabenstellungen der Modellierung, bei denen eine Einschränkung weder auf Knoten n noch auf Kanten a als Basiselement β (vgl. Abschnitte 3.1 und 3.2) eine zufrieden stellende Lösung liefert, können durch die Definition einer gemischt-einfachen Struktur GES bearbeitet werden. Hierfür wird zugelassen, dass Knoten und Kanten gemeinsam die Menge der Basiselemente B^{GES} bilden. Vergleichbar dazu werden auch die Verbindungselemente α der Knoten-Kanten und Kanten-Traversen Strukturen (a und t) in der Menge der Verbindungselemente A^{GES} zusammengefasst. Gemeinsam bilden die kombinierten Basis- und Verbindungselemente den gerichteten und bewerteten Graph $G^{GES} = (B^{GES}, A^{GES})$.

Bei der Zusammenfassung von Knoten-Kanten und Kanten-Traversen Netzwerken sind zusätzliche Verbindungselemente zu definieren, deren Notwendigkeit bei getrennter Betrachtung der Daten nicht gegeben ist: Das Verbindungselement zwischen Knoten des Knoten-Kanten Netzwerkes und den Kanten des Kanten-Traversen Netzwerkes. Erst durch dieses zusätzliche Element können Suchbäume ohne Einschränkung auf eine bestimmte Datenstruktur über den kombinierten Datensatz ausgebreitet werden.

Tabelle 3.7: Basis- und Verbindungselement in einer gemischt-einfachen Struktur.



In Tabelle 3.7 sind beispielhaft Knoten und Kanten als Basiselemente abgebildet, die durch entsprechende Verbindungselemente in eine nachbarschaftliche Beziehung gebracht werden. Im speziellen sind zwei Verbindungen angeführt: Eine zwischen zwei Knoten – entspricht einer Kante bei einer Knoten-Kanten Modellierung – und eine Zweite zwischen einem Knoten und einer Kante. Dieses zweite Verbindungselement entspricht dem zuvor definierten Bindeglied zwischen Knoten-Kanten und Kanten-Traversen Strukturen. In der Tabelle ist auch die Richtung dieser Verbindung durch das Pfeilsymbol erkennbar, wobei nur eine Bewegung im gemeinsamen Netzwerk vom Knoten kommend hin zur Kante definiert wird.

Auswirkungen durch die Definition gemischt-einfacher Strukturen auf den Ausbreitungsvorgang des Suchbaumes entfallen, da die geometrische Repräsentanz der Basiselemente, nach der hier unterschieden wird, im Algorithmus nach Abschnitt 2.2 nicht zu berücksichtigen ist. Ob ein Basiselement durch eine Punkt- oder Liniengeometrie beschrieben oder die Verbindung zwischen zwei Basiselementen als Kante oder als Traverse bezeichnet wird und eine optionale geometrischen Repräsentationsform besitzt, ist nur für die Gestaltung der Datenbasis von Bedeutung, nicht jedoch für einen Algorithmus.

Soll die Datenbasis alle Arten von Basiselementen (Knoten und Kanten) in einer Relation vereinen, so unterscheidet sich die Definition dieser Relation, verglichen mit jener der Tabellen 3.2 und 3.5, nur im Geometrieattribut. Hier ist auf die gemeinsame Organisation von Punkt- und Linienobjekten zu achten. [Ope06a] definiert ein *GEOMETRYCOLLECTION* Objekt, in dem unter anderem Objekte von Typ *POINT* und *LINestring* kombiniert organisiert werden können.

3.4 Polygonale Modelle

Die Verknüpfung von flächenhaften Objekten mit der Thematik der Routenplanung ist vermeintlich nur schwer zu erkennen, kann aber bei verschiedenen Beispielen von räumlichen Analyseprozessen, wie etwa bei [Kis09], [PK09] oder [ACD⁺03], gefunden werden. Auch im Funktionsumfang von *ArcGIS*, einem Softwarepaket zur Organisation, Analyse und Präsentation räumlicher Daten befindet sich mit der *Cost weighted Distance* Funktion, [Env10a] jenes Werkzeug, mit dem sich Suchbäume basierend auf flächenhaften Daten ausbreiten lassen.

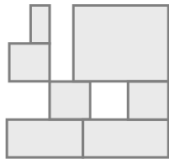
In dieser Arbeit werden die 2-dimensionalen Daten im Sinne der Vektorgeometrie behandelt: Polygone, wie sie bei [Int01] oder [Ope06a] definiert werden. Im Vergleich dazu handelt es sich bei der Datengrundlage der zuvor angeführten Beispiele in den meisten Fällen um Rasterdaten (z.B.: [RR97]). Der Methodik des Suchbaumausbreitens selbst tut diese Differenzierung in Raster- und Vektordatenformate jedoch keinen Abbruch und kann unabhängig davon beschrieben werden.

Der wesentliche Unterschied zu den Knoten-Kanten Modellen, den Kanten-Traversen Modellen oder den gemischt-einfachen Strukturen liegt in der Beschreibung der Nachbarschaft zwischen den Basiselementen β . Wird hierfür in den Abschnitten 3.1, 3.2 und 3.3 immer auf ein Verbindungselement α verwiesen, so kann im Fall einer polygonalen Modellierung zugunsten einer impliziten Nachbarschaftsbeschreibung auf das Verbindungselement verzichtet werden. Die Implikation der Nachbarschaft entsteht hierbei durch die Berührung der einzelnen Basiselemente untereinander.

Bei 2-dimensionalen Basisdaten kann diese Berührung entlang einer gemeinsamen Linie oder punktuell erfolgen (vgl. Abschnitt 6.2). Voraussetzung hierfür ist jeweils eine topologisch korrekt aufbereitete Datengrundlage. Daten, die einer *Pizza-Topologie* entsprechen, wie

sie bei [Bar05] beschrieben wird, bei denen Überlappungen von Polygonen vorliegen oder unbeabsichtigte Lücken vermeintliche Berührungen aufheben, sind einem Korrekturprozess zu unterziehen. Erst danach können diese Daten für eine Suchbaumausbreitung genutzt werden.

Tabelle 3.8: Basis- und Verbindungselement in einer polygonalen Struktur.

Basiselement	Verbindungselement
	Berührung von Basiselementen

Die daraus resultierende Datengrundlage für eine Suchbaumausbreitung besteht somit nur aus 2-dimensionalen Objekten (*polygon*) $p \in P\{p_1, p_2, \dots, p_n\}$ im \mathfrak{R}^2 und \mathfrak{R}^3 (siehe Tabelle 3.8). Die Richtung einer Nachbarschaftsbeziehung, wie sie bei den zuvor angeführten Modellierungsvarianten explizit im Verbindungselement durch Vorgänger ${}^p\alpha$ und Nachfolger ${}^s\alpha$ definiert werden kann, steht hier nicht zur Verfügung. Daraus folgt, dass bei einer Berührung von Polygon A mit Polygon B eine Nachbarschaft von A nach B und eine Nachbarschaft von B nach A beschrieben wird.

Tabelle 3.9: Relationale Darstellung der Eigenschaften von Basiselementen in einer polygonalen Struktur.

Id	[PS]	p_i
Kosten		${}^c p_i$
Labelwert		${}^l p_i$
Vorgänger	[FS]	${}^p p_i$
Geometrie		POLYGON($((x_1 p_i \ y_1 p_i, x_2 p_i \ y_2 p_i, \dots, x_1 p_i \ y_1 p_i)(\dots))$)

Die relationale Strukturierung der Basisdaten ist in Tabelle 3.9 dargestellt. Für die Beschreibung des 2-dimensionalen Objektes wird eine OGC konforme Nomenklatur nach [Ope06a] vom Geometrietyp *POLYGON* verwendet. Hierbei wird das Polygon durch eine sich schließende äußere Randlinie (*exterior ring*) und beliebig viele innere, sich ebenfalls schließende Randlinien (*interior ring*) für Aussparungen beschrieben (Formel 3.3). Anm.: Bei sich schließenden Liniengeometrien entspricht der erste Polygonpunkt koordinativ dem letzten Polygonpunkt.

$$\text{POLYGON}((\text{exterior ring})(\text{interior ring}_1)(\dots)(\text{interior ring}_n)). \quad (3.3)$$

Bei der Definition von Kostenwerten des Basiselementes ${}^c p_i$ kann beispielsweise die Fläche des geometrischen Objektes mit `st.area(<geometry>)` oder der Umfang, eine Kombination der Analysefunktionen `st.length(<geometry>)` und `st.exteriorring(<geometry>)`,

nach [Ope06b] als Kenngröße ermittelt werden und Berücksichtigung finden. Auch die Form des Polygons, wie etwa die Kompaktheit (z.B.: [BRVI00]), kann durch die Kombination von Flächenausmaß und Umfang für die Kostenberechnung einfließen. Ein anderer Ansatz für die Bewertung, eine *äquimonetäre Modellierung*, wäre die Aufteilung oder die Zusammenfassung von Polygonen, sodass alle geometrischen Objekte den gleichen Kostenwert aufweisen: Günstig zu benutzende Objekte besitzen eine größere Fläche als ungünstig zu benutzende Objekte. In Tabelle 3.10 sind beide Bewertungsstrategien gegenübergestellt, wobei die Färbung die Kosteneigenschaften widerspiegelt: Je dunkler eine Fläche dargestellt wird, umso ungünstiger ist deren Benutzung.

Tabelle 3.10: Bewertungsverfahren bei polygonalen Struktur.

Individuelle Bewertung	Äquimonetäre Modellierung
------------------------	---------------------------



Ein alternativer Ansatz zur Organisation von flächenhaften Datengrundlagen, der auf die bekannte Knoten-Kanten Struktur zurück greift, erfolgt durch eine duale Repräsentationsform. Bei [Bar05] wird diese Definition von Flächennachbarschaften in einer Knoten-Kanten Geometrie als *Regionsadjazenzgraph* beschrieben. Ein *Voronoi-Graph*, wie er bei [AK96] oder [OBS94] angeführt wird, befasst sich mit der selben Thematik der Nachbarschaftsbeschreibung.

Tabelle 3.11: Basis- und Verbindungselement in einer Knoten-Kanten Struktur abgeleitet aus einer polygonalen Struktur.

Basiselement	Verbindungselement	Kombinierte Darstellung
--------------	--------------------	-------------------------



Wie in Tabelle 3.11 aufbereitet, werden die Basiselemente mit 2-dimensionaler räumlicher Ausprägung p zu einem 0-dimensionalen Objekt n transformiert und hier im Zentrum des Polygons positioniert. Da ab diesem Augenblick eine Berührung der Basiselemente nicht mehr vorherrscht und somit die Nachbarschaftsverhältnisse nicht mehr implizit erkannt werden können, ist ein Verbindungselement a (vgl. Abschnitt 3.1) zu definieren. Dieses Verbin-

dungselement lässt eine Beschreibung der Richtung einer Nachbarschaft, wie sie beispielsweise in Tabelle 3.3 zu finden ist, wieder zu. Das Wissen über die zuzulassenden Richtungen im Sinne eines gerichteten Graphen ist bei einem Transformationsprozess von polygonalen Modellen zu Knoten-Kanten Strukturen aus zusätzlichen Informationsquellen mit einzubeziehen, da dieses in der rein geometrischen Beschreibung der flächenhaften Basisdaten nicht zur Verfügung steht.

Algorithmus 3 Erzeugen des Basiselementes des Regionsadjazenzgraphen aus einer polygonalen Struktur.

```

1:  $V \leftarrow \{\}$ 
2: INSERT INTO  $V$  ( $geometry$ ) SELECT st_centroid( $geometry$ ) FROM  $P$ 

```

Die Transformation der Basiselemente von einer polygonalen zu einer punktförmigen Repräsentationsform ist in Algorithmus 3 ausgewiesen. Hier wird unter der Verwendung einer räumlichen Analysefunktion – `st_centroid(<geometry>)` zur Bestimmung des Zentrums einer Flächengeometrie gemäß [Ope06b] – für jedes Polygon p der Menge P ein Knoten n in der Menge V angelegt (Zeile 2).

In Algorithmus 4 werden die Verbindungen zwischen den einzelnen Polygonen innerhalb der Schleifenkonstruktion zwischen Zeile 2 und Zeile 8 bestimmt. Für jedes Polygon p_i werden in Zeile 4 alle berührenden Polygone detektiert und temporär der Menge P^α zugewiesen. Die Analysefunktion `st_touches(<geometry>, <geometry>)` liefert bei einer Berührung den booleschen Wert `true`, ansonsten den Wert `false` (vgl. [Ope06b]). Diese Funktionsausgabe wird für die Formulierung einer Bedingung zur Erkennung von Nachbarpolygonen genutzt. Anschließend wird für jedes erkannte Nachbarpolygon p_j^α ein Verbindungselement erstellt: Das Ausgangspolygon p_i wird hierbei als Vorgänger pa und das dazugehörige Nachbarpolygon p_j^α als Nachfolger sa der Menge E als Kante zugewiesen (Zeile 6).

Algorithmus 4 Erzeugen des Verbindungselementes des Regionsadjazenzgraphen aus einer polygonalen Struktur.

```

1:  $E \leftarrow \{\}$ 
2: for all  $p_i \in P$  do
3:    $P^\alpha \leftarrow \{\}$ 
4:    $P^\alpha \leftarrow$  SELECT id FROM  $P$  WHERE st_touches( $geometry$ ,
      (SELECT  $geometry$  FROM  $P$  WHERE id =  $p_i$ ))
5:   for all  $p_j^\alpha \in P^\alpha$  do
6:     INSERT INTO  $E$  ( $^pa$ ,  $^sa$ ) VALUES ( $p_i$ ,  $p_j^\alpha$ )
7:   end for
8: end for

```

Bezüglich der Problematik der Richtung von Nachbarschaften wird bei dem in Algorithmus 4 beschriebenen Verfahren zwar ein gerichteter Graph $G = (V, E)$ abgeleitet, aber es werden für jede Nachbarschaft zwei gerichtete Kanten bzw. Pfeile angelegt. Sollte zusätzliche Information hinsichtlich einer eingeschränkten Nachbarschaft vorliegen, so kann durch Löschen einzelner Verbindungselemente eine Anpassung der bestehenden Menge E erfolgen.

3.5 Polyhedrale Modelle

Ein konsequentes Fortführen der bisher beschriebenen Modellierungsstrategien hat die Auseinandersetzung mit 3-dimensionalen Datenstrukturen zur Folge. Die Einbeziehung räumlicher Daten in einen Routenplanungsprozess geht mit der Bewegungsfreiheit des potentiellen Anwenders einher: Er muss sich in allen drei Raumrichtungen bewegen können. Dadurch eröffnen sich für die Navigation zusätzliche Anwendungsspektren hinsichtlich Ausbreitungssimulationen, Immissionsprognosen oder Versickerungsanalysen.

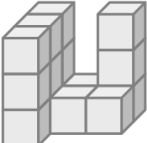
Die Notwendigkeit dieser Analyseprozesse und Prognosen zeigte ein Vorfall im April 2010, wo zu dieser Zeit der Vulkan *Eyjafjallajökull* auf Island eruptierte. Den größten ökonomischen Einfluss nahm dieses Naturphänomen auf den europäischen Luftverkehr, der aus Sicherheitsgründen vorübergehend gesperrt wurde (z.B.: [Gam10]). Die wissenschaftliche Grundlage für diese Sperre boten Simulationsrechnungen, wie etwa jene von [FNSE10], die auf einem Ausbreitungsmodell von [Mas07] die Bewegung der Aschewolke vorhersagten.

Die geometrische Beschreibung eines Mediums mit volumenhafter Repräsentanz, in der ein Suchbaum ausgebreitet werden kann, dient der Routenberechnung im Raum als Datengrundlage. Die Modellierungsverfahren hierfür sind wegen der Komplexität 3-dimensionaler Objekte umfassender als jene der 0-, 1- oder 2-dimensionalen Verfahren einzustufen. [Kad07] beschreibt zu diesem Thema drei grundlegende Ansätze der Festkörpermodellierung:

- Dekompositionsmodelle.
- Konstruktive Modelle.
- Modellierung mittels Randbeschreibung.

Im Hinblick auf Abschnitt 6.3 wird für die weitere Arbeit ein *Enumerationsverfahren* (z.B.: [KCY93]), das zu den Dekompositionsmodellen zählt, als volumenbeschreibende Nomenklatur verwendet. Die Form der einzelnen Basiselemente β entspricht einem Quader mit gleichlangen Kanten: Würfel (*cube*) $c \in C\{c_1, c_2, \dots, c_n\}$ im \mathfrak{R}^3 , der im Umfeld der Computergraphik auch als *Voxel* bezeichnet wird (vgl. Tabelle 3.12).

Tabelle 3.12: Basis- und Verbindungselement in einer polyhedralen Struktur.

Basiselement	Verbindungselement
	Berührung von Basiselementen

Verglichen mit den polygonalen Modellen aus Abschnitt 3.4, wird für die Realisierung der Nachbarschaftsverhältnisse zwischen den einzelnen Basiselementen ebenfalls ein implizites Verfahren gewählt: Sich berührende Würfel befinden sich in einer Nachbarschaft (vgl. Tabelle 3.12). Können Berührungen von Flächenobjekten durch geometrische Analysefunktionen,

wie `st_touches(<geometry>,<geometry>)` gemäß [Ope06b], erkannt werden, so sollte eine entsprechende geometrische Analyse für Volumenobjekte genauso zur Verfügung stehen. Das setzt jedoch eine geometrische Beschreibung der Würfel voraus!

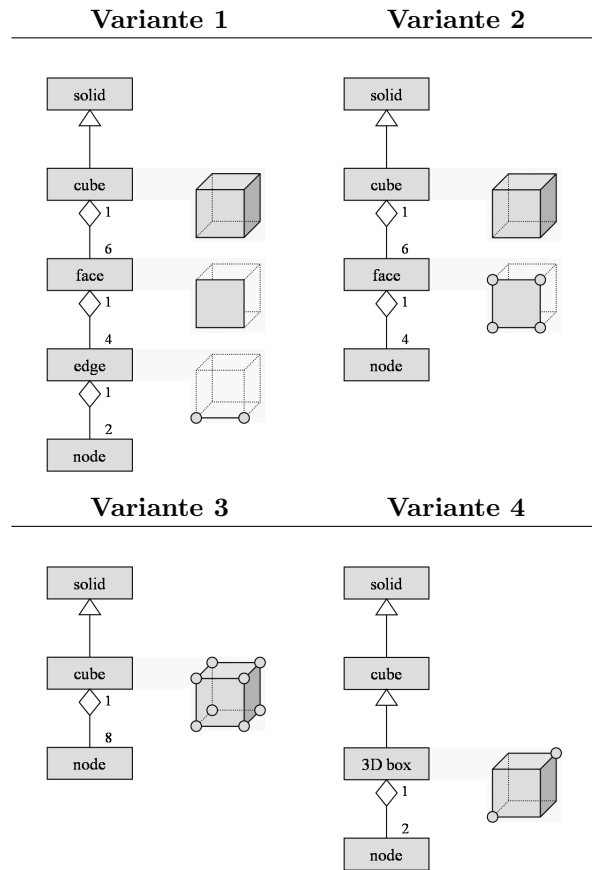
Im Hinblick auf Abschnitt 6.3, der sich mit der Nachbarschaftsanalyse eingehend auseinandersetzt, ist bereits im geometrischen Modellierungsprozess auf diese Analyse Rücksicht zu nehmen. Die Berührungen der einzelnen Würfel kann als ein Phänomen der Randelementekonstellation zwischen diesen identifiziert werden. Somit stellen die Randflächen einen wesentlichen Bestandteil dieser Modellierung dar. Umgesetzt kann diese geometrische Definition in einem *Randbeschreibungsverfahren* nach [Kad07] werden, bei dem zur Charakterisierung des Volumenkörpers seine umschließenden Flächen spezifiziert werden. Für den Fall des Würfels wird ein *Polyeder* mit sechs quadratischen Randflächen verwendet.

Da in der Geoinformatik ein Punkt den elementaren Baustein zur Beschreibung von geometrischen Objekten bildet, und Formen, wie Linien oder Flächen, sich durch die Kombination und topologische Verknüpfung von Punkten definieren lassen, wird auch hier für die Beschreibung des Polyeders ein 0-dimensionales Element im \mathfrak{R}^3 eingesetzt. Anschließend sind vier Varianten zur Definition eines Würfels angeführt, die in Verbindung mit Tabelle 3.13 zu sehen sind:

- Variante 1: Hier wird der Würfel (*cube*), der zur Gruppe der Volumenkörper (*solid*) zu zählen ist, in einem 3-stufigen Modellierungsverfahren beschrieben. Zwei Punkte (*node*) definieren eine Seitenkante (*edge*) des Würfels. Mit vier dieser Kanten wird eine Seitenfläche (*face*) aufgespannt und bei Kombination von sechs dieser Seitenflächen erhält man einen Würfel.
- Variante 2: Bei diesem Modellierungsverfahren werden zwei Schritte durchlaufen. Im ersten Schritt wird mit Hilfe von vier Punkten eine Seitenfläche des Würfels beschrieben. Verglichen mit Variante 1 werden somit zwei Schritte des Modellierungsprozesses zusammengefasst. Der zweite Schritt geschieht analog zu Variante 1, in der sechs Seitenflächen den Würfel definieren.
- Variante 3: Ist die Form des zu modellierenden 3-dimensionalen Objektes im Vorhinein bereits bekannt, kann auf ein stufenweises aufeinander aufbauendes Verfahren (*bottom-up*), wie es in den Varianten 1 und 2 passiert, verzichtet werden. Hier wird das Wissen über acht Eckpunkte eines Würfels genutzt und für die Modellierung des Objektes verwendet. Werden Seitenflächen oder Kanten für Analyseprozesse benötigt, so können diese aus der Gesamtinformation über den Würfel extrahiert werden (*top-down* Verfahren). Wie aus Tabelle 3.16 ersichtlich ist, wird in der vorliegenden Arbeit diese Variante gewählt.
- Variante 4: Ein Sonderfall der Modellierung entsteht bei der Nutzung einer *3D box* gemäß [Pos10]. Damit werden nur zwei Punkte angegeben, die eine räumliche Diagonale eines Quaders beschreiben. Der Vorteil, nur zwei Punkte definieren zu müssen, geht mit der Einschränkung, einen Quader mit fixer Orientierung im Raum – die Seitenkanten liegen parallel zu den Achsen des Koordinatensystems – nutzen zu müssen, einher.

Generell betrachtet muss die Organisation von 3-dimensionalen Daten als nicht trivial angesehen werden. Im Gegensatz dazu ist die Nomenklatur für 0-, 1- und 2-dimensionale Objekte bekannt, standardisiert und weit verbreitet. Zum einen können das proprietäre *shapefile* der Firma ESRI und zum anderen OGC konforme Beschreibungen im WKT bzw.

Tabelle 3.13: Modellierung von polyhedralen Elementen.



Well Known Binary (WKB) Format hierfür verwendet werden. Dazu passend findet man in [Ope06a, S. 15] folgende Aussage: „[...] *The instantiable subclasses of Geometry defined in this Specification are restricted to 0, 1 and 2-dimensional objects that exist in 2, 3 or 4-dimensional coordinate space (\mathfrak{R}^2 , \mathfrak{R}^3 or \mathfrak{R}^4). [...]*“. In Datenbankmanagementsystemen, wie *PostgreSQL/PostGIS*, [The10b] bzw. [Pos10], in denen ein Geometriedatentyp nach dieser OGC Spezifikation bereitsteht (vgl. [CARZ08]), sind somit Volumenobjekte meist nur über Zwischenschritte organisierbar.

[Sv02] beschreibt hierfür unter anderem die Nutzung von Polygonen im \mathfrak{R}^3 zur Definition der Seitenflächen eines Polyeders. Diese Modellierungsvariante wird auch bei [Int01] unter der Bezeichnung *GM.PolyhedralSurface* angeführt. Bei dieser Art von Modellierungsansätzen, wie sie auch aus Tabelle 3.13 in den Varianten 1 und 2 ersichtlich ist, wird ein und dasselbe Element öfters benötigt: Im Fall des Würfels in Variante 1 wird jeder Punkt dreimal und jede Kante zweimal definiert. In Variante 2 werden durch die verkürzte Modellierung nur die Punkte in dreifacher Weise benötigt. In der Aufhebung dieser redundanten Information über Punkte und Kanten ist laut [ASv05] ein Einsparungspotential hinsichtlich

des benötigten Speicherplatzes zu sehen.

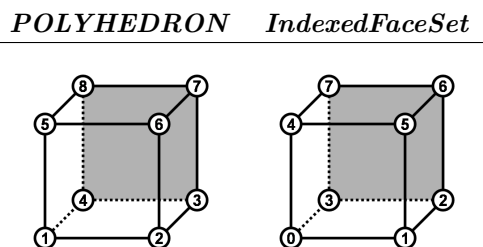
Tabelle 3.14: Gegenüberstellung von Nomenklaturen zu Beschreibung eines Würfels.

POLYHEDRON
POLYHEDRON(PolygonInfo(6, 24), SumVertexList(8), SumPolygonList(4, 4, <u>4</u> , 4, 4, 4), VertexList(0, 0, 0, 1, 0, 0, <u>1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1</u>), PolygonList(1, 2, 6, 5, 2, 3, 7, 6, <u>3, 4, 8, 7</u> , 4, 1, 5, 8, 1, 2, 3, 4, 5, 6, 7, 8))
IndexedFaceSet
<IndexedFaceSet coordIndex=0 1 5 4 -1 1 2 6 5 -1 <u>2 3 7 6 -1</u> 3 0 4 7 -1 0 1 2 3 -1 4 5 6 7 -1> <Coordinate point='0 0 0,1 0 0, <u>1 1 0,0 1 0</u> ,0 0 1,1 0 1, <u>1 1 1,0 1 1</u> ' /> < /IndexedFaceSet>

Der Datentyp *POLYHEDRON* wird als Konsequenz daraus bei [CARZ08] eingeführt. Innerhalb dieses Datentyps werden die einzelnen Eckpunkte koordinativ aufgelistet (*VertexList*), die aufzuspannenden Flächen durch eine Abfolge von Indizes der Eckpunkte beschrieben (*PolygonList*) und Informationen über die gesamte Menge an Eckpunkten (*SumVertexList*), die Anzahl der Eckpunkte pro Polygon (*SumPolygonList*) und die Anzahl der Flächen bzw. die Menge der für die Flächendefinition notwendigen Eckpunkte (*PolygonInfo*) angeführt.

Diese Nomenklatur kann vom Prinzip her mit jener Definition eines *IndexedFaceSet* Knotens aus der Beschreibungssprache extensible 3D (X3D) gemäß [Int08] verglichen werden. Auch hier werden die Eckpunkte für alle zu definierenden Flächen gemeinsam aufgelistet (*Coordinate point*) und anschließend nur über den Index der Eckpunkte die jeweiligen Flächen gebildet (*coordIndex*). In Tabelle 3.14 werden diese beiden Nomenklaturen am Beispiel eines Einheitswürfels ($s = 1$) und der Position eines Eckpunktes im Koordinatensprung gegenübergestellt.

Tabelle 3.15: Einheitswürfel zur Gegenüberstellung der Nomenklaturen.



Ergänzend zu der angeführten Syntax sind die damit beschriebenen Einheitswürfel in Tabelle 3.15 abgebildet. Die beiden Würfel unterscheiden sich in den Indizes der einzelnen

Eckpunkte. Wird im Fall des *POLYHEDRON* der erste Punkt mit dem Wert *1* indiziert, so beginnt man bei einem *IndexedFaceSet* mit *0* zu zählen. Für einen besseren Vergleich der beiden Nomenklaturen wird eine Seitenfläche des Würfels gesondert betrachtet. In Tabelle 3.15 ist diese Fläche grau gefüllt und in Tabelle 3.14 unterstrichen dargestellt. Abgesehen von der Indizierung des ersten Eckpunktes unterscheiden sich die beiden Nomenklaturen innerhalb der Seitenflächendefinition. Die im Vorfeld bereits spezifizierte Anzahl benötigter Eckpunkte pro Seitenfläche bei einem *POLYHEDRON* (*SumPolygonList*), lässt eine durchgehende Aneinanderreihung der Eckpunkte in *PolygonList* zu. Bei einer *IndexedFaceSet* Definition, wird hierfür im *coordIndex* ein Sonderzeichen für die Aufteilung der einzelnen Seitenflächen notwendig. Diese Funktion übernimmt der Indexwert *-1*.

Die Tabelle 3.16 zeigt den Aufbau der Basisdatenrelation, die in der weiteren Arbeit genutzt wird. Bis auf die geometrische Eigenschaft dieser Art von Basiselement ist der Aufbau mit jener der Knoten-Kanten, Kanten-Traversen und polygonalen Strukturen ident. Da für die Modellierung jene Variante mit den acht Eckpunkten gewählt wurde, finden sich diese als acht separat angeführte Attribute im Relationsschema wieder: *1. Eckpunkt* bis *8. Eckpunkt*. Verglichen mit den bisher verwendeten Geometriebeschreibungen wird hier zusätzlich die dritte Koordinatenkomponente in der OGC konformen WKT Beschreibung verwendet (z_{c_i}). Vor dem Hintergrund der eingesetzten Geodateninfrastruktur – ein *PostgreSQL/PostGIS Datenbankmanagementsystem* – und dem Ziel der Arbeit, sich auf konzeptioneller Ebene mit der Thematik der Routenberechnung auseinanderzusetzen, aber nicht neue Datentypen in bestehenden Datenbanken zu implementieren, wurde diese geometrische Modellierungsvariante für 3-dimensionale Objekte gewählt.

Tabelle 3.16: Relationale Darstellung der Eigenschaften von Basiselementen in einer polyhedralen Struktur unter der Verwendung von Eckpunkten.

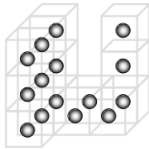

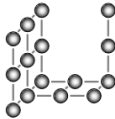
Id	[PS]	c_i
Kosten		c_{c_i}
Labelwert		l_{c_i}
Vorgänger	[FS]	p_{c_i}
Geometrie	1. Eckpunkt	POINT($x_1 c_i y_1 c_i z_1 c_i$)
	2. Eckpunkt	POINT($x_2 c_i y_2 c_i z_2 c_i$)
	3. Eckpunkt	POINT($x_3 c_i y_3 c_i z_3 c_i$)
	4. Eckpunkt	POINT($x_4 c_i y_4 c_i z_4 c_i$)
	5. Eckpunkt	POINT($x_5 c_i y_5 c_i z_5 c_i$)
	6. Eckpunkt	POINT($x_6 c_i y_6 c_i z_6 c_i$)
	7. Eckpunkt	POINT($x_7 c_i y_7 c_i z_7 c_i$)
	8. Eckpunkt	POINT($x_8 c_i y_8 c_i z_8 c_i$)

Im Vergleich mit polygonalen Strukturen, bei denen die Nachbarschaftsbeziehung ebenfalls implizit über Berührungen von Basiselementen geregelt wird, besteht bei polyhedralen Strukturen eine bidirektionale Nachbarschaftsbeziehung sobald sich zwei Würfel berühren. Im Verlauf der Nachbarschaftsanalyse in Abschnitt 6.3 wird auf eine richtungsabhängige Suchbaumausbreitung eingegangen, die in einem thematischen Nahverhältnis zu gerichteten Verbindungselementen steht. Abgesehen davon können gerichtete Nachbarschaften nur durch Hinzunahme externer Informationen während der Suchbaumausbreitung berücksich-

tigt werden.

Der Einsatz einer dualen Repräsentationsform, wie sie bereits als Regionsadjazenzgraph bei den polygonalen Modellen beschrieben wird, kann als alternativer Modellierungsansatz gewählt werden (z.B.: [BG09] oder [AP08, Kapitel 4]). Der hierfür notwendige Transformationsprozess der Basiselemente in ihre duale Repräsentationsform – vom Würfel c zum Knoten n – und die Schaffung von Verbindungselementen in Form von Kanten a lässt sich mit Hilfe einer *Poincaré Dualität* (vgl. [Mun95]) beschreiben. [Lee04] differenziert bei solchen Transformationen zwischen den Vorgängen im Euklidischen Raum und im topologischen Raum: Wird die Überführung im ersten Fall als *Skelettierung* beschrieben, so wird die Bezeichnung *Poincaré Dualität* für den zweiten Fall eingeführt. Im Bereich der *Indoor Navigation* setzt beispielsweise [BNK09] diese Skelettierungsverfahren und Poincaré Dualitäten zur Umbildung von Räumen innerhalb von Gebäuden ein.

Tabelle 3.17: Basis- und Verbindungselement in einer Knoten-Kanten Struktur abgeleitet aus einer polyhedralen Struktur.

Basiselement	Verbindungselement	Kombinierte Darstellung
		

In Tabelle 3.17 sind polyhedrale Strukturen in ihrer dualen Darstellung – einer Knoten-Kanten Struktur – angeführt. Das Basiselement wird zu einem Knoten transformiert und in der Mitte des korrespondierenden Würfels platziert. Die Erzeugung des Verbindungselementes beruht auf einer Analyse aller berührenden Würfel. In dieser Darstellung werden nur jene Kanten gezeigt, die auf eine Berührung von Seitenflächen zurückzuführen sind. In Abschnitt 6.3 werden zusätzlich Nachbarschaften definiert, die sich auf Berührungen von Kanten oder einzelnen Eckpunkten des Würfels beziehen.

3.6 Gemischt-komplexe Strukturen

Eine Kombination von Basis- und Verbindungselementen aus unterschiedlichen geometrischen Repräsentationsformen, wie sie bereit in Abschnitt 3.3 beschreiben wird, kann so gestaltet werden, dass polygonale und polyhedrale Daten gemeinsam mit den gemischt-einfachen Strukturen für eine Suchbaumausbreitung genutzt werden können. Da hierbei einerseits Daten mit einer expliziten Nachbarschaftsbeschreibung und andererseits Daten, deren Nachbarn implizit über Berührung erkannt werden, zusammenzufassen sind, ist im Algorithmus darauf Rücksicht zu nehmen: Sobald ein Übergang von expliziter auf impliziter Nachbarschaftsbeschreibung erfolgt, muss der Algorithmus von der Nutzung der Verbindungselemente auf eine berührungsbasierte Analyse umstellen. Der umgekehrte Fall tritt bei einem Übergang von impliziter auf expliziter Nachbarschaftsbeschreibung auf.

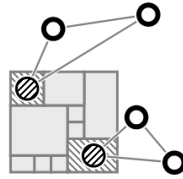
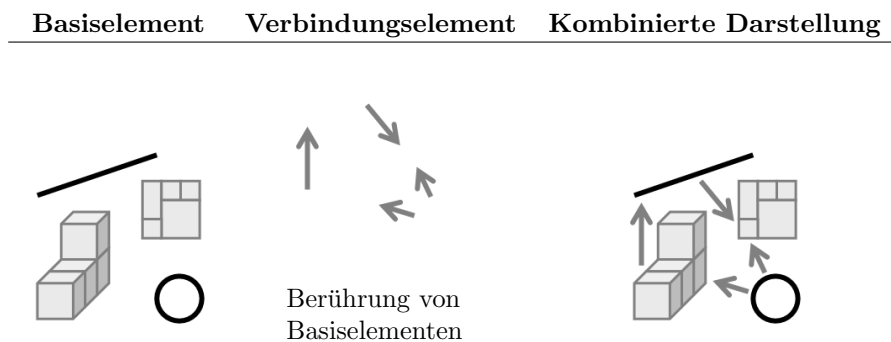


Abbildung 3.1: Verbindungselement zwischen expliziten und impliziten Datenmodellen.

Zur Kenntlichmachung dieser besonderen Stelle innerhalb der Basisdaten, sind die involvierten Basiselemente mit einem zusätzlichen Attributwert zu versehen. In Abbildung 3.1 werden jene Basiselemente einer Knoten-Kanten Struktur und einer polygonalen Struktur schraffiert dargestellt, an denen ein Wechsel zwischen den unterschiedlichen Basisdatenarten möglich ist. Diese Sonderform eines Basiselementes kann als Road Transition Element (RTE) bezeichnet werden. Das Anwendungsgebiet eines RTE ist nicht eingeschränkt auf die abgebildete Kombination von Knoten-Kanten und polygonalen Strukturen zu sehen, sondern auf alle geometrischen Repräsentationsformen erweiterbar.

In Tabelle 3.18 sind alle Arten von Basisdatentypen – Knoten, Kanten, Polygone und Polyeder – und die unterschiedlichen Nachbarschaftsbeschreibungsverfahren angeführt. Durch die kombinierte Nutzung verschiedener Modellierungsverfahren, angepasst auf die jeweilige Aufgabenstellung, sind Analyseprozesse vor dem Hintergrund der Routenberechnung in einem Guss durchführbar.

Tabelle 3.18: Basis- und Verbindungselement in einer gemischt-komplexen Struktur.



Werden polygonale und polyhedrale Modelle durch ihre jeweilige duale Repräsentationsform verwaltet, so ist bei einer kombinierten Nutzung die Herangehensweise mit jener der gemischt-einfachen Strukturen aus Abschnitt 3.3 gleichzusetzen. Die zusätzliche Spezifikation eines RTE und eine Erweiterung des Algorithmus hinsichtlich des Wechsels zwischen impliziter und expliziter Nachbarschaftsbeschreibung entfällt somit.

Kapitel 4

Beschreibung zeitabhängiger Vorgänge

Die Frage *Wie groß ist Österreich?* soll als Einstieg in dieses Kapitel dienen. Bezogen auf den aktuellen Zeitpunkt lässt sich diese Frage leicht beantworten – man sucht auf Wikipedia danach und erhält umgehend eine Antwort: 83.871 km² [Wik10]. Erweitert man jedoch die anfangs gestellte Frage auf *Wie groß war Österreich im 20. Jahrhundert?* fällt die Antwort mehrdeutig aus. Die Semantik *Österreich* kann nicht wie zuvor eindeutig einer Geometrie zugeordnet werden, aus der die Größe abzuleiten ist, sondern je nach Zeitpunkt zu dem man Österreich betrachtet – Österreich-Ungarn, Deutsch-Österreich, Erste Republik, Anschluss ans Dritte Reich oder Zweite Republik (vgl. [KM08a] od. [KM08b]) – kann sich seine geometrische Form und die daraus abgeleitete Größe unterscheiden. Anm.: Laut Bundesamt für Eich- und Vermessungswesen, [Dun10] ist bei einer Größenfeststellung des Staatsgebietes der Einfluss *beweglicher Grenzen*, wie beispielsweise die Flussmitte, zu berücksichtigen. Somit stellt nicht nur die Raumkoordinate sondern auch die Zeitkoordinate einer Semantik einen wesentlichen Bestandteil dar, um diese Frage richtig beantworten zu können!

Der Bezug zur Thematik zwischen zeitabhängigen Datenmodellen und der Routenplanung wird in vielen Facetten widerspiegelt: Einen wesentlichen Punkt stellt die Aktualität jener Geodaten dar, die als Grundlage der Routenplanung dienen. Neubauten, Ausbauten und Rückbauten der Verkehrsinfrastruktur sind hierbei die zeitkritischen Elemente. Als ein weiterer wichtiger Punkt ist die Fahrzeugklasse zu sehen, für die eine Route zu planen ist. Laut österreichischer Straßenverkehrsordnung sind Lastkraftwagen ab bestimmten Gewichtsklassen einem Nachtfahrverbot (§42 Abs. 6), einem Wochenendfahrverbot (§42 Abs. 1.) oder einem Ferienfahrverbot (§42 Abs. 5) unterworfen. Unabhängig von der Fahrzeugklasse wirken sich saisonale Sperrungen, wie etwa die Wintersperre der Glockner Hochalpenstraße [Gro10], als einschränkender Faktor auf die Datengrundlage aus. Auch der Fahrer selbst beeinflusst durch seine Fahr- und Ruhezeiten nach §14a des österreichischen Arbeitszeitgesetzes die Routenplanung. Für eine optimale Lösung eines SPP sind zumindest all die angeführten Faktoren in Einklang zu bringen, wobei die Zeit hierfür als ebenbürtige Objekteigenschaft zur Geometrie zu sehen ist.

Dieses Kapitel ist als Ergänzung zum vorangegangenen Kapitel 3 zu sehen. Hier werden die Modellierungsverfahren – fokussiert auf Knoten-Kanten und Kanten-Traversen Strukturen aus den Abschnitten 3.1 und 3.2 – zur Beschreibung zeitabhängiger Vorgänge unter

der Prämisse weitergeführt, keine speziellen Effekte aus der Relativitätstheorie, [EI56, S. 120-131] zu berücksichtigen. Die Zeit wird hier aus der Sicht der klassischen Mechanik betrachtet. In diesen geschaffenen Datenmodellen können in weiterer Folge die Algorithmen aus Kapitel 5 ihre Suchbäume ausbreiten und kürzeste Wege finden.

Im nachfolgenden Abschnitt 4.1 werden grundlegende Elemente – Zeitpunkt und Zeitspanne – für eine Modellierung zeitabhängiger Vorgänge eingeführt und deren Beziehungsvarianten zueinander aufgeschlüsselt. Darauf aufbauend lassen sich logische Analysefunktionen formulieren. Der Abschnitt 4.2 befasst sich mit der Darstellung von zeitabhängigen Vorgängen. Diese rein graphischen Darstellungsvarianten sind als Ideengeber und Vorstufe zu einem geometrisch/topologischen Modellierungsprozess zu verstehen, der im Anschluss in Abschnitt 4.3 speziell für den Fahrplanverkehr erarbeitet wird.

4.1 Beziehung zwischen zeitabhängigen Vorgängen

Die Formulierung unterschiedlichster Beziehungen zwischen geometrischen Objekten ist in der Literatur bereits intensiv diskutiert und finden praktische Anwendung in den Analysefunktionen diverser Geoinformationssysteme. [ESM93] beschreibt unter anderem Beziehungstypen von 0-, 1- und 2-dimensionalen Geometrieobjekten im \mathfrak{R}^1 und \mathfrak{R}^2 . Beziehungen zwischen 3-dimensionalen Objekten wird zu dieser Thematik ergänzend in [Zla00] erörtert. Das In-Beziehungbringen zeitlicher Ereignisse kann als geometrische Analyse aufgefasst werden, worin Zeitpunkt zu Zeitpunkt, Zeitpunkt zu Zeitspanne und Zeitspanne zu Zeitspanne im Sinne von [Int02] zueinander betrachtet werden (vgl. Abschnitte 4.1.2, 4.1.3 und 4.1.4).

4.1.1 Definitionen

Die angestrebte geometrische Analyse zeitabhängiger Vorgänge bedarf einer geometrischen Beschreibung dieser Vorgänge. Die Definition eines Punktes oder einer gerichteten Linie – der Anfang der Linie muss zeitlich betrachtet vor dem Ende der Linie liegen – im \mathfrak{R}^1 bildet hierfür das entsprechende geometrische Pendant zu einem Zeitpunkt oder einer Zeitspanne (vgl. [Bar05]). Wiederkehrende Ereignisse, seien sie einem festen Zyklus unterworfen oder treten sie in unregelmäßigen zeitlichen Abständen erneut auf, können beispielsweise als *MULTIPOINT* für wiederkehrende Zeitpunkte, *MULTILINESTRING* für wiederkehrende Zeitspannen oder *GEOMETRYCOLLECTION* für eine wiederkehrende Mischung aus Zeitpunkten und Zeitspannen nach [Ope06a] beschrieben werden. Im Gegensatz zu den zuvor angesprochenen zeitlichen Grundphänomenen und deren Kombinationen daraus, deren Anfang und Ende definierbar ist, wird eine kontinuierliche Linie zur Repräsentanz zeitlich unveränderbarer Objekte benötigt.

Vergleichbar zur Festlegung eines Lage- und Höhenreferenzsystems bei 2- und 3-dimensionalen räumlichen Daten, wie Spatial Reference Identifier (SRID) zum Beispiel gemäß European Petrol Survey Group (EPSG) bzw. International Association of Oil and Gas Producers (OGP) *Survey and Positioning Committee* [OGP10], ist für zeitliche Daten genauso eine Zeitreferenz zu benennen, der die Zeitkoordinaten entsprechen (vgl. [Int02, Abschnitt 5.3]). Beispielsweise findet 12:00 Mittag, bei Vernachlässigen der zeitlichen Quantifizierung in Zeitzonen, auf der Erde kontinuierlich statt und lässt somit eine Eindeutigkeit der Zeitkoordinate nicht zu. Bei [HLC01, Abschnitt 3.3] werden hierzu die wichtigsten Referenzen, wie beispielsweise die Universal Time (UT), die International Atomic Time (IAT), die Universal Time Coordinated (UTC) und die Global Positioning System Time (GPST) beschrieben, die mit

navigatorischen Aufgaben in Verbindung gebracht werden können. Die zuvor verwendeten Begriffe *Zeitreferenz* und *Zeitkoordinate* sind nicht auf die Uhrzeit alleine zu beziehen, sondern gemeinsam mit dem Datum – Tag, Monat, Jahr – zu betrachten. Auch hierzu wird bei [HLC01, Abschnitt 3.3] auf das Julianische Datum JD, das modifizierte Julianische Datum MJD und die Global Positioning System Standard Epoch (GPSSE) wegen ihrer Affinität zur Navigation hingewiesen. Diese Datumsangaben – JD, MJD oder GPSSE – sind nicht wie klassische Datumsangaben nach [Int04] (z.B.: 1981-05-28) zu verstehen, sondern stellen die Summe aller Tage seit einem definierten Ausgangszeitpunkt dar. Im Falle des JD handelt es sich dabei um das Zusammenfallen des 28-jährigen Sonnenzyklus, des 19-jährigen Metonischen Zyklus und des 15-jährigen Indiktionszyklus am 1. Jänner 4713 v.Chr. um 12:00 Uhr. Der 28. Mai 1981 wäre in diesem Datumsformat bis zur Mittagsstunde der Tag Nr. 2444752 und danach der Tag Nr. 2444753. Eine Uhrzeitinformation geht in solchen Zeitformaten als Bruchteil der Tagzahl ein: 00:00 Uhr am 28. Mai 1981 entspricht somit dem Wert 2444752.5.

Diese durchgehende Nummerierung von Zeiteinheiten in einem Dezimalsystem kann für eine geometrische Modellierung zeitabhängiger Vorgänge eingesetzt werden. Nachfolgend sind für 12:00 am 28. Mai 1981, für das 20. Jahrhundert und für die EU-Ratspräsidentschaft Österreichs in den Jahren 1998 (Jänner bis Juni) und 2006 (Juli bis Dezember) jeweils eine an die OGC-Spezifikationen angelehnte Geometriebeschreibung im \mathfrak{R}^1 nach dem JD angeführt:

- POINT(2444753.0)
- LINESTRING(2415385.5, 2451910.5)
- MULTILINESTRING((2450814.5, 2450995.5), (2453736.5, 2453917.5))

4.1.2 Beziehungen zwischen zwei Zeitpunkten

In Tabelle 4.1 werden die topologischen Beziehungen nach [Int02] zwischen zwei Zeitpunkten – hier als A und B bezeichnet – aufgelistet. Es wird hierbei eine ungleichzeitige, als vorzeitig oder nachzeitig betrachtbar, und eine gleichzeitige Situation angeführt. [Int02, Abschnitt 5.2.3.5a] bezeichnet diese als *Before*, *Equals* und *After*.

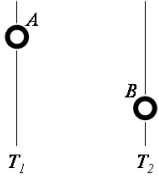

In Tabelle 4.1, wie auch in den anschließenden Tabellen 4.2 und 4.3, handelt es sich bei den Graphiken um ein 1-dimensionales Diagramm, bestehend aus einer Zeitachse mit den diskreten Zeitpunkten T_i entlang der Abszisse. Die Verschiebung von Zeitpunkten und Zeitspannen entlang der Ordinate dient ausschließlich einer verständlicheren visuellen Wahrnehmung durch den Betrachter.

4.1.3 Beziehung zwischen Zeitpunkt und Zeitspanne

Die einzelnen topologischen Beziehungen zwischen einem Zeitpunkt A und einer Zeitspanne b werden in Tabelle 4.2 angeführt. Im Vergleich zu den vorangegangenen Beziehungen zwischen zwei Zeitpunkten aus Abschnitt 4.1.2 kann hier bei der zeitgleichen Beziehung in *zu Beginn*, *während* und *am Ende* differenziert werden. Diese Untergliederung lässt bei temporären Analysen eine schärfere Formulierung der Analysefunktion oder des Analyseergebnisses zu.

Zusätzlich sind unterschiedliche Topologiebeschreibungen für ein und dieselbe Situation zu beachten, je nachdem welches der Zeitelemente – Zeitpunkt oder Zeitspanne – als Subjekt dieser Beschreibung gesehen wird (vgl. [Int02, Abschnitt 5.2.3.5b/c]): *Before*, *Begins* für Zeitpunkt zu Zeitspanne bzw. *BegunBy* für Zeitspanne zu Zeitpunkt, *During* für Zeitpunkt zu

Tabelle 4.1: Beziehung zwischen zwei Zeitpunkten.

Graphik	Beschreibung
	<p>Zeitpunkt A, $POINT(T_1)$, vor Zeitpunkt B, $POINT(T_2)$. Entspricht:</p> <ul style="list-style-type: none"> • A Before B. • B After A.
	<p>Gleicher Zeitpunkt A, $POINT(T_1)$, und B, $POINT(T_1)$. Entspricht:</p> <ul style="list-style-type: none"> • A Equals B.

Zeitspanne bzw. *Contains* für Zeitspanne zu Zeitpunkt, *Ends* für Zeitpunkt zu Zeitspanne bzw. *EndedBy* für Zeitspanne zu Zeitpunkt und *After*. Anm.: Diese subjektbezogene Beschreibung tritt auch bei den zuvor beschriebenen Beziehungen zwischen zwei Zeitpunkten auf, wird aber hier deutlicher wahrgenommen!

4.1.4 Beziehung zwischen zwei Zeitspannen

Die letzte Gruppe der Beschreibungen topologischer Beziehungen zeitabhängiger Vorgänge umfasst jene der zwei Zeitspannen a und b . Die Steigerung der Komplexität einer Beschreibung, wie sie von Abschnitt 4.1.2 zu Abschnitt 4.1.3 erfolgte, wird hier im Vergleich zum vorangegangenen Abschnitt über die Beziehung zwischen Zeitpunkt und Zeitspanne weiter fortgesetzt.

Bringt man zwei Zeitspannen miteinander in Beziehung, so können zusätzlich Überlappungseffekte und Aneinanderreihungen entstehen, die mit den bisherigen Kombinationen der Zeitelemente – Zeitpunkt zu Zeitpunkt oder Zeitpunkt zu Zeitspanne – nicht in Erscheinung treten konnten. In Tabelle 4.3 sind hierzu die einzelnen Beziehungen zwischen zwei Zeitspannen aufgelistet. Diese Einteilung erfolgte dabei nach [Int02, Abschnitt 5.2.3.5d]: *Before*, *Meets* bzw. *MetBy*, *Overlaps* bzw. *OverlappedBy*, *Begins* bzw. *BegunBy*, *During* bzw. *Contains*, *Equals*, *Ends* bzw. *EndedBy* und *After*. Auch hier, rückblickend auf Abschnitt 4.1.3, ist die Formulierung der Beziehung vom jeweiligen Subjekt abhängig.

Tabelle 4.2: Beziehung zwischen einem Zeitpunkt und einer Zeitspanne.

Graphik	Beschreibung
<p>The diagram shows three vertical lines representing time points T_1, T_2, and T_3. A point A is marked on the T_1 line. A horizontal line segment b connects the T_2 and T_3 lines.</p>	<p>Zeitpunkt A, $POINT(T_1)$, vor Zeitspanne b, $LINESTRING(T_2, T_3)$. Entspricht:</p> <ul style="list-style-type: none"> • A Before b. • b After A.
<p>The diagram shows three vertical lines representing time points T_1, T_2, and T_3. A point A is marked on the T_1 line. A horizontal line segment b connects the T_1 and T_2 lines.</p>	<p>Zeitpunkt A, $POINT(T_1)$, zu Beginn der Zeitspanne b, $LINESTRING(T_1, T_2)$. Entspricht:</p> <ul style="list-style-type: none"> • A Begins b. • b BegunBy A.
<p>The diagram shows three vertical lines representing time points T_1, T_2, and T_3. A point A is marked on the T_2 line. A horizontal line segment b connects the T_1 and T_3 lines.</p>	<p>Zeitpunkt A, $POINT(T_2)$, innerhalb der Zeitspanne b, $LINESTRING(T_1, T_3)$. Entspricht:</p> <ul style="list-style-type: none"> • A During b. • b Contains A.
<p>The diagram shows three vertical lines representing time points T_1, T_2, and T_3. A point A is marked on the T_2 line. A horizontal line segment b connects the T_1 and T_2 lines.</p>	<p>Zeitpunkt A, $POINT(T_2)$, am Ende der Zeitspanne b, $LINESTRING(T_1, T_2)$. Entspricht:</p> <ul style="list-style-type: none"> • A Ends b. • b EndedBy A.
<p>The diagram shows three vertical lines representing time points T_1, T_2, and T_3. A point A is marked on the T_3 line. A horizontal line segment b connects the T_1 and T_2 lines.</p>	<p>Zeitpunkt A, $POINT(T_3)$, nach der Zeitspanne b, $LINESTRING(T_1, T_2)$. Entspricht:</p> <ul style="list-style-type: none"> • A After b. • b Before A.

Tabelle 4.3: Beziehung zwischen zwei Zeitspannen

Graphik	Beschreibung
	<p>Zeitspanne a, $LINESTRING(T_1, T_2)$, vor Zeitspanne b, $LINESTRING(T_3, T_4)$. Entspricht:</p> <ul style="list-style-type: none"> • a Before b. • b After a.
	<p>Zeitspanne b, $LINESTRING(T_2, T_3)$, schließt an Zeitspanne a, $LINESTRING(T_1, T_2)$, an. Entspricht:</p> <ul style="list-style-type: none"> • a Meets b. • b MetBy a.
	<p>Zeitspanne a, $LINESTRING(T_1, T_3)$, beginnt mit Zeitspanne b, $LINESTRING(T_1, T_2)$. Entspricht:</p> <ul style="list-style-type: none"> • a BegunBy b. • b Begins a.
	<p>Zeitspanne a, $LINESTRING(T_1, T_4)$, umschließt Zeitspanne b, $LINESTRING(T_2, T_3)$. Entspricht:</p> <ul style="list-style-type: none"> • a Contains b. • b During a.
	<p>Zeitspanne a, $LINESTRING(T_1, T_2)$, entspricht Zeitspanne b, $LINESTRING(T_1, T_2)$. Entspricht:</p> <ul style="list-style-type: none"> • a Equals b.

Fortsetzung auf nächster Seite

Graphik	Beschreibung
	<p>Zeitspanne a, $LINESTRING(T_1, T_3)$, endet mit Zeitspanne b, $LINESTRING(T_2, T_3)$. Entspricht:</p> <ul style="list-style-type: none"> • a EndedBy b. • b Ends a.
	<p>Zeitspanne a, $LINESTRING(T_1, T_3)$, überlappt mit Zeitspanne b, $LINESTRING(T_2, T_4)$. Entspricht:</p> <ul style="list-style-type: none"> • a Overlaps b. • b OverlappedBy a.

4.2 Abbildung zeitabhängiger Vorgänge

So wie Karten und Pläne eine adäquate Variante zur Abbildung räumlicher Gegebenheiten darstellen und eine Beschleunigung der Informationsaufnahme der teilweise komplexen Inhalte dem Betrachter ermöglichen, können zeitabhängige Vorgänge genauso graphisch vermittelt werden. Die Auseinandersetzung mit der graphischen Gestaltung solcher Vorgänge dient hier als Ideengeber und Vorstufe zu den in Abschnitt 4.3 beschriebenen Modellierungsverfahren.

Die Komplexität der Darstellungen – jeweils als Diagramm mit einer Zeitachse – wird in den nachfolgenden zwei Abschnitten 4.2.1 und 4.2.2 gesteigert: Von der Visualisierung eines einzelnen Themas – geringe Komplexität der Visualisierung – bis hin zu einer 2-dimensionalen räumlichen Information – hohe Komplexität der Visualisierung.

4.2.1 Thema/Zeit Diagramme

Die Darstellung eines bestimmten Themas bezogen auf seine zeitabhängige Variation vereinfacht im Gegensatz zu einer rein verbalen Beschreibung des Sachverhaltes die Informationsaufnahme durch den Betrachter. Eine geläufige Diagrammform besteht aus einer Abszisse, entlang der die einzelnen Zeitpunkte aufgetragen werden, und einer Ordinate, die in verschieden viele Themen unterteilt wird. Diese Einteilung der Diagrammachsen kann man beispielsweise bei Seismogrammen, Elektrokardiogrammen oder Elektroenzephalogrammen genauso feststellen.

Ein Beispiel aus [BK09, S. 331], Abbildung 4.1, zeigt ein solches Diagramm. Hier sind die Zeiteinheiten entlang der Abszisse als *Lebensalter* aufgetragen, wobei einzelne Zeitpunkte und Zeitspannen graphisch hervorgehoben werden. Die Zeitspannen können in dieser speziellen Graphik durch ihre gefeilte Ausformung vor dem Hintergrund der *Fuzzy-Logik* nach [Bar05] betrachtet werden. Die Ordinate umfasst zwei Themen: Die *Bruttokapitalanlage* und die *Bruttokapitalaufnahme*.

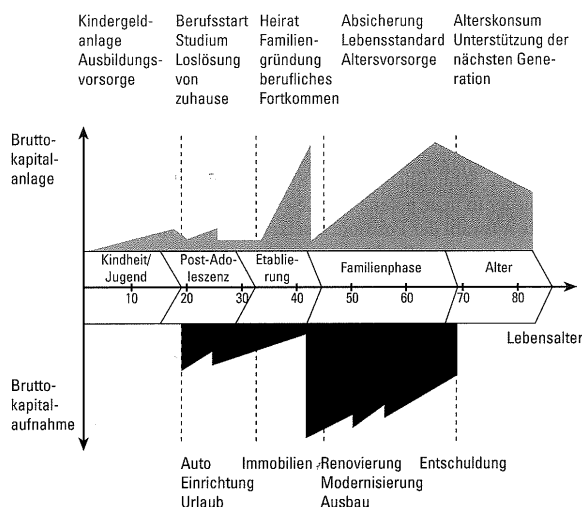


Abbildung 4.1: Zeitabhängige Darstellung von Finanz- und Anlagebedarf in einem 2-dimensionalen Thema/Zeit Diagramm. Quelle: [BK09, S. 331].

Den Übergang zum nachfolgenden Abschnitt 4.2.2 bilden beispielsweise diagrammetrische Abbildungen aus dem Themenbereich der Beschleunigungsintegration: Beginnend bei Beschleunigung/Zeit Diagrammen, über Geschwindigkeit/Zeit Diagrammen bis hin zu Weg/Zeit Diagramme, wobei der Weg bereits als räumliche Repräsentanz verstanden werden kann.

4.2.2 Raum/Zeit Diagramme

Die Auseinandersetzung mit Raum/Zeit Diagrammen ist hier als letzte Vorstufe zum eigentlichen Modellierungsprozess zeitabhängiger Vorgänge in Abschnitt 4.3 zu sehen. Je nach Komplexität der Rauminformation besitzen die Raum/Zeit Diagramme eine unterschiedliche graphische Ausprägung. Bei 1-dimensionaler Rauminformation, wie im vorherigen Abschnitt 4.2.1 durch das Weg/Zeit Diagramm angesprochen, wird die Zeit entlang der Abszisse aufgetragen und der Raumanteil des Diagrammes entlang der Ordinate. Hier können beispielsweise der zurückgelegte Weg in einer Längeneinheit oder spezielle Ortsbezeichnungen als Quantifizierung der Ordinate dienen. Ein Beispiel, Abbildung 4.2, für ein 2-dimensionales Raum/Zeit Diagramm findet sich in [EI56, S. 139]. Hierbei wurde sogar die Ordinate als *Raumachse* beschriftet!

Für eine diagrammetrische Darstellung zeitabhängiger 2-dimensionaler Rauminformation kann ein 3-dimensionales Diagramm mit zwei Raumachsen und einer Zeitachse verwendet werden. Hierbei wechselt die Zeitachse von der Abszisse bei der vorherigen Diagrammvariante mit 1-dimensionaler Rauminformation hin zur Applikate. Abszisse und Ordinate sind somit der 2-dimensionalen Rauminformation vorbehalten. Ein Beispiel für ein 3-dimensionales Raum/Zeit Diagramm ist in [Bar08, S. 112] zu finden (vgl. Abbildung 4.3). Hier stellt jede Schicht mit konstantem Applikatenwert die Raumsituation zu einem bestimmten Zeitpunkt dar. Speziell in dieser Abbildung über die Entwicklung des Universums ist der 2-dimensionale Raum nur als Stellvertreter aller Raumdimensionen zu verstehen.

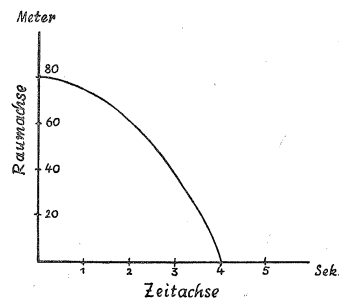


Abbildung 4.2: Zeitabhängige Darstellung eines Fallexperimentes in einem 2-dimensionalen Raum/Zeit Diagramm. Quelle: [EI56, S. 139].

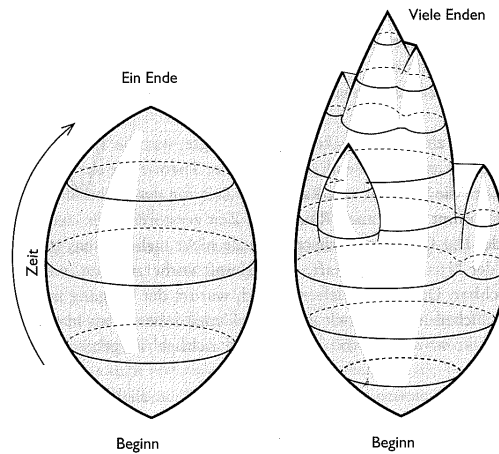


Abbildung 4.3: Zeitabhängige Darstellung über die Entwicklung des Universums in einem 3-dimensionalen Raum/Zeit Diagramm. Quelle: [Bar08, S. 112].

Diese letzte Abbildung ist als direkte Vorstufe zur geometrischen Modellierung von zeitabhängigen Vorgängen zu sehen. Für das angestrebte Ziel, Fahrplandaten zu modellieren und für die Suchbaumausbreitung zu nutzen, ist die Handhabung 1- (z.B.: bei Haltestellenbezeichnung/Zeit Diagrammen) bzw. 2-dimensionaler räumlicher Information (z.B.: bei X- und Y-Koordinate der Haltestelle/Zeit Diagrammen) ausreichend.

4.3 Modellierung zeitabhängiger Vorgänge

Basierend auf den Erkenntnissen aus den vorangegangenen Abschnitten dieses Kapitels werden hier speziell für die kürzeste Wegesuche in Fahrplänen und für eine kombinierte Wegesuche in Fahrplänen und zeitunabhängigen Daten verschiedene Haltestellensituationen modelliert. In Abbildung 4.4 wird zu diesem Thema passend der Ausschnitt eines Bildfahrplanes dargestellt, der als unmittelbare Vorstufe zum geometrisch/topologischen Modellierungsprozess zu sehen ist. In dieser Darstellung von [Ruc10] ist im Gegensatz zu den bisher beschrie-

benen 2-dimensionalen Raum/Zeit Diagrammen die Zeit entlang der Ordinate – 10-minütige Einteilung zwischen 11:00 Uhr und 12:00 – und die Rauminformation entlang der Abszisse – ausgeschriebene und abgekürzte Stationsnamen – aufgetragen, was jedoch speziell für Bildfahrpläne als Usus zu werten ist. Die schräg verlaufenden Linien beschreiben die Zugverbindungen. Somit kann abgelesen werden zu welcher Zeit an welchem Ort ein bestimmter Zug verkehrt. Die Identifikation des Zuges erfolgt über die oberhalb der Linie platzierte Bezeichnung. Zusätzlich sind die Ankunfts- und Abfahrtszeiten der einzelnen Züge an den Stationen als Minutenwert über (Ankunft) und unter (Abfahrt) der Linie angegeben. Zum Beispiel fährt der *Nq 16119 W Sa* um 11:50 Uhr in der Station *Morbach* ein und verlässt diese um 11:51.

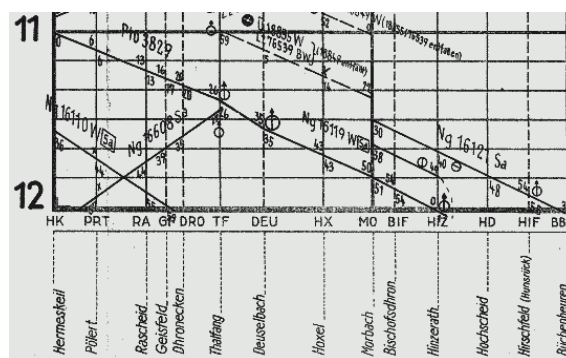


Abbildung 4.4: Ausschnitt eines Bildfahrplanes der Deutschen Bundesbahn, Bundesbahndirektion Saarbrücken aus dem Jahr 1970/71. Quelle: [Ruc10].

Für eine Aufbereitung von Fahrplandaten in ähnlicher Weise zu den 3-dimensionalen Raum/Zeit Diagrammen aus Abschnitt 4.2.2 werden die Bildfahrpläne als Vorbild herangezogen und die semantische Raumbeschreibung durch ein 2-dimensionales Koordinatenpaar ersetzt. Die Grundelemente, die Zeitpunkte und Zeitspannen aus Abschnitt 4.1.1, werden hierfür durch 3-dimensionale OGC-konforme Punkt- und Liniengeometrien nach [Ope06a] beschrieben. Das hierbei verwendete Koordinatentripel setzt sich aus Rechts-, Hoch- und Zeitwert zusammen. Nachfolgend sind ein Zeitpunkt – fahrplanmäßige Abfahrt des Intercity (IC) Zuges 552 der Österreichischen Bundesbahn OeBB von Graz Hauptbahnhof nach Wien Meidling am 01.01.2011 um 06:26 Uhr – und die Zeitspanne – fahrplanmäßige Route des OeBB IC 552, beschränkt auf die Abfahrt in Graz am 01.01.2011 um 06:26 Uhr und der Ankunft in Wien Meidling am 01.01.2011 um 08:58 – angeführt (Raumkoordinate: Universal Transverse Mercator (UTM) Zone 33N bezogen auf das World Geodetic System 1984 (WGS84); Zeitkoordinate: JD):

- POINT(531680 5213299 2455562.76806).
- LINESTRING(531680 5213299 2455562.76806,599141 5336558 2455562.87360).

Mit Hilfe dieser geometrischen Beschreibungsform räumlicher Vorgänge wird im Anschluss das Modellierungsverfahren für drei Situationen im Fahrplanverkehr im Detail beschrieben. Bei diesen drei Situationen handelt es sich um die einfache Haltestelle (Abschnitt 4.3.1), um die komplexe Haltestelle (Abschnitt 4.3.2) und um den Wechsel zwischen dem

Individualverkehr, als Vertreter zeitunabhängiger Vorgänge, und dem Fahrplanverkehr (Abschnitt 4.3.3).

4.3.1 Einfache Haltestellen

Als *einfache Haltestelle* wird in diesem Abschnitt jene Situation beschrieben, bei der ein Fahrgast nur mit dem Verkehrsmittel einer einzelnen Linie interagieren kann und keine Möglichkeit zum Umstieg in eine andere Linie besitzt. Die Interaktionsmöglichkeiten teilen sich somit wie folgt auf:

- Warten auf das Verkehrsmittel an der Haltestelle.
- Einsteigen in das Verkehrsmittel an der Haltestelle.
- Aussteigen aus dem Verkehrsmittel an der Haltestelle.
- Fahrt mit dem Verkehrsmittel an der Haltestelle fortsetzen.

In Abbildung 4.5 wird eine einfache Haltestelle als 2-dimensionales Raum/Zeit Diagramm dargestellt. Die diagrammetrische Aufbereitung ist hierbei dem Bildfahrplan nachempfunden – Rauminformation entlang der Abszisse und Zeitinformation entlang der Ordinate – wobei die Orientierung der Zeitachse entgegengesetzt verläuft.

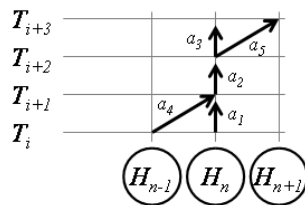


Abbildung 4.5: Situationsdarstellung einer einfachen Haltestelle H_n .

Der zeitliche Horizont, zu dem die Situation modelliert wird, erstreckt sich vom Zeitpunkt T_i bis zum Zeitpunkt T_{i+3} und ist als zeitlicher Ausschnitt aus dem Gesamtfahrplan zu verstehen. Räumlich wird die Haltestelle H_n , ihr Vorgänger H_{n-1} und ihr Nachfolger H_{n+1} innerhalb des Liniennetzes betrachtet. Die einzelnen Pfeile, a_1 bis a_5 , stellen Prozesse dar, wie *Warten an der Haltestelle* oder *Fahren zwischen zwei Haltestellen*.

Die geometrisch/topologische Modellierung erfolgt nach den Gesichtspunkten einer Kanten-Traversen Struktur (vgl. Abschnitt 3.2). Mit Hilfe der Kanten werden Prozesse beschrieben, die mittels Traversen zu einer zusammenhängenden Topologie verknüpft werden. Durch die thematische Besonderheit, zeitabhängige Vorgänge zu modellieren, ist sowohl auf die Ausrichtung der Kanten wie auch auf die Definition der Traversen zu achten. Keinesfalls dürfen zeitbezogene Unstimmigkeiten, wie zeitlich rückwärts verlaufende Prozesse im Zuge der Modellierungsphase entstehen.

Das Warten auf das Verkehrsmittel an der Haltestelle hat Tabelle 4.4 zum Inhalt und bezieht sich hierbei auf die zuvor angesprochene Abbildung 4.5. Der Vorgang des Wartens kann als alleinige Änderung entlang der Zeitachse verstanden werden. Die OGC-konforme

2-dimensionale geometrische Beschreibung des Wartens verdeutlicht diese Eigenheit: Nur die zweite Koordinatenkomponente weist eine Veränderung im positiven Sinne auf. Definiert man eine Traverse, die zwei Warte-Prozesse miteinander in Beziehung bringt, entspricht dies einem Fortsetzen des Wartens. Vor dem Hintergrund des Abschnittes 4.1 über die Arten von Beziehungen zeitabhängiger Vorgänge, ist das Fortführen des Wartens nur bei einer *Meets*-Beziehung zweier Warte-Prozesse zuzulassen.

Tabelle 4.4: Modellierung einer einfachen Haltestelle – Warten auf das Verkehrsmittel an der Haltestelle.

Graphik	Beschreibung
	<p>Geometrische Elemente:</p> <ul style="list-style-type: none"> • Warten $a_1 = \text{LINESTRING}(H_n T_i, H_n T_{i+1})$. • Warten $a_2 = \text{LINESTRING}(H_n T_{i+1}, H_n T_{i+2})$. <p>Topologische Beziehung:</p> <ul style="list-style-type: none"> • Warten fortsetzen $t = (a_1, a_2)$ entspricht a_1 Meets a_2.

Die nächste Interaktionsmöglichkeit, das Einsteigen in das Verkehrsmittel an der Haltestelle, wird in Tabelle 4.5 behandelt. Hier wird der Vorgang des Einsteigens durch die topologische Beziehung eines Warte-Prozesses mit einem Fahr-Prozess modelliert. In den Graphiken der Tabelle 4.5 kommt zusätzlich zu den Warte-Prozess beschreibenden Kanten jene Kante hinzu, die eine räumliche – von Ort H_n zu Ort H_{n+1} – und zeitliche Änderung – von Zeitpunkt T_{i+2} zu Zeitpunkt T_{i+3} – beschreibt: a_5 .

Als Bedingung, in ein Verkehrsmittel einsteigen zu können, ist dessen Anwesenheit an der Haltestelle zu sehen. Bezogen auf Abbildung 4.5 entsprechen die Pfeile a_4 , a_2 und a_5 dem raumzeitlichen Verhalten des Verkehrsmittels. Somit ist bereits ab dem Zeitpunkt T_{i+1} das Einsteigen in das Verkehrsmittel möglich. Für die Modellierung ergeben sich daraus folgende Konsequenzen: Je nach Länge der Zeitspanne zwischen dem Warte-Prozess und dem Fahr-Prozess, die für den Einsteigevorgang kombiniert werden, kann in langsames – bei großer Zeitspanne bzw. *Before*-Beziehung nach Abschnitt 4.1, wie bei der Kombination von a_1 mit a_5 – und schnelles Einsteigen – bei direktem Anschluss der beiden Prozesse bzw. *Meets*-Beziehung nach Abschnitt 4.1, wie bei der Kombination von a_2 mit a_5 – unterschieden werden. Hinsichtlich der Modellierung dieses Vorganges in einer Kanten-Traversen Struktur stellt die Lücke zwischen den Kanten a_1 und a_5 , die durch eine Traverse zu verbinden sind, kein Problem dar – Topologien sind unabhängig von deren geometrischen Repräsentationsformen zu betrachten.

Verglichen mit dem Einsteigen in ein Verkehrsmittel an der Haltestelle lässt sich das Aussteigen aus dem Verkehrsmittel an der Haltestelle ähnlich behandeln. Die involvierten Prozesse sind die selben – ein Warte-Prozess und ein Fahr-Prozess – nur deren zeitlicher

Tabelle 4.5: Modellierung einer einfachen Haltestelle – Einsteigen in das Verkehrsmittel an der Haltestelle.

Graphik	Beschreibung
	<p>Geometrische Elemente:</p> <ul style="list-style-type: none"> • Warten $a_1 = \text{LINESTRING}(H_n T_i, H_n T_{i+1}).$ • Fahren $a_5 = \text{LINESTRING}(H_n T_{i+2}, H_{n+1} T_{i+3}).$ <p>Topologische Beziehung:</p> <ul style="list-style-type: none"> • Langsames Einsteigen $t = (a_1, a_5)$ entspricht a_1 Before a_5.
	<p>Geometrische Elemente:</p> <ul style="list-style-type: none"> • Warten $a_2 = \text{LINESTRING}(H_n T_{i+1}, H_n T_{i+2}).$ • Fahren $a_5 = \text{LINESTRING}(H_n T_{i+2}, H_{n+1} T_{i+3}).$ <p>Topologische Beziehung:</p> <ul style="list-style-type: none"> • Schnelles Einsteigen $t = (a_2, a_5)$ entspricht a_2 Meets a_5.

Ablauf hat gewechselt: Zuerst erfolgt hier der Fahr-Prozess a_4 und danach der Warte-Prozess a_2 bzw. a_3 .

Unter Rücksichtnahme auf die Verweildauer des Verkehrsmittels an der Haltestelle lassen sich langsame und schnelle Prozesse beschreiben (vgl. langsames und schnelles Einsteigen). Solange das Verkehrsmittel an der Haltestelle steht, ist ein Aussteigen möglich. Die beiden modellierbaren Vorgänge, das langsame Aussteigen bei Kombination von a_4 mit a_3 und das schnelle Aussteigen bei der Kombination von a_4 und a_2 , sind in Tabelle 4.6 graphisch dargestellt und unter dem Gesichtspunkt einer Kanten-Traversen Struktur beschrieben.

Die letzte Interaktionsart zwischen Fahrgast und Verkehrsmittel an einer Haltestelle ist das Fortsetzen der Fahrt. Für die in Abbildung 4.5 dargestellte Gesamtsituation der zu modellierenden Haltestelle listet Tabelle 4.7 die einzige Möglichkeit einer Weiterfahrt auf. Hierfür werden die beiden Fahr-Prozesse a_4 und a_5 durch eine entsprechende Traverse in Beziehung gebracht.

In diesem speziellen Fall der Haltestellensituation besteht eine zeitliche *Before*-Beziehung nach Abschnitt 4.1, das einer längeren Verweildauer des Verkehrsmittels an der Haltestelle entspricht. Im Zuge der Modellierung einer komplexen Haltestelle (vgl. Abschnitt 4.3.2)

Tabelle 4.6: Modellierung einer einfachen Haltestelle – Aussteigen aus dem Verkehrsmittel an der Haltestelle.

Graphik	Beschreibung
	<p>Geometrische Elemente:</p> <ul style="list-style-type: none"> • Fahren $a_4 = \text{LINESTRING}(H_{n-1} T_i, H_n T_{i+1})$. • Warten $a_3 = \text{LINESTRING}(H_n T_{i+2}, H_n T_{i+3})$. <p>Topologische Beziehung:</p> <ul style="list-style-type: none"> • Langsames Aussteigen $t = (a_4, a_3)$ entspricht a_4 Before a_3.
	<p>Geometrische Elemente:</p> <ul style="list-style-type: none"> • Fahren $a_4 = \text{LINESTRING}(H_{n-1} T_i, H_n T_{i+1})$. • Warten $a_2 = \text{LINESTRING}(H_n T_{i+1}, H_n T_{i+2})$. <p>Topologische Beziehung:</p> <ul style="list-style-type: none"> • Schnelles Aussteigen $t = (a_4, a_2)$ entspricht a_4 Meets a_2.

wird auch eine unmittelbare Fortführung des Verkehrsmittels in der Haltestellensituation beschrieben. Eine spezielle Unterscheidung zwischen langsamem und schnellem Fortsetzen der Fahrt erfolgt nicht, da der Fahrgast sein Verhalten nicht an die Situation anpassen muss – er bleibt im Verkehrsmittel.

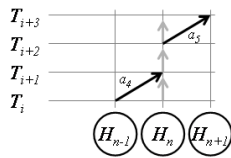
Betrachtet man erneut den Bildfahrplan aus Abbildung 4.4 so können Situationen erkannt werden, die mit langsamen und schnellen Einsteige- und Aussteige-Vorgängen zu assoziieren sind. Zur Verdeutlichung sind nachfolgend zwei Beispiele angeführt, die sich auf die Zugverbindung *Ng 16199 W Sa* beziehen:

- Station *Hoxel* mit Ankunft um 11:43 Uhr und Abfahrt um 11:43 Uhr als Beispiel für schnelle Vorgänge: Das Ein- und Aussteigen hat vom Fahrgast sofort zu erfolgen.
- Station *Morbach* mit Ankunft um 11:50 Uhr und Abfahrt um 11:51 Uhr als Beispiel für langsame Vorgänge: Für das Ein- und Aussteigen hat der Fahrgast eine Minute Zeit.

Weiters kann die Eigenschaft *schnell* oder *langsam* des jeweiligen Vorganges in der Kosteneigenschaft der modellierten Traverse $^c t$ Eingang finden. Dadurch kann im Zuge der

kürzesten Wegesuche auf Personengruppen Rücksicht genommen werden, die für das Ein- und Aussteigen mehr Zeit benötigen. Diese Bewertungsmöglichkeit beeinflusst in weiterer Folge auch das Umsteigen und kann genauso an Personengruppen angepasst werden.

Tabelle 4.7: Modellierung einer einfachen Haltestelle – Fahrt mit dem Verkehrsmittel an der Haltestelle fortsetzen.

Graphik	Beschreibung
	<p>Geometrische Elemente:</p> <ul style="list-style-type: none"> • Fahren $a_4 = \text{LINESTRING}(H_{n-1} T_i, H_n T_{i+1})$. • Fahren $a_5 = \text{LINESTRING}(H_n T_{i+2}, H_{n+1} T_{i+3})$. <p>Topologische Beziehung:</p> <ul style="list-style-type: none"> • Fahrt fortsetzen $t = (a_4, a_5)$ entspricht a_4 Before a_5.

4.3.2 Komplexe Haltestellen

Die hier beschriebene komplexe Haltestelle unterscheidet sich von einer einfachen Haltestelle durch die Möglichkeit, die Linie zu wechseln. Verkehrsmittel mindestens zweier Linien fahren somit diese Haltestelle an. Die Anzahl der Interaktionsmöglichkeiten, die in weiterer Folge modelliert werden, bleibt jedoch im Vergleich zur einfachen Haltestelle aus Abschnitt 4.3.1 unverändert. Das Umsteigen kann als eine Kombination der Aus- und Einsteige-Vorgänge beschrieben werden.

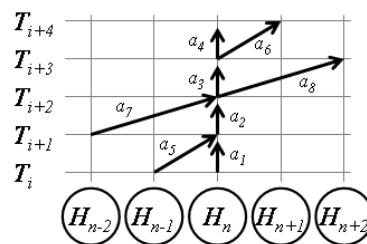


Abbildung 4.6: Situationsdarstellung einer komplexen Haltestelle H_n .

Abbildung 4.6 zeigt eine komplexe Haltestellensituation, die vom diagrammetrischen Aufbau jener aus Abbildung 4.5 gleicht. Der hier abgebildete räumliche und zeitliche Ausschnitt ist jedoch auf H_{n-2} bis H_{n+2} und T_i bis T_{i+4} erweitert. Die Zuteilung dieser Haltestellensituation zur Gruppe der komplexen Haltestellen begründet sich in der Nutzung der Haltestelle

H_n durch zwei Linien: $L_1 = (a_5, a_2, a_3, a_6)$ von H_{n-1} über H_n nach H_{n+1} und $L_2 = (a_7, a_8)$ von H_{n-2} über H_n nach H_{n+2} .

Tabelle 4.8: Modellierung einer komplexen Haltestelle – Umsteigen an der Haltestelle von einem Verkehrsmittel der Linie L_1 in ein Verkehrsmittel der Linie L_2 .

Graphik	Beschreibung
	<p>Geometrische Elemente:</p> <ul style="list-style-type: none"> • Fahren mit Linie L_1: $a_5 = \text{LINESTRING}(H_{n-1} T_i, H_n T_{i+1})$. • Warten: $a_2 = \text{LINESTRING}(H_n T_{i+1}, H_n T_{i+2})$. <p>Topologische Beziehung:</p> <ul style="list-style-type: none"> • Schnelles Aussteigen aus Linie L_1: $t = (a_5, a_2)$ entspricht a_5 Meets a_2.
	<p>Geometrische Elemente:</p> <ul style="list-style-type: none"> • Warten: $a_2 = \text{LINESTRING}(H_n T_{i+1}, H_n T_{i+2})$. • Fahren mit Linie L_2 : $a_8 = \text{LINESTRING}(H_n T_{i+2}, H_{n+2} T_{i+3})$. <p>Topologische Beziehung:</p> <ul style="list-style-type: none"> • Schnelles Einsteigen in Linie L_2: $t = (a_2, a_8)$ entspricht a_2 Meets a_8.

Einer der beiden beschreibbaren Umsteige-Vorgänge der komplexen Haltestellensituation aus Abbildung 4.6 – von Linie L_1 zu Linie L_2 – wird in Tabelle 4.8 stellvertretend für alle anderen Umsteige-Vorgänge aufgeschlüsselt. Hierbei werden ein schnelles Aussteigen aus Linie L_1 , modelliert durch die Traverse $t = (a_5, a_2)$, mit einem schnellen Einsteigen in Linie L_2 , modelliert durch die Traverse $t = (a_2, a_8)$, miteinander kombiniert.

Eine explizite Modellierung eines Umsteige-Vorganges als Traverse $t = (a_5, a_8)$, wie er in Tabelle 4.8 durch zwei separate Vorgänge beschrieben ist, wird nicht umgesetzt. In Anlehnung an die reale Welt muss der Fahrgast zuerst aus dem Verkehrsmittel der ersten Linie aussteigen, anschließend einen mehr oder weniger langen Warte-Prozess durchlaufen bis er schließlich in das Verkehrsmittel der zweiten Linie einsteigen kann. Ein umgangssprachlicher *direkter Anschluss*, vergleichbar mit einer *Meets*-Beziehung nach Abschnitt 4.1, bei der der Ankunftszeitpunkt der ersten Linie T_1^{Ankunft} dem Abfahrtszeitpunkt der zweiten Linie T_2^{Abfahrt} entspricht – eine *Equals*-Beziehung nach Abschnitt 4.1 – würde der Fahrgast den

Anschluss sprichwörtlich *vor der Nase davon fahren* sehen.

4.3.3 Wechsel zwischen Individualverkehr und Fahrplanverkehr

In multimodalen Verkehrsszenarien, wie sie bei [NH06] oder bei [Sch05] auch als *kombinierter Verkehr* beschrieben werden, sind für eine gesamtheitliche Planung individuelle Verkehrsnetze und Netzwerke, die einem Fahrplan zugrunde liegen, gemeinsam zu betrachten. Bei einer Kombination dieser unterschiedlichen Datengrundlagen ist auf die barrierefreie Ausbreitung der Suchbäume im Zuge der kürzesten Wegesuche zu achten. Dementsprechend sind die Haltestellenmodelle aus den Abschnitten 4.3.1 und 4.3.2, an denen der Wechsel zwischen Individualverkehr und Fahrplanverkehr stattfindet, zu ergänzen.

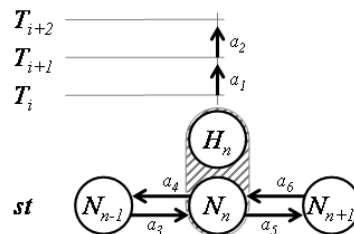


Abbildung 4.7: Situationsdarstellung einer Haltestelle H_n mit anschließendem Individualverkehrsnetz über den Wegeknoten N_n .

Das 2-dimensionale Raum/Zeit Diagramm in Abbildung 4.7 zeigt die Verknüpfungsstelle zwischen Individualverkehr und Fahrplanverkehr auf und beschränkt sich dabei auf jene Elemente, die für eine Modellierung des Netzwerkwechsels benötigt werden:

- Netzwerkelemente ohne spezielle zeitliche Zuordnung (st): Die gerichteten Kanten a_3 und a_4 zwischen den Knoten N_{n-1} und N_n bzw. die gerichteten Kanten a_5 und a_6 zwischen den Knoten N_n und N_{n+1} .
- Elemente des Fahrplannetzes: Die Warteprozesse a_1 zwischen den Zeitpunkten T_i und T_{i+1} und a_2 zwischen den Zeitpunkten T_{i+1} und T_{i+2} .

Die Verknüpfungsstelle kann räumlich als N_n beziehungsweise H_n identifiziert werden. Graphisch wird auf diese Gemeinsamkeit des Ortes im Diagramm durch eine Schraffur speziell hingewiesen. Auf die Thematik der zeitlichen Verknüpfung der beiden Datengrundlagen wird an späterer Stelle dieses Abschnittes eingegangen (siehe Tabellen 4.10 und 4.11).

In Tabelle 4.9 werden beispielhaft zwei Vorgänge dargestellt, die sich zur Gänze im Netzwerk des Individualverkehrs befinden. Räumlich betrachtet ist die Haltestelle in beiden Vorgängen involviert, jedoch die zeitabhängigen Elemente a_1 und a_2 bleiben ungenützt. Der erste Vorgang (a_3, a_5) beschreibt ein Fortsetzen der Bewegung über die Haltestelle N_n bzw. H_n hinaus. Die beteiligten Kanten des zweiten Vorganges, a_3 und a_4 , können unterschiedlicher Weise – in Abhängigkeit der Reihenfolge – in Beziehung gebracht werden: Kombiniert man die Kanten als (a_3, a_4) , so erfolgt an der Haltestelle N_n bzw. H_n ein Umkehren. Werden die Kanten in umgekehrter Reihenfolge zu vorhin in Beziehung gebracht, (a_4, a_3) , so kann es als Rückkehr zur Haltestelle N_n bzw. H_n verstanden werden.

Tabelle 4.9: Modellierung einer Haltestelle – Bewegung im Netz des Individualverkehrs an der Haltestelle fortsetzen.

Graphik	Beschreibung
	<p>Geometrische Elemente:</p> <ul style="list-style-type: none"> • Bewegung im Netz des Individualverkehrs: $a_3 = \text{LINESTRING}(N_{n-1} \text{ st}, N_n \text{ st})$. • Bewegung im Netz des Individualverkehrs: $a_5 = \text{LINESTRING}(N_n \text{ st}, N_{n+1} \text{ st})$. <p>Topologische Beziehung:</p> <ul style="list-style-type: none"> • Bewegung fortsetzen: $t = (a_3, a_5)$.
	<p>Geometrische Elemente:</p> <ul style="list-style-type: none"> • Bewegung im Netz des Individualverkehrs: $a_3 = \text{LINESTRING}(N_{n-1} \text{ st}, N_n \text{ st})$. • Bewegung im Netz des Individualverkehrs: $a_4 = \text{LINESTRING}(N_n \text{ st}, N_{n-1} \text{ st})$. <p>Topologische Beziehung:</p> <ul style="list-style-type: none"> • Bewegung fortsetzen: $t = (a_3, a_4)$ bzw. $t = (a_4, a_3)$.

Das Betreten der Haltestelle, der Übergang vom Netzwerk des Individualverkehrs in das zeitabhängige Netzwerk des Fahrplanverkehrs, hat Tabelle 4.10 zum Inhalt. Hierbei wird eine den Bewegungsprozess beschreibende Kante des Individualverkehrsnetzes mit einer den Warte-Prozess beschreibenden Kante des Fahrplannetzes durch eine Traverse in Beziehung gebracht. Auf die Reihenfolge der Kanten in der Traversendefinition ist besonders zu achten, da bei einer Vertauschung nicht das Betreten, sondern das Verlassen der Haltestelle (vgl. Tabelle 4.11) beschrieben wird!

Das Betreten ist der einzige zeitkritische Vorgang bei einer multimodalen Suchbaumausbildung dieser Art. Je nach Zeitpunkt, an dem der Suchbaum den Knoten N_n des Individualverkehrsnetzes erreicht, ist für eine weitere Ausbreitung des Suchbaumes der zeitlich passende Warte-Prozess zu berücksichtigen. Dementsprechend werden in Tabelle 4.10 die Kantenkombinationen (a_3, a_1) und (a_3, a_2) angeführt, die sich nur durch die zeitliche Zuordnung des Warte-Prozesses unterscheiden. Dieser Warte-Prozess muss hierfür entweder eine *After*- oder eine *MetBy*-Beziehung zum Bewegungsprozess aufweisen.

Diese Anforderung bedingt ein ständiges Mitführen einer Zeitinformation zu den einzelnen Suchbaumelementen während der gesamten kürzesten Wegesuche: Der Zeitpunkt,

an dem die Benutzung des jeweiligen Elementes abgeschlossen ist. Für die hier eingesetzte Kanten-Traversen Struktur kann dieser Zeitpunkt als T_a und in allgemeiner Nomenklatur nach Abschnitt 2.2 als T_β beschrieben werden. Wesen und Wirken dieses Zeitattributes kann großteils mit jenem des Labelwertes eines Basiselementes ${}^t\beta$ gleichgesetzt werden: Der in T_β ausgewiesene Zeitpunkt berechnet sich unter Abhängigkeit des Startzeitpunktes T_S aus der Summe aller Zeitspannen, die den Basis- ${}^t\beta$ und Verbindungselementen ${}^t\alpha$ zugewiesen sind (vgl. Formel 4.1). So wie der Labelwert die räumliche Ausbreitung des Suchbaumes in Algorithmus 1 aus Abschnitt 2.2.3 steuert, so beeinflusst das Zeitattribut die zeitliche Ausbreitung des Suchbaumes an der Haltestelle.

$$T_S + \sum_{\{\beta_S, \dots, \beta_i\}} {}^t\beta + \sum_{\{\beta_S, \dots, \beta_i\}} {}^t\alpha = T_\beta. \quad (4.1)$$

Die Zeitspannen ${}^t\beta$ und ${}^t\alpha$ können mit den Kostenwerten ${}^c\beta$ und ${}^c\alpha$ verglichen werden, wobei die numerischen Werte der Zeitspannen beispielsweise als Funktionen der Kosten und des Bewegungsmodus M (z.B.: Fußgänger oder Autofahrer) abzuleiten sind, zum Beispiel ${}^t\beta = f({}^c\beta, M)$.

Tabelle 4.10: Modellierung einer Haltestelle – Betreten der Haltestelle.

Graphik	Beschreibung
	<p>Geometrische Elemente:</p> <ul style="list-style-type: none"> • Bewegung im Netz des Individualverkehrs: $a_3 = \text{LINESTRING}(N_{n-1} \text{ st}, N_n \text{ st})$. • Warten: $a_1 = \text{LINESTRING}(H_n T_i, H_n T_{i+1})$. <p>Topologische Beziehung:</p> <ul style="list-style-type: none"> • Betreten der Haltestelle: $t = (a_3, a_1)$ entspricht a_1 After a_3.
	<p>Geometrische Elemente:</p> <ul style="list-style-type: none"> • Bewegung im Netz des Individualverkehrs: $a_3 = \text{LINESTRING}(N_{n-1} \text{ st}, N_n \text{ st})$. • Warten: $a_2 = \text{LINESTRING}(H_n T_{i+1}, H_n T_{i+2})$. <p>Topologische Beziehung:</p> <ul style="list-style-type: none"> • Betreten der Haltestelle: $t = (a_3, a_2)$ entspricht a_2 After a_3.

Im Vergleich zum Betreten der Haltestelle ist das Verlassen der Haltestelle weit weniger zeitkritisch zu betrachten. Die Modellierung dieses Vorganges erfolgt durch die Beziehung zwischen einem Warte-Prozess und einem Bewegungsprozess in genau dieser Reihenfolge. In Tabelle 4.11 werden hierfür die Traversen (a_1, a_4) und (a_2, a_4) angeführt. Da diese Traversen die Vorgänge von einem zeitabhängigen Fahrplannetzwerk hin zu einem zeitunabhängigen Netzwerk – Netzwerk des Individualverkehrs – beschreiben, kann dieser Wechsel ohne spezielle Rücksichtnahme auf das Zeitattribut der involvierten Kanten $^T a$ durchgeführt werden.

Tabelle 4.11: Modellierung einer Haltestelle – Verlassen der Haltestelle.

Graphik	Beschreibung
	<p>Geometrische Elemente:</p> <ul style="list-style-type: none"> • Warten: $a_1 = \text{LINESTRING}(H_n T_i, H_n T_{i+1})$. • Bewegung im Netz des Individualverkehrs: $a_4 = \text{LINESTRING}(N_n \text{ st}, N_{n-1} \text{ st})$. <p>Topologische Beziehung:</p> <ul style="list-style-type: none"> • Verlassen der Haltestelle: $t = (a_1, a_4)$ entspricht a_1 Before a_4.
	<p>Geometrische Elemente:</p> <ul style="list-style-type: none"> • Warten: $a_2 = \text{LINESTRING}(H_n T_{i+1}, H_n T_{i+2})$. • Bewegung im Netz des Individualverkehrs: $a_4 = \text{LINESTRING}(N_n \text{ st}, N_{n-1} \text{ st})$. <p>Topologische Beziehung:</p> <ul style="list-style-type: none"> • Verlassen der Haltestelle: $t = (a_2, a_4)$ entspricht a_2 Before a_4.

Kapitel 5

Methoden der Suchbaumausbreitung

Im Verlaufe des Kapitels *Methoden der Suchbaumausbreitung* werden ausgehend von einem Standardverfahren, wie es in Kapitel 2 beschrieben wird, stufenweise Ergänzungen eingeführt, die schlussendlich schneller zu einer Lösung eines SPP führen. Das Standardverfahren, das hier in Abschnitt 5.1 aufbereitet wird, kennt ein einzelnes Startelement zur Ausbreitung des Suchbaumes. Ein zusätzliches Startelement für einen zweiten Suchbaum, das am Ziel der gesuchten Route zu definieren ist, wird in Abschnitt 5.2 über bidirektionale Routenberechnung beschrieben. Berücksichtigt man bei der Suchbaumausbreitung hierarchische Strukturen innerhalb der Datengrundlage, so eröffnen sich zusätzliche Möglichkeiten das Ausbreitungsverfahren zu beschleunigen (vgl. Abschnitt 5.3). Der letzte Abschnitt dieses Kapitels – Abschnitt 5.4 – führt die Methode des hierarchischen Suchbaumausbreitens soweit fort, dass nicht nur der Wechsel in eine höhere Hierarchie genutzt wird, sondern auch der Abstieg von einer höheren in eine niedrigere hierarchische Ebene für die Suchbaumausbreitung als eine zulässige Aktion Verwendung finden kann.

Alle angeführten Beispiele innerhalb dieses Kapitels beziehen sich durchgehend auf eine Kanten-Traversen Struktur, wie sie in Abschnitt 3.2 erläutert wird. Die Übertragung der nachfolgend beschriebenen Verfahren auf Knoten-Kanten Strukturen, polygonale und polyhedrale Modelle sowie gemischte Strukturen ist zulässig, solange keine speziellen Vorgänge zu modellieren und zu analysieren sind. Zu diesen Spezialfällen zählt unter anderem die bidirektionale Suchbaumausbreitung, basierend auf Knoten-Kanten Strukturen bei gleichzeitiger Berücksichtigung von Abbiegeverboten.

5.1 Monodirektionale Suchbaumausbreitung

Monodirektionale Verfahren der Suchbaumausbreitung, bei [KK97] auch als unidirektionale Verfahren beschrieben, sind durch die Nutzung eines einzelnen Startelementes gekennzeichnet. In Abschnitt 2.2 wird der Algorithmus nach *Dijkstra* bereits als monodirektionales Verfahren aufbereitet. In [Dij59], worin unter *Problem 2* der Weg mit der geringsten Länge zwischen den Knoten P und Q gesucht wird, beginnt der Algorithmus mit „[...] We now transfer node P to set A [...]“ und wird so lange ausgeführt bis „[...] until node Q is trans-

ferred to set A [...]“. Diese Herangehensweise, den Suchbaum solange von einem einzelnen Startelement auszubreiten, bis das Zielelement als ein Bestandteil des Suchbaumes vorliegt, kann als Hauptcharakteristikum eines monodirektionalen Verfahrens gesehen werden.

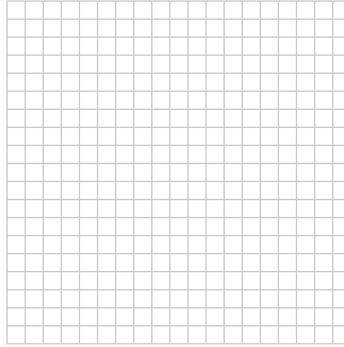


Abbildung 5.1: Ausgangsdaten – Kanten einer Kanten-Traversen Struktur.

Zur Verdeutlichung dieser Suchbaumausbreitungsmethode wird basierend auf einer Kanten-Traversen Struktur in *Manhattan Metrik* (vgl. [Eng10], [GLN01]) ein Suchbaum ausgebreitet. Dieser Datensatz, wie er in Abbildung 5.1 dargestellt wird, besteht aus Basis- und Verbindungselementen, die jeweils den identen Kostenwert aufweisen. Eine Route, die in solch einer Manhattan Metrik ermittelt wird, ist als eine optimale Lösung zu sehen, solange sie sich horizontal bzw. vertikal in Richtung Ziel bewegt.

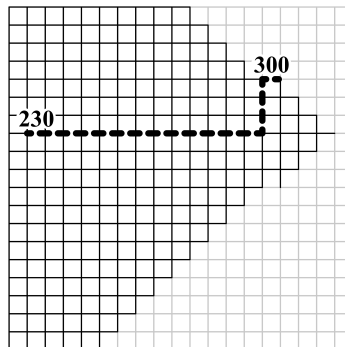


Abbildung 5.2: Monodirektionale Suchbaumausbreitung vom Startelement 230 zum Zielelement 300.

Für den Fall einer monodirektionalen Suchbaumausbreitung wird eine Route zwischen den Kanten 230 (Startkante) und 300 (Zielkante) berechnet. Hierbei handelt es sich um ein klassisches SSSDP. Das Berechnungsergebnis ist in Abbildung 5.2 angeführt. Die grau gezeichneten Linien des gesamten Basisdatensatzes (Menge B) werden durch die durchgezogenen schwarzen Linien des ausgebreiteten Suchbaumes (Menge $P \subseteq B$) überlagert. Im Zentrum des Suchbaumes befindet sich die Startkante 230. Durch eine schwarz-strichlierte

Linie wird die berechnete Route dargestellt, die das Start- mit dem Zielelement verbindet. Es ist erkennbar, dass der Ausbreitungsalgorithmus zu jenem Zeitpunkt beendet wurde, an dem das Zielelement Teil des Suchbaumes wurde: Es sind keine Suchbaumelemente zu erkennen, die sich jenseits der Kante 300 befinden.

Bei der Anwendung des Berechnungsergebnisses innerhalb eines Zielführungsprozesses, werden beispielsweise von einem Autonavigationssystem jene Manöver audiovisuell aufbereitet, mit Hilfe derer sich der Autofahrer entlang der optimalen Route fortbewegen kann. Verlässt der Autofahrer die berechnete Route, so ist eine neue Route zwischen einem Startelement – die aktuellen Position des Autos – und dem bekannten Zielelement zu ermitteln. Die Notwendigkeit einer erneuten Suchbaumausbreitung lässt sich mit den Eingangsgraden der Basiselemente ($d_{in}(n_i) = 1$) von Baumstrukturen begründen (vgl. Abschnitt 2.2.3).

Eine alternative Variante des monodirektionalen Suchbaumausbreitungsverfahrens besitzt das Zielelement der gesuchten Route als Ausgangselement eines Suchbaumes: $\beta_D \equiv \beta'_S$. Wird hier der Suchbaum vom Ziel- zum Startelement – gegen die eigentliche Bewegungsrichtung – ausgebreitet, so sind die expliziten Nachbarschaftsverhältnisse, wie sie in den Traversen einer Kanten-Traversen Struktur dokumentiert sind, aufzubereiten. Die angeführten Vorgänger und Nachfolger einer Traverse sind zu vertauschen: $^p\alpha = ^s\alpha'$ und $^s\alpha = ^p\alpha'$. Die Basiselemente bleiben hierbei unverändert: $\beta = \beta'$. Daraus ergibt sich für die monodirektionale Suchbaumausbreitung vom Zielelement zum Startelement der Graph $G' = (B', A')$ als Datenbasis.

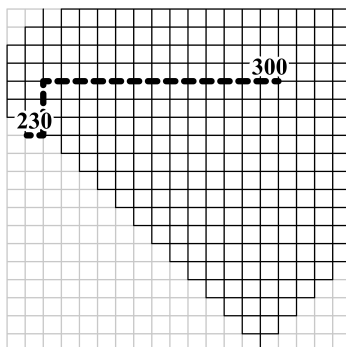


Abbildung 5.3: Monodirektionale Suchbaumausbreitung vom Zielelemente 300 zum Startelement 230.

In Abbildung 5.3 ist das Ergebnis einer Routenberechnung vom Zielelement 300 zum Startelement 230 dargestellt. Deutlich ist der ausgebreitete Suchbaum um die Kante 300 erkennbar, die als Ausgangselement verwendet wurde. Das hier abgebildete Ergebnis unterscheidet sich zwar optisch von jenem aus Abbildung 5.2, jedoch durch die Eigenschaft einer *Manhattan Metrik* sind die Kosten der beiden Routen bzw. in diesem Sonderfall die Anzahl der benutzten Kanten ident.

Im Falle des vom Kurs abkommenden Autofahrers muss nur dann der Suchbaum neu berechnet werden – Ausgangselement des neuen Suchbaumes bleibt das Zielelement der Route und Zielelement des Suchbaumes entspricht der aktuellen Position des Autos – wenn sich das Auto auf einer vom Suchbaum nicht erreichten Kante ($\beta \in U$) beziehungsweise auf einer Kante mit vorübergehendem Labelwert ($\beta \in T$) befindet.

5.2 Bidirektionale Suchbaumausbreitung

Eines jener Verfahren der Routenberechnung, das von [FSR06] zu den heuristischen Methoden gezählt wird, hat die gleichzeitige Ausbreitung zweier Suchbäume zum Inhalt: Die bidirektionale Suchbaumausbreitung. Hierbei wird ein Suchbaum um das Startelement der Route und ein weiterer Suchbaum um das Zielelement der Route ausgebreitet. Beim Eintreten eines vordefinierten Abbruchkriteriums – eine bestimmte Überlagerungssituation der beiden Suchbäume – wird der Ausbreitungsvorgang beendet. Daraus ergeben sich nach [FSR06] die zwei wesentlichen Faktoren, die diese Methode beeinflussen:

- Koordinierte Ausbreitung beider Suchbäume.
- Definition des Abbruchkriteriums.

In [Nic66] wird ein bidirektionales Verfahren beschrieben, das Suchbäume frei von Heuristiken ausbreitet (vgl. Abschnitt 5.1) und ein Abbruchkriterium nach Formel 5.1 verwendet. Hierbei wird die Summe der Labelwerte des Startsuchbaumes L^o mit jenem des Zielsuchbaumes L^d am sich überlagernden Knoten i mit der Summe der Labelwerten der übrigen Knoten j in Bezug gebracht. Die Leistungssteigerung dieses bidirektionalen Ansatzes im Vergleich zu einer monodirektionalen Suchbaumausbreitung wird jedoch in [Dre69] als geringfügig beurteilt.

$$L_{(i)}^o + L_{(i)}^d \leq \min_{j \in N} \{L_{(j)}^o\} + \min_{j \in N} \{L_{(j)}^d\}. \quad (5.1)$$

Eine Erweiterung der bidirektionalen Suchbaumausbreitung wird bei [Poh71] beschrieben. Hier werden die Suchbäume unter Zuhilfenahme einer Richtungsinformation heuristisch ausgebreitet: Bidirectional Heuristic Path Algorithm (BHPA). Durch die in Abschnitt 2.1.2 beschriebenen Eigenschaften heuristischer Methoden – Verkürzen der Berechnungszeit auf Kosten optimaler Ergebnisse – und der Wahl des Abbruchkriteriums liefert BHPA keine optimale Route (vgl. [KK97, S. 288]). Aus BHPA abgeleitete und weiterentwickelte Varianten finden sich unter anderem bei [Kwa89], [Fu96] und [PP06]. Für diese Art der bidirektionalen Suchbaumausbreitung – *Front-to-End* Algorithmen, bei denen der Suchbaum jeweils auf das gegenüberliegende Zielelement ausgerichtet wird – werden in [Nil80] Bedenken formuliert, die sich auf den Ort der Überlagerung und dem damit verknüpften Abbruchkriterium beziehen: Bieten die Datengrundlagen für das gestellte SSSDP eine ungünstige Umgebung, so treffen sich die Suchbäume nicht auf halbem Weg zwischen Start und Ziel. Durch diese Eigenheit können die beiden Suchbäume des bidirektionalen Verfahrens in Summe auf die doppelte Größe eines Suchbaumes anwachsen, der einem monodirektionalen Routing-Algorithmus zugrunde liegt. Der Begriff *Missile Metaphor*, geprägt durch [Nil80], spiegelt diese Problematik wider, wurde aber bei [KK93] kritisch hinterfragt.

Als eine Reaktion auf den *Missile Metaphor* sind die *Front-to-Front* Algorithmen zu sehen, bei denen die Suchbäume durch aktives Eingreifen in den Ausbreitungsprozess zueinander ausgerichtet werden und nicht auf das gegenüberliegende Zielelement zustreben. Bei [DS77] wird ein Bidirectional Heuristic Front-to-Front Algorithm (BHFFA) beschrieben, der in weiterer Folge als BHFFA2 weiterentwickelt wurde (vgl. [De 83]). Der als *d-node retargeting* bezeichnete Algorithmus, [PP84] ist ebenfalls dieser Methodik zuzuordnen. Den erheblichen Berechnungsaufwand dieser Algorithmen kann laut [KK97] weder eine Einschränkung der Basisdaten, noch die Richtungsvorgabe der Ausbreitung kompensieren.

Eine Weiterentwicklung der bidirektionalen Algorithmen, auch als *non-traditional* Ansatz bezeichnet (vgl. [KK97]), setzt sich unter anderem intensiv mit der Verspeicherung des äußeren Randes des Suchbaumes auseinander. In diese Klasse der Algorithmen fallen beispielsweise der Iterative-Deepening Perimeter Search (IDPS*) von [DN94] oder der Bidirectional Iterative-Deepening A* (BIDA*) von [Man95], die beide als *perimeter search* zu bezeichnen sind. Auch das hybride Verfahren von [KK97] ist zu den *non-traditional* Ansätzen zu zählen.

In dieser Arbeit wird ein bidirektionaler Algorithmus zur Suchbaumausbreitung als funktionelle Grundlage für eine hierarchische Suchbaumausbreitung in den Abschnitten 5.3 und 5.4 benötigt. Generell wird auf eine Kombination von bidirektionalen und hierarchischen Verfahren in der Literatur verwiesen (u.a. [BDS⁺10], [FSR06] oder [CTB01]). Um diese Funktionalität zu erlangen wird jene Komplexität vom Algorithmus verlangt, wie sie bereits bei [Nic66] beschrieben wird: Heuristisch freie Ausbreitung zweier Suchbäume – Start- und Zielelement der gesuchten Route als Ausgangselement dieser Suchbäume – und Abbruch der Ausbreitung bei der ersten Überlagerung der Suchbäume.

Die Datengrundlage bilden hierfür zwei Graphen, ein primaler Graph $G = (B, A)$ und sein inverser Graph $G' = (B', A')$. Die Inversion von G zu G' bezieht sich ausschließlich auf die Vertauschung von Vorgänger und Nachfolger innerhalb des Verbindungselementes α (vgl. alternative Suchbaumausbreitung in Abschnitt 5.1). Je nach Startelement β_S des jeweiligen Suchbaumes ST wird entweder der primale Graph (Startelement der Route: $\beta_{Sroute} \equiv \beta_{SST(G)}$) oder der inverse Graph (Zielelement der Route $\beta_{Droute} \equiv \beta_{SST(G')}$) als Datenbasis genutzt. Der Ausbreitungsprozess selbst erfolgt analog zu jenem aus Abschnitt 2.2.3. In Algorithmus 5 werden einerseits die koordinierte Ausbreitung der Suchbäume (Zeilen 3 und 4) und andererseits das Abbruchkriterium (Zeile 5) dargestellt: Pro Schleifendurchlauf (Zeilen 2 bis 5) wird jeder der Suchbäume – $ST(G)$ und $ST(G')$ – um ein Basiselement erweitert. Abgebrochen wird die Ausbreitung sobald ein Basiselement β_i und dessen korrespondierendes inverses Element β'_i existiert, das einen fixierten Labelwert aufweist und somit Teil der Menge P beziehungsweise P' ist (vgl. Zeile 5).

Algorithmus 5 Bidirektionale Suchbaumausbreitung.

- 1: Initialisierung
 - 2: **repeat**
 - 3: Ausbreitung von $ST(G)$
 - 4: Ausbreitung von $ST(G')$
 - 5: **until** $\beta_i \in P \wedge \beta'_i \in P'$
-

Die sich daraus ergebende Eigenschaft einer optimalen Lösung lässt sich zum einen im Ausbreitungsverfahren – Algorithmus nach *Dijkstra*, der einen optimal kürzesten Weg liefert – und zum anderen im eingesetzten Abbruchkriterium begründen. Der Algorithmus nach *Dijkstra* liefert für den bidirektionalen Fall hier zwei Teilrouten:

1. Teilroute A vom Start der Gesamtroute β_S bis zu jenem Element β_i , das als erstes gleichzeitig in den Mengen P und P' vorkommt: $route_A = \{\beta_S, \dots, \beta_i\}$.
2. Teilroute B vom Ziel der Gesamtroute $\beta_D \equiv \beta'_S$ bis zu jenem Element β'_i , das als erstes gleichzeitig in den Mengen P und P' vorkommt: $route'_B = \{\beta'_S, \dots, \beta'_i\}$ bzw. $route_B = \{\beta_i, \dots, \beta_D\}$ nach der Reinvertierung.

Bei der Verknüpfung der beiden Teilrouten $route_A$ und $route_B$ kombiniert man einen optimal kürzesten Weg von β_S nach β_i mit dem anderen optimal kürzesten Weg von β_i nach

β_D . Daraus ergibt sich ein optimal kürzester Weg von β_S über das Verknüpfungselement β_i nach β_D : $route = \{\beta_S, \dots, \beta_i, \dots, \beta_D\}$.

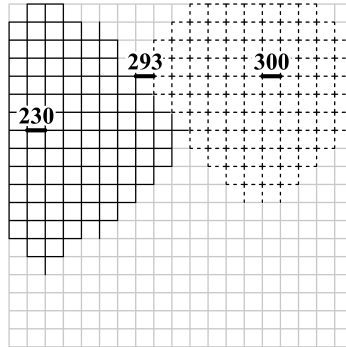


Abbildung 5.4: Abbruchkriterium bei bidirektionaler Suchbaumausbreitung – Überlagerung des Startsuchbaumes um Kante 230 und des Zielsuchbaumes um Kante 300 bei Kante 293.

In Abbildung 5.4 sind diese drei wesentlichen Elemente explizit dargestellt. Die Kante 230, Startelement der gesuchten Route β_S und gleichzeitig Startelement des ersten Suchbaumes $\beta_{SST(G)}$ (in der Abbildung als durchgezogene schwarze Linie dargestellt), und die Kante 300, Zielelement der gesuchten Route β_D und gleichzeitig Startelement des zweiten Suchbaumes $\beta_{SST(G')}$ (in der Abbildung als schwarz-strichlierte Linie dargestellt) bilden die Vorgaben der bidirektionalen Suchbaumausbreitung. Bei Kante 293 handelt es sich um jene Kante, die als erstes in die Mengen P und P' aufgenommen wurde und somit das Abbruchkriterium auslöste.

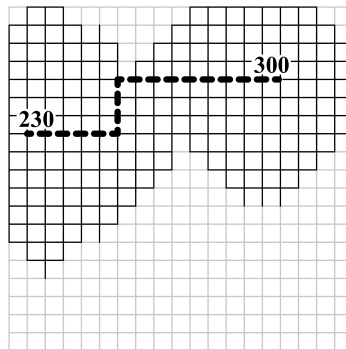


Abbildung 5.5: Bidirektionale Suchbaumausbreitung vom Startelement 230 zum Zielelement 300.

Die kombinierte Route aus den beiden Teilrouten zwischen Kante 230 und Kante 293 beziehungsweise zwischen Kante 300 und Kante 293 wird in Abbildung 5.5 als schwarz-strichlierte Linie angeführt. Ergänzt wird diese Abbildung durch die beiden Suchbäume als durchgezogene schwarze Linien ohne zwischen Start- und Zielelementsuchbaum zu unter-

scheiden.

5.3 Hierarchische Suchbaumausbreitung

Eine weitere Variante zur Ausbreitung von Suchbäumen nutzt hierarchische Eigenschaften der Datenbasis, um ein Start- mit einem Zielelement zu verbinden. Eines der offensichtlichsten Beispiele hierfür gibt der Straßenverkehr: Das Spektrum an Hierarchien reicht von einer Nebenstraße bis hin zur Autobahn. Eng mit dieser straßenverkehrsrechtlichen Einteilung ist die darauf erlaubte Höchstgeschwindigkeit verknüpft. Und je schneller sich ein Fahrzeug bewegen darf, umso kürzer ist die Zeitspanne zwischen Abfahrt und Ankunft, die für die Zufriedenheit des Autofahrers sorgt.

Das technische Ziel einer hierarchischen Suchbaumausbreitung besteht in einer stufenweisen Einschränkung der zu analysierenden Datenmenge (vgl. [FSR06], [JSQ02] oder [CTB01]). Je höher die hierarchische Ebene einzustufen ist, der die einzelnen Basisdaten zugeordnet sind, umso geringer ist im Allgemeinen deren Anzahl. Hierfür sind in Tabelle 5.1 die Straßenlängen in Kilometer und die Anzahl der damit verbundenen Liniengeometrien der jeweiligen Hierarchie, Functional Road Class (FRC), nach [Tel07] für Österreich angeführt (Datengrundlage: Tele Atlas, 2008).

Tabelle 5.1: Statistische Kennwerte des österreichischen Straßennetzes nach Tele Atlas in hierarchisch absteigender Reihung (Stand 2008).

FRC	FRC Beschreibung	Länge	Anzahl d. Elemente
0	Motorway, Freeway	3 882 km	10 088 Stk.
1	Major Road	1 695 km	7 992 Stk.
2	Other Major Road	7 551 km	50 107 Stk.
3	Secondary Road	6 106 km	37 293 Stk.
4	Local Connecting Road	35 898 km	206 331 Stk.
5	Local Road of High Importance	1 426 km	13 403 Stk.
6	Local Road	17 205 km	117 083 Stk.
7	Local Road of Minor Importance	136 326 km	578 164 Stk.
8	Other Roads	1 175 km	12 077 Stk.
-1	Not Applicable	75 km	1 428 Stk.

Dieser technischen Vorgehensweise liegt ein mentaler Prozess zugrunde. Wenn man sich im Geiste eine Fahrt von Graz nach Wien überlegt, wird man sich um den Ausgangs- und Endpunkt der Fahrt mit Details des Straßennetzes auseinandersetzen. Dazwischen – auf der Autobahn – treten Details mehr und mehr in den Hintergrund. [Zha97, Abschnitt 5.5] beschreibt diese Vorgehensweise als zweistufiges *top-down* Verfahren, Bidirectional Heuristic-based Hierarchical Search (BHHS), wobei als erstes eine Problemlösung im *abstraction space* erfolgt – eine Datengrundlage mit stark vereinfachter Situationsdarstellung. Darauf aufbauend wird eine Detaillösung erarbeitet. Bei [TEF97] werden drei Ebenen eingesetzt, um den mentalen Prozess zu beschreiben: Eine Planungsebene im Kleinmaßstäblichen, eine Instruktionsebene im mittleren Maßstab und eine Fahrerebene im Großmaßstäblichen.

In [Car97] werden unter der Bezeichnung Hierarchical Spatial Reasoning (HSR) für hierarchische Suchbaumausbreitungen die menschlichen Erfahrungen und das Wissen über den

Umgang mit Hierarchien mit bekannten Routing-Algorithmen kombiniert. Innerhalb einer Hierarchieebene wird der Suchbaum nach *Dijkstra* ausgebreitet, Non-Hierarchical Wayfinding Algorithm (NHWA). Die Berücksichtigung der Hierarchien bei einem Hierarchical Wayfinding Algorithm (HWA) erfolgt durch ein zusätzliches Regelwerk für die Hierarchieübergänge, um das der NHWA erweitert wird. Das Abbruchkriterium des NHWA wird zum einen bei Erreichung des Zieles oder zum anderen bei Zugang zu einem höheren Level erfüllt (*bottom-up* Verfahren). Wurde das zweite Kriterium erfüllt, so wird anschließend ein neuer Suchbaum innerhalb der höheren Hierarchie ausgebreitet (vgl. [CTB01]).

Ein vergleichbares *bottom-up* Verfahren wird bei [HLW03, Abschnitt 14.2.2] bzw. [SW93] angeführt, wobei die Verbindung zwischen den verschiedenen hierarchischen Ebenen durch Hinzufügen einer zusätzlichen künstlichen Kante in die bestehenden Knoten-Kanten Struktur erfolgt. Der beschriebene Level Graph Search (LGS) beschränkt sich prinzipiell auf zwei Ebenen, eine Erweiterung auf beliebig viele Ebenen ist jedoch laut [Stü93, S. 28] realisierbar.

Mit Hilfe des Heuristic Hierarchical Path Search (HHPS), beschrieben bei [HJR97], werden zeitliche Variationen der Basisdaten in den Routing-Prozess mit einbezogen. Sollte eine Verbindung innerhalb einer höheren Hierarchie durch ein unvorhergesehenes Ereignis (z.B.: Unfall, Baustelle usw.) blockiert oder der Verkehrsfluss stark beeinträchtigt sein, so wird hier dem Fahrer eine alternative Route angeboten. Diese verläuft zwar auf einer niedrigeren hierarchischen Ebene, aber das Hindernis in der höheren hierarchischen Ebene kann somit umfahren werden. Für diese Art von Routing-Verfahren, die auf Daten in Echtzeit bzw. auf zeitnahe Information angewiesen sind (z.B.: Traffic Message Channel (TMC)), stellen intelligente Transportsysteme ITS die notwendige Funktionalität, Kommunikationsnetze und andere Infrastruktureinrichtungen bereit.

Im Kontext der in [CF94] angeführten Überprüfung der Korrektheit und Effizienz einer hierarchischen Route im Vergleich zu einem Ergebnis eines nicht-hierarchischen Verfahrens ist die Arbeit von [JSQ02] zu sehen. Das Wesen des hierarchischen Suchbaumausbreitens ist meist durch das Streben, eine höhere hierarchische Ebene zu erreichen, charakterisiert. Jedoch liefert dieses zwingende Ausnutzen höherer Hierarchien nicht immer ein sinnvolles Resultat. Als problematisch sind jene Situationen zu sehen, bei denen Start- und Zielelement durch ein quer verlaufendes Element höherer Hierarchie voneinander getrennt werden (vgl. *Route A* in Tabelle 5.5). [JSQ02] beschreibt eine Methode, in der dieses Element höherer Hierarchie vom Suchbaum „überquert“ werden kann. Hierfür sind die benachbarten Wegenetze um das Start- und Zielelement zu vereinen und für eine anschließende Suchbaumausbreitung alleine zu nutzen.

Für die Umsetzung eines hierarchischen Routing-Algorithmus ist nach [CF94] die Auseinandersetzung mit nachfolgenden drei Punkten ein wesentlicher Bestandteil der Konzeption:

- Einteilung der Basisdaten in unterschiedliche Hierarchien. Je nach Aufgabenstellung sind Hierarchiezuordnungen der einzelnen Basiselemente zu restrukturieren. Die hierarchische Aufteilung eines Straßennetzes unterscheidet sich beispielsweise zwischen einem Radfahrer und einem Autofahrer, der die berechnete Route befahren soll.
- Wechsel zwischen den einzelnen Hierarchien. Innerhalb der Datenbasis ist zu regeln, wie der Suchbaum während des Ausbreitungsprozesses von einer hierarchischen Ebene in eine andere hierarchische Ebene wechseln kann. Ein Beispiel hierfür liefert [HLW03] mit einem zusätzlichen künstlichen Verbindungselement.
- Steuerung der Suchbaumausbreitung. Hier muss der Algorithmus entscheiden, ab wann ein Wechsel in eine höhere hierarchische Ebene für die konkrete Aufgabenstellung als

günstig zu erachten ist. Spezielle Aufgaben, wie sie bei [JSQ02] beschrieben werden, können besser gelöst werden, wenn auf den Wechsel in eine höhere Hierarchie verzichtet wird.

Zum zuvor angeführten Punkt über die Einteilung der Basisdaten finden sich Geo-Datenquellen, die bereits eine hierarchische Gliederung aufweisen. Ein kommerzielles Produkt der Firma Tele Atlas beinhaltet die Hierarchieinformation in der Eigenschaft FRC innerhalb des *Network* (*_ nw*) Datensatzes (vgl. [Tel07]), der hier als Grundlage der Tabelle 5.1 dient. In OpenStreetMap, ein Vertreter freier Geo-Daten, werden die einzelnen *Way* Elemente durch ihre jeweilige *highway* Eigenschaft kategorisiert. Diese Art der Hierarchieinformation wird beispielsweise bei [Liu96, Abschnitt 3.1] unter der Bezeichnung *road types* zur hierarchischen Suchbaumausbreitung genutzt.

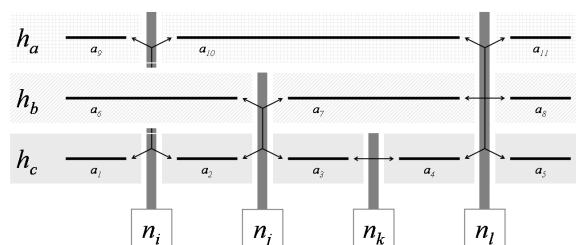


Abbildung 5.6: Querschnitt eines hierarchisch strukturierten Datensatzes.

Den beispielhaften Querschnitt einer hierarchischen Datenstruktur zeigt Abbildung 5.6 auf. Hier werden die Kanten a_1 bis a_{11} (z.B.: Straßen) drei verschiedenen Hierarchien zugeordnet: h_a als höchste, h_b als mittlere und h_c als niedrigste Ebene. Weiters sind Knoten (n_i , n_j , n_k und n_l) angeführt, die beispielsweise Kreuzungen repräsentieren. Ohne spezielle Kennzeichnung sind schemenhaft die Traversen zwischen den einzelnen Kanten durch Pfeile angedeutet. Je nach verwendeter Datenstruktur – Knoten-Kanten oder Kanten-Traversen Strukturen – finden diese Eingänge in den Routing-Algorithmus.

Aus Abbildung 5.6 entsteht die Frage, welche Elemente einer bestimmten Hierarchie zuzuordnen sind. Die dargestellten Kanten a können eindeutig einer hierarchischen Ebene zugewiesen werden. Bei den Knoten n und Traversen t hingegen ist diese Zuweisung mehrdeutig. Beispielsweise finden am Knoten n_i Kanten aus den Hierarchien h_a (a_1 und a_2) und h_c (a_9 und a_{10}) zusammen (vgl. *mixed junctions* bei [Liu96]). Das gleiche gilt für die Traversen, die jene Verbindungen zwischen den angeführten Kanten realisieren.

Wird nun für die Suchbaumausbreitung eine Knoten-Kanten Struktur gemäß Abschnitt 3.1 als Datengrundlage gewählt, so sollte das Basiselement – der Knoten – einer bestimmten Hierarchie zuzuordnen sein. Auswirkungen hat diese Zuordnung auf Bewertungsverfahren, die sich mit der Kreuzungssituation befassen: Überquert ein Fahrzeug die Kreuzung auf der höchsten Hierarchieebene, biegt das Fahrzeug von einer höheren in eine niedrigere hierarchische Ebene ab oder fährt das Fahrzeug von einer niedrigen in eine höhere Ebene auf. All diese Situationen sind in hierarchischen Verfahren in einer Kreuzung zu beachten. Datenmodelle beinhalten beispielsweise für eine Kreuzung einen Knoten pro Hierarchie (vgl. [HLW03, Abschnitt 14.2.2] oder [Car97, Abschnitt 7.3.3]), oder die Knoten werden jener

Hierarchie zugewiesen, die als höchste an der Kreuzung vorkommt (siehe u.a. [HJR97, Abschnitt 3.2.1]).

Bei Kanten-Traversen Modellen sind die Basiselemente – die Kanten – eindeutig einer Hierarchie zuweisbar. Die hierarchische Mehrdeutigkeit des Verbindungselementes, sobald dieses Kanten verschiedener Ebenen miteinander verbindet, muss in dem hier verwendeten Algorithmus nicht berücksichtigt werden. Die Erweiterung des Basisdatensatzes ist in Tabelle 5.2 für eine Kanten-Traversen Struktur dargestellt, in der die Hierarchie der jeweiligen Kante als $^h a$ ausgewiesen wird: Je höher die hierarchische Zugehörigkeit, desto kleiner der numerische Wert (vgl. Tabelle 5.3). Die restlichen Eigenschaften des Basiselementes, wie auch die Verbindungselementerelation bleiben unverändert im Vergleich mit der nicht-hierarchischen Modellierung aus Abschnitt 3.2.

Tabelle 5.2: Relationale Darstellung der Eigenschaften von Basiselementen in einer hierarchisch strukturierten Kanten-Traversen Struktur.

Id	[PS]	a_i
Kosten		$c a_i$
Hierarchie		$^h a_i$
Labelwert		$l a_i$
Vorgänger	[FS]	$p a_i$
Geometrie		$\text{LINESTRING}(x_1 a_i \ y_1 a_i, x_2 a_i \ y_2 a_i, \dots, x_n a_i \ y_n a_i)$

Das Ergebnis einer visuellen Interpretation der Hierarchieeigenschaft einzelner Basiselemente ist in Abbildung 5.7 ersichtlich. Die bereits in den Abschnitten 5.1 und 5.2 für nicht-hierarchische Suchbaumausbreitungen verwendete Datenstruktur wurde für eine hierarchische Nutzung aufbereitet. Die graphische Darstellung der hierarchischen Zugehörigkeit ist durch die Linienstärke des jeweiligen Basiselementes charakterisiert: Je dicker eine Kante dargestellt wird, umso höher ist deren hierarchische Zuordnung.

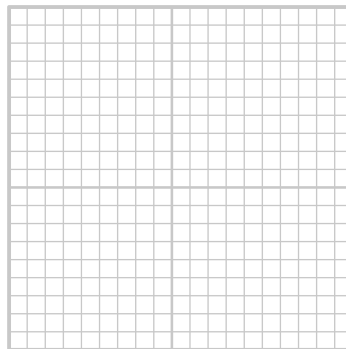

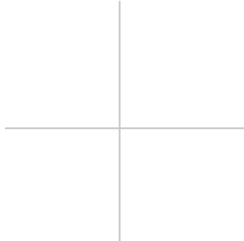
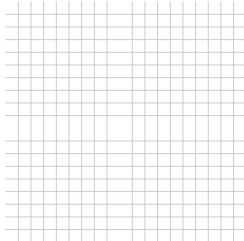


Abbildung 5.7: Ausgangsdaten – Kanten einer Kanten-Traversen Struktur mit hierarchischer Zusatzinformation.

Diese hierarchische Zuordnung wird in Tabelle 5.3 im Detail aufgeschlüsselt. Jene Menge an Basiselementen, die der höchsten hierarchischen Ebene zugewiesen wurde ($^h B = 1$),

bildet ein den restlichen Datensatz umschließendes Rechteck. Die zweithöchste Hierarchie innerhalb der Basisdaten (${}^hB = 2$) weist die Form zweier sich schneidenden Geraden auf. Die letzte und niedrigste hierarchisch attributierte Menge an Basisdaten (${}^hB = 3$) umfasst die übrigen Kanten.

Tabelle 5.3: Aufbau der Ausgangsdaten.

Höchste Hierarchie ${}^hB = 1$	Mittlere Hierarchie ${}^hB = 2$	Niedrigste Hierarchie ${}^hB = 3$
		

Dieser hierarchisch strukturierte Datensatz dient einer adaptierten bidirektionalen Suchbaumausbreitung als Grundlage. Die in Abschnitt 5.2 vorgestellte Methode wird zu einem *bottom-up* Suchbaumausbreitungsverfahren erweitert. Im Vergleich mit einer monodirektionalen Suchbaumausbreitungsmethode gemäß Abschnitt 5.1, bei der nach einem Hierarchieaufstieg im Bereich des Startelementes ein regelbasierter Hierarchieabstieg – bezogen auf den Zeitpunkt und die Weise, in der dieser Abstieg durchzuführen ist – in der Nähe des Zieles erfolgen muss, ist ein bidirektionales Verfahren hinsichtlich des Hierarchieabstieges im Zielbereich als ein praktikabler Ansatz zu sehen.

Der erste Suchbaum um das Startelement $ST(G)$ wird so gesteuert, dass dieser bei Erreichen einer höheren Hierarchie sich nur mehr in dieser weiter ausbreitet. Ein Abstieg in eine niedrigere hierarchische Ebene ist in diesem Abschnitt nicht vorgesehen. Genauso wie der erste Suchbaum wird jener Suchbaum um das Zielelement $ST(G')$ erweitert: Wird eine höhere Hierarchie erreicht, so ist der Suchbaum nur mehr in dieser weiter auszubreiten. Durch diesen bidirektionalen Ansatz entgeht man der Fragestellung, ab wann der Suchbaum in der Hierarchie wieder abzustiegen hat.

Das sich daraus ergebende Verfahren ist in Algorithmus 6 aufgezeigt. Hier wird, verglichen mit Algorithmus 5 aus Abschnitt 5.2, eine Regel eingeführt, ab wann der jeweilige Suchbaum erweitert werden darf (vgl. Zeile 3 für die Suchbaumausbreitung um das Startelement und Zeile 6 für die Suchbaumausbreitung um das Zielelement). Dabei sind zwei Fälle des Ausbreitungsmodus zu unterscheiden:

1. *Synchrone* Ausbreitung der Suchbäume: $ST(G)$ und $ST(G')$ werden in einem Schleifendurchlauf (Zeilen 2 bis 9) um jeweils ein Basiselement erweitert. Dieser Modus erfolgt nur dann, wenn sich beide Suchbäume in derselben hierarchischen Ebene befinden: $\min({}^hP) \equiv \min({}^hP')$.
2. *A-Synchrone* Ausbreitung der Suchbäume: Es wird entweder $ST(G)$ oder $ST(G')$ innerhalb des Schleifendurchlaufes (Zeilen 2 bis 9) um ein Basiselement erweitert. Je

nachdem welcher der beiden Suchbäume sich in einer niedrigeren hierarchischen Ebene befindet (entspricht einem größeren Wert in ${}^h P$), wird nur dieser auch erweitert. Damit wird versucht, den Suchbaum der niedrigeren Ebene den Zugang zu einer höheren Ebene – jene des anderen Suchbaumes – zu ermöglichen, und den Suchbaum der höheren Hierarchie nicht unnötig auszubreiten.

Algorithmus 6 Hierarchische Suchbaumausbreitung.

```

1: Initialisierung
2: repeat
3:   if  $\min({}^h P) \geq \min({}^h P')$  then
4:     Ausbreitung von  $ST(\{G \mid {}^h G \leq \min({}^h P)\})$ 
5:   end if
6:   if  $\min({}^h P') \geq \min({}^h P)$  then
7:     Ausbreitung von  $ST(\{G' \mid {}^h G' \leq \min({}^h P')\})$ 
8:   end if
9: until  $\beta_i \in P \wedge \beta'_i \in P'$ 

```

Das Abbruchkriterium des Algorithmus in Zeile 9 bleibt im Vergleich mit dem nicht-hierarchischen Algorithmus unverändert. Hierbei ist es unbedeutend, in welcher hierarchischen Ebene sich ein Suchbaum befindet; sobald ein Basiselement in beiden Suchbäumen der Menge P bzw. P' angehört, wird die Ausbreitung beendet. Das hat zur Folge, dass, obwohl einer der beiden Suchbäume eine höhere Ebene erreichen konnte, die resultierende Route jedoch diese höhere Ebene nicht zwangsläufig nützen muss (vgl. Tabelle 5.4).

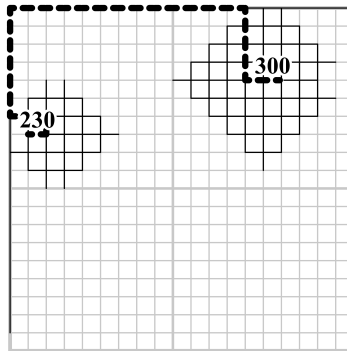
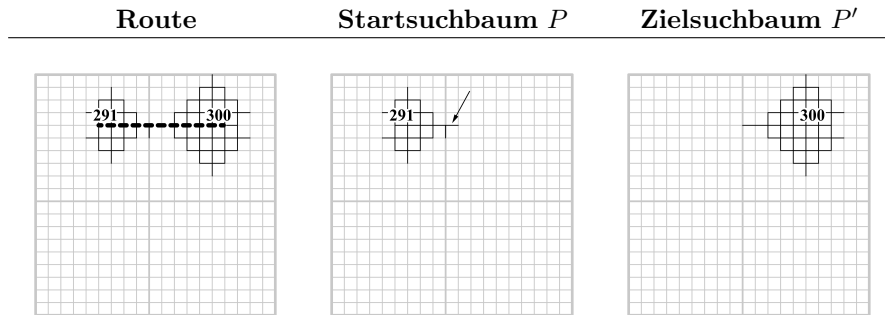


Abbildung 5.8: Hierarchische Suchbaumausbreitung vom Startelement 230 zum Zielelement 300.

Das Ergebnis einer hierarchischen Suchbaumausbreitung von der Startkante 230 zur Zielkante 300 ist in Abbildung 5.8 dargestellt. Die zu Grunde liegenden Daten entsprechen jenen aus Abbildung 5.7, wobei sich Start und Ziel jeweils in der niedrigsten Hierarchie (${}^h \beta_S = 3$ bzw. ${}^h \beta_D = 3$) befinden. Die beiden Suchbäume (P und P') werden hier als durchgezogene schwarze Linien visualisiert, die sich solange konzentrisch um Start- bzw. um Zielelement darstellen, bis sie Elemente der höchsten Hierarchie erreichen. Die ermittelte Route, als schwarz-strichlierte Linie abgebildet, beschreibt einerseits den kürzesten Weg vom Start-

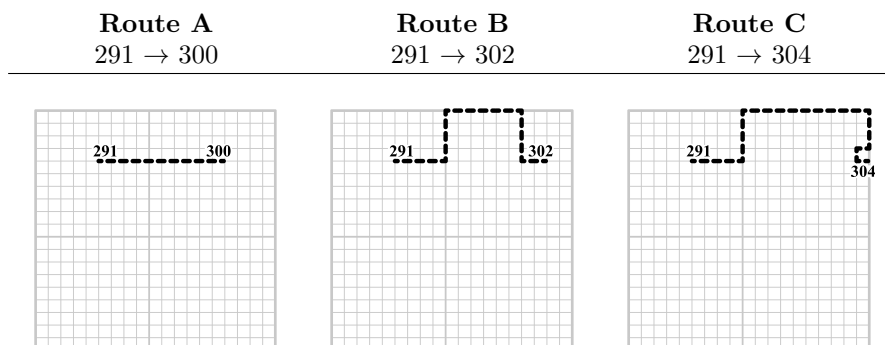
bzw. Zielelement in die höchste Hierarchie und andererseits die kürzeste Verbindung innerhalb der höchsten Hierarchie.

Tabelle 5.4: Hierarchische Suchbaumausbreitung vom Startelement 291 zum Zielelement 300 mit gesonderter Ausweisung einer Übersprungstelle eines Suchbaumes.



Eine spezielle Konstellation von Start und Ziel wird in Tabelle 5.4 detaillierter betrachtet: Die Suchbäume um Start- und Zielelement (Kante 291 und Kante 300) werden getrennt von einander dargestellt. Als ein wichtiges Element wird jene Kante mit einem Pfeil ausgewiesen, bei der es sich um eine Übersprungstelle handelt. Jene Thematik, die bei [JSQ02] eine zusätzliche Fallunterscheidung und anschließender Datenkombination bedarf, wird hier durch die Modellierung der Datengrundlage als Kanten-Traversen Struktur implizit behandelt. Die aus der Suchbaumausbreitung resultierende Route zwischen den Kanten 291 und 300 führt somit über einen Kantenzug höherer Hierarchie, jedoch ohne ihn zu nutzen.

Tabelle 5.5: Hierarchische Suchbaumausbreitung bei unterschiedlichem räumlichen Kontext.



Die Kontextsensitivität dieses Algorithmus wird in Tabelle 5.5 anhand dreier ausgewählter Situationen verdeutlicht. Generell wird hierbei die Entfernung zwischen Start- und Zielelement und die Entfernung zu Strukturen höherer Hierarchie für die Suchbaumausbreitung berücksichtigt: Als *Route A* wie die zuvor vorgestellte Route zwischen Kante 291 und Kante

300 angeführt. Die *Route B* verbindet die Kanten 291 und 302. In diesem Fall werden hierarchisch höher gereihte Kanten in das Ergebnis miteinbezogen. Ebenfalls werden bei *Route C* zwischen den Kanten 291 und 304 hierarchische Strukturen berücksichtigt. Auf Grund der räumlichen Nähe zu hierarchisch höheren Elementen bei Kante 304 erfolgt der Übertritt hier früher als bei *Route B*. Anm.: Die Schleife bei Kante 304 ergibt sich durch die Verwendung von gerichteten Kanten und das Fehlen von Traversen, die gegengleich orientierte Basiselemente entlang der selben Straße miteinander verbinden.

5.4 Multi Level Hopping

Die hierarchische Aufbereitung von Routing-Algorithmen vor dem Hintergrund eines *bottom-up* Ansatzes ist durch das Bestreben charakterisiert, eine immer höhere hierarchische Ebene erreichen zu wollen. Dazu passend beschreibt unter anderem [JSQ02, Kapitel III] ein zwei-stufiges Verfahren, das von einem niedrigeren Verkehrsnetz ausgehend – *O-grid* um den Ausgangspunkt der Route und *D-grid* um das Ziel der Route – Einstiegselemente (*E-nodes*) in ein höheres Netz sucht. Sobald der Einstieg ins höhere Netz erfolgt ist, kann der Suchbaum in diesem ausgebreitet werden. Auch bei [CTB01, Abschnitt 3.2] wird ein Verfahren vorgestellt, dass bei Erreichen einer nächst höheren Hierarchieebene den Graphen bis auf jene Elemente ausdünnert, die entweder der aktuell erreichten Ebene oder einer höheren Ebene angehören.

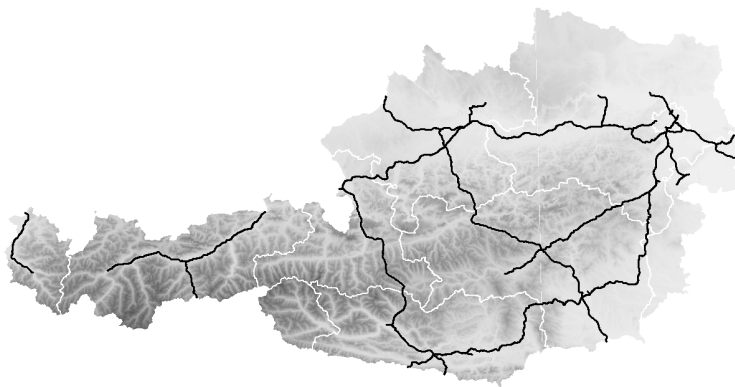


Abbildung 5.9: Österreichisches Autobahn- und Schnellstraßennetz. Datengrundlage: Tele Atlas Straßennetz (`__nw`) beschränkt auf $FRC = 0$ – Stand 2008; Tele Atlas Administrative Grenzen (`__a0`) – Stand 2008; NASA SRTM-3 – Stand 2000.

All diese Verfahren gehen davon aus, dass die Elemente der höheren Hierarchieebenen mit

einander verbunden sind. Betrachtet man das in Abbildung 5.9 dargestellte österreichische Autobahn- und Schnellstraßennetz, so ist zu erkennen, dass dieses nicht in seiner Gesamtheit verbunden ist. Bis zu einem gewissen Anteil verhindert die Topographie Österreichs einen Zusammenschluss der in Vorarlberg und Tirol gelegenen Abschnitte mit jenen Teilen Mittel- und Ostösterreichs.

Als thematisch affin zu der Problematik der nicht zusammenhängenden Elemente höherer Hierarchieebenen ist eine Arbeit von [Liu97] zu sehen. Darin wird beschrieben, wie Abkürzungen – Straßen niedrigerer Hierarchie, die bei Benutzung eine wesentliche Ersparnis an Weg oder Zeit im Vergleich zu Straßen höherer Hierarchie liefern – innerhalb des Basisdatensatzes modelliert werden können. Es werden *pseudo major roads* definiert: Straßen, die von ihrer hierarchischen Stellung einer niedrigeren Ebene angehören, aber durch die positive Auswirkung bei ihrer Benutzung einer höheren Ebene zugeteilt werden. Auf das österreichische Straßennetz bezogen würde es eine Aufwertung der Verbindungsstraßen zwischen den Autobahn- und Schnellstraßenabschnitten in Vorarlberg, Tirol und Salzburg bedeuten.

Was mit diesem Ansatz jedoch nicht geregelt wird, ist der hierarchische Abstieg des Suchbaumes im Zuge seiner Ausbreitung. Im Gegensatz dazu werden Basiselemente einer höheren Hierarchie zugeordnet, der sie eigentlich nicht angehören. Für die Entwicklung eines Algorithmus, der hierarchische Auf- und Abstiege während der Suchbaumausbreitung bewältigen kann, sind nachfolgende Fragen zu berücksichtigen:

- *Wann soll die Ausbreitung des Suchbaumes in einer niedrigeren hierarchischen Ebene fortgesetzt werden?* Sobald keine weiteren Basiselemente gleicher oder höherer Hierarchie dem Ausbreitungsprozess zur Verfügung stehen, muss der Suchbaum hierarchisch niedriger eingestufte Elemente zur Ausbreitung nutzen.
- *Wo soll sich der Suchbaum weiter ausbreiten?* Anknüpfungselemente für eine Fortsetzung der Suchbaumausbreitung innerhalb einer niedrigeren Hierarchieebene bilden Level Exit Arcs LEA. Dabei handelt es sich um Elemente mit temporärem Label, die einer niedrigeren Hierarchie angehören und entlang der höheren Hierarchie abzweigen, in der sich der Suchbaum aktuell ausbreitet (vgl. *mixed junctions* bei [Liu96]).
- *In welcher hierarchischen Ebene soll der Suchbaum weiter ausgebreitet werden?* Für eine Weiterführung der Suchbaumausbreitung werden LEA Elemente jener hierarchischen Ebene verwendet, die der nächst niedrigeren Hierarchie zugeordnet sind – unabhängig ihrer räumlichen Lage.

Für eine algorithmische Umsetzung einer Suchbaumausbreitung mit hierarchischer Auf- und Abstiegsfunktionalität wird der hierarchische Algorithmus 6 aus Abschnitt 5.3 hinsichtlich der zuvor angeführten Fragestellungen adaptiert und erweitert. Daraus ergibt sich der Multi Level Hopping (MLH) Algorithmus. Der Unterschied zwischen Algorithmus 7 – MLH – und Algorithmus 6 – hierarchisches *bottom-up* Verfahren – ist in der Auswahl der Basisdaten zu sehen, die für eine Suchbaumausbreitung verwendet werden: War es bei Algorithmus 6 die Menge $\{G \mid {}^hG \leq \min({}^hP)\}$, so werden diese bei MLH auch hinsichtlich ihrer hierarchischen Zuordnung eingeschränkt, aber diese Einschränkung bezieht sich nicht unmittelbar auf die vom Suchbaum erreichte höchste Hierarchieebene $\min({}^hP)$, sondern wird individuell über einen Hierarchiewert h gesteuert: $\{G \mid {}^hG \leq h\}$.

Die eigentliche Suchbaumausbreitung ist so zu gestalten, dass nur jene Elemente der Menge T für die Überführung in die Menge P in Frage kommen, die der Bedingung ${}^h\beta_{T_i} \leq h$

entsprechen. Um zusätzlich LEA Elemente zu erhalten, werden für die Überführung von Elementen der Menge U in die Menge T keine Hierarchiebedingungen verwendet.

Ab wann und in welche Ebene ein hierarchischer Abstieg im Verlauf der Suchbaumausbreitung zu erfolgen hat, wird im Algorithmus 7 für den Startsuchbaum in den Zeilen 7 bis 10 und für den Zielsuchbaum in den Zeilen 14 bis 17 geregelt. In diesen Abschnitten wird anfänglich die höchste Hierarchie des aktuellen Suchbaumes dem Hierarchiewert h zugewiesen. Nachfolgend wird überprüft, ob Elemente der Menge T existieren, die eine Fortsetzung des Suchbaumes in der aktuellen Hierarchieebene erlauben. Werden keine passenden Elemente hierfür gefunden, so hat ein Abstieg in jene Hierarchie zu erfolgen, die als höchste in der Menge T zur Verfügung steht: $\min({}^hT)$.

Algorithmus 7 Multi Level Hopping.

```

1: Initialisierung
2:  $h := {}^h\beta_S$ 
3:  $h' := {}^h\beta_D$ 
4: repeat
5:   if  $h \geq h'$  then
6:     Ausbreitung von  $ST(\{G \mid {}^hG \leq h\})$ 
7:      $h := \min({}^hP)$ 
8:     if  $\{T \mid {}^hT \leq h\} = \{\}$  then
9:        $h := \min({}^hT)$ 
10:    end if
11:  end if
12:  if  $h' \geq h$  then
13:    Ausbreitung von  $ST(\{G' \mid {}^hG' \leq h'\})$ 
14:     $h' := \min({}^hP')$ 
15:    if  $\{T' \mid {}^hT' \leq h'\} = \{\}$  then
16:       $h' := \min({}^hT')$ 
17:    end if
18:  end if
19: until  $\beta_i \in P \wedge \beta'_i \in P'$ 

```

Eine sukzessive Ausdünnung der Datengrundlage, wie sie bei [Car97] beschrieben wird, ist bei dieser Variante eines hierarchischen Suchbaumausbreitungsverfahrens wegen der benötigten LEA Elemente nur bedingt umsetzbar. Die Verringerung der Datenmenge darf nicht alleine auf der hierarchischen Zuordnung eines Elementes geschehen, sondern es müssen alle benachbarten Elemente niedrigerer Hierarchie bestehen bleiben. Weiters, konträr zum Ausdünnungsprozess, hat bei einem Hierarchieabstieg die Verdichtung der Datengrundlage zu erfolgen.

Das Wesen und Wirken des MLH Algorithmus wird für ein einfaches SSSDP zwischen den Kanten 175 und 186 aufbereitet, wobei als Datengrundlage die bereits in Abschnitt 5.3, Abbildung 5.7 vorgestellte Kanten-Traversen Struktur dient. Das Ergebnis dieser Routenberechnung ist in Abbildung 5.10 als schwarz-strichlierte Linie dargestellt und befindet sich zur Gänze in der zweiten Hierarchieebene. Den Unterschied zwischen einem gewöhnlichen *bottom-up* Verfahren und dem MLH Algorithmus zeigt Tabelle 5.6 auf, in der die einzelnen relevanten Teilmengen – P und T bzw. P' und T' – der Basisdaten aus den verschiedenen Algorithmen gegenübergestellt werden. Hierbei ist darauf zu achten, dass es sich bei der Da-

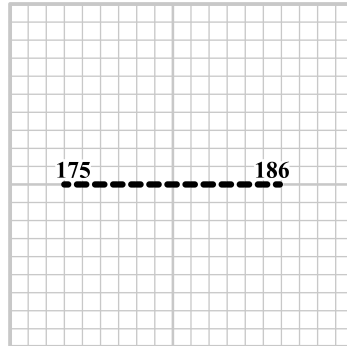


Abbildung 5.10: Route zwischen Kante 175 und Kante 186 zur Erläuterung der *Level Exit Arcs*.

tengrundlage um einen gerichteten Graph handelt. Start- und Zielkante sind von links nach rechts ausgerichtet, das die einseitige Ausbreitung des Suchbaumes erklärt: Die Ausbreitung des Startsuchbaumes in Richtung der Kantenorientierung und jene des Zielsuchbaumes gegen die Richtung der Kantenorientierung.

Tabelle 5.6: Gegenüberstellung der permanenten und temporären Kantenmengen der Start- und Zielsuchbäume bei der Routenberechnung zwischen Kante 175 und Kante 186.

P <i>bottom-up</i> u. MLH	T <i>bottom-up</i>	T MLH
 P' <i>bottom-up</i> u. MLH	 T' <i>bottom-up</i>	 T' MLH
 P' <i>bottom-up</i> u. MLH	 T' <i>bottom-up</i>	 T' MLH

In der ersten Spalte der Tabelle 5.6 sind die jeweiligen Mengen P bzw. P' abgebildet, die sich bei einem *bottom-up* Verfahren nicht von einem MLH Verfahren unterscheiden. Der Start- bzw. Zielsuchbaum, der der Mengen P bzw. P' entspricht, verbleibt über seine gesamte Länge in der mittleren Hierarchieebene.

Unterschiede zwischen *bottom-up* und MLH Verfahren sind bei jenen Basiselementen erkennbar, die über einen vorübergehend zugewiesenen Labelwert verfügen: T . Da sich im Fall des *bottom-up* Verfahrens die Suchbaumausbreitung – Überführung von Elementen der Menge T in die Menge P bzw. von Elementen der Menge U in die Menge T – auf Basiselemente gleicher oder höherer Hierarchie beschränkt, lassen sich Elemente der Menge T ausschließlich innerhalb der mittleren Hierarchie finden.

Im Fall des MLH Verfahrens, das eine hierarchische Einschränkung bei der Überführung von Elementen der Menge U in die Menge T nicht kennt, sind zum einen die selben Kanten im Vergleich zum *bottom-up* Verfahren ersichtlich. Zum anderen können weitere Kanten erkannt werden, die der niedrigsten Hierarchie angehören und in direkter Nachbarschaft zu den Elementen der Menge P stehen. Bei diesen Kanten handelt es sich um die zuvor erwähnten LEA, die als Anknüpfungselemente bei Hierarchieabstiegen des Suchbaumes dienen.

Um die Notwendigkeit der LEA Elemente zu verdeutlichen, wird die hierarchische Datenbasis aus Abschnitt 5.3, Abbildung 5.7 dahingehend eingeschränkt, dass der Zusammenhang der höchsten Hierarchieebene nicht mehr gegeben ist. Die Einschränkung betrifft all jene Basiselemente, die sich außerhalb eines definierten Bereiches befinden. In Abbildung 5.11 wird die Situation so dargestellt: Der äußere Rand des definierten Bereiches wird durch eine dünne schwarze Linie visualisiert. Jene Kanten, die für eine Suchbaumausbreitung zur Verfügung stehen, befinden sich innerhalb dieses Bereiches und werden in einem dunklen Grau dargestellt. Alle anderen Kanten, in einem hellen Grau gezeichnet, werden im Zuge der Suchbaumausbreitung nicht berücksichtigt.

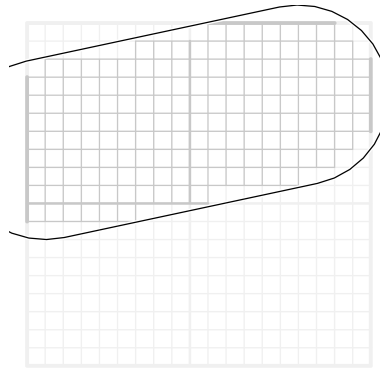


Abbildung 5.11: Ausgangsdaten – Kanten einer Kanten-Traversen Struktur mit hierarchischer Zusatzinformation. Einschränkung der Daten auf einen definierten Bereich, wodurch eine Verbindung der Elemente innerhalb der höchsten Hierarchieebene nicht mehr gegeben ist.

Das Ergebnis der MLH Routenberechnung zwischen der Startkante 230 und der Zielkante 300 wird in Abbildung 5.12 als schwarz-strichlierte Linie dargestellt. Obwohl Start- und Zielsuchbaum, als durchgezogene schwarze Linien erkennbar, die höchste Hierarchie erreichen – links der Startkante 230 und oberhalb der Zielkante 300 – und sich darin ausbreiten, werden diese Kanten jedoch nicht für die ermittelte Lösung verwendet: Der Algorithmus erkennt, dass in der höchsten Hierarchieebene keine Verbindung zwischen den einzelnen Suchbäumen zustande kommt und weicht auf niedrigere Ebenen aus. Die Ergebnisroute besteht aus Kanten der niedrigsten und der mittleren Hierarchie.

Ein LEA Element kommt jedoch bei der ermittelten Route dieses Szenarios nicht zum Einsatz. Würde ein LEA Element verwendet werden, so müsste ein hierarchischer Abstieg der Route zu erkennen sein, der nicht durch den gewählten bidirektionalen Suchbaumausbreitungsansatz zu begründen ist. Hier sind nur zwei hierarchische Aufstiege – einer im Umfeld des Start- und der andere im Umfeld des Zielsuchbaumes – zu erkennen: Von der niedrigsten in die mittlere Hierarchieebene.

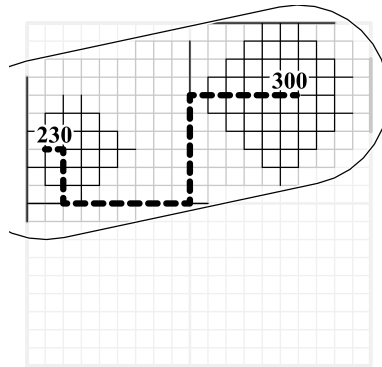


Abbildung 5.12: Multi Level Hopping Suchbaumausbreitung von Startelement 230 zum Ziel-element 300 ohne Verwendung eines LEA Elementes in der resultierenden Route.

Die Route zwischen den Kanten 1261 und 1380, dargestellt in Abbildung 5.13, beinhaltet zwei LEA Elemente (mittels Pfeile ausgewiesen). Die Datengrundlage der Routenberechnung entspricht hierbei ebenfalls jener aus Abschnitt 5.3, jedoch wird im Vergleich zu Abbildung 5.11 ein anderer Ausschnitt daraus verwendet – eine durchgehende Verbindung in der höchsten Hierarchieebene besteht dennoch nicht. In diesem Szenario sind Start- und Zielkante der höchsten hierarchischen Ebene zugeordnet. Nach Ausbreitung des Suchbaumes in dieser Ebene und der Erkenntnis, durch alleiniges Nutzen dieser oder höherer Hierarchien keine Lösung des SSSDP zu erhalten, muss ein hierarchischer Abstieg des Suchbaumes und in weiterer Folge der Route zugelassen werden. Nach einem Abstieg in die mittlere Ebene finden sich Start- und Zielsuchbaum und eine Route kann beschrieben werden.

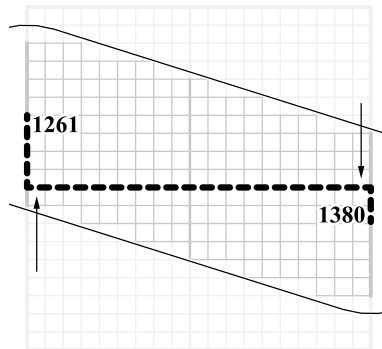


Abbildung 5.13: Multi Level Hopping Suchbaumausbreitung von Startelement 1261 zum Zielelement 1380 mit Verwendung zweier LEA Elemente in der resultierenden Route.

Kapitel 6

Anwendungskonzepte

Die grundlegenden Erkenntnisse und konzeptionellen Ansätze aus den vorangegangenen Kapiteln 3, 4 und 5 werden in diesem Kapitel in unterschiedlicher Weise kombiniert und verfeinert. Die hier entstehenden Anwendungsentwürfe sind großteils auf einer konzeptionellen Ebene angesiedelt, verstehen sich aber als notwendiger Unterbau für weiterführende Entwicklungen, die praxisorientierte Anwendungen bedienen. In der nachfolgenden Auflistung werden die wesentlichen Merkmale dreier Anwendungskonzepte dargestellt, die in den anschließenden Abschnitten im Detail erläutert werden:

1. Multimodale Routenplanung, Abschnitt 6.1: Berechnen kürzester Wege innerhalb einer Datenbasis bestehend aus einem Wegenetz für Fußgänger – eine Knoten-Kanten Struktur nach Abschnitt 3.1 – und einem zeitabhängigen Netz des öffentlichen Verkehrs – eine Kanten-Traversen Struktur nach Abschnitt 3.2. Diese Datenbasis wird als gemischt-einfache Struktur gemäß Abschnitt 3.3 realisiert und für eine monodirektionale Suchbaumausbreitung nach Abschnitt 5.1 aus einem Guss genutzt.
2. Short-Cut Routing, Abschnitt 6.2: Berechnung kürzester Wege innerhalb einer Datenbasis bestehend aus einem Wegenetz für Fußgänger – eine Kanten-Traversen Struktur nach Abschnitt 3.2 – und freiem Gelände – eine polygonale Struktur nach Abschnitt 3.4. Diese Datenbasis wird als gemischt-komplexe Struktur gemäß Abschnitt 3.6 unter zusätzlicher Definition von RTE realisiert und für eine monodirektionale Suchbaumausbreitung nach Abschnitt 5.1 aus einem Guss genutzt. Die Bezeichnung *Short-Cut Routing* begründet sich durch die entstehenden Wegeabkürzungen über das freie Gelände.
3. Cube Routing, Abschnitt 6.3: Berechnung kürzester Wege innerhalb eines 3-dimensionalen Datenbestandes – eine polyhedrale Struktur nach Abschnitt 3.5. Wegen der Nutzung eines Polyeder mit quadratischen Randflächen wird die Suche nach kürzesten Wegen durch diesen Datenbestand auch als *³routing* (sprich: *Cube Routing*) bezeichnet. Die Ausbreitung des Suchbaumes erfolgt monodirektional nach Abschnitt 5.1.

Zur programmatischen Umsetzung der Anwendungskonzepte wird für alle Szenarien ein *PostgreSQL* Datenbankmanagementsystem [The10b] mit der räumlichen Erweiterung *PostGIS* [Pos10] eingesetzt. Die Algorithmen wurden mittels *pl/pgSQL* [The10b, Kapitel 38] als Funktionen innerhalb des Datenbankmanagementsystems abgebildet. Die in den nachfolgenden Abschnitten gezeigten Graphiken wurden für 0-, 1- und 2-dimensionale Datengrundlagen durch den *MapServer* [The10a] erzeugt. Im Gegensatz dazu erfolgt die Darstellung

3-dimensionaler Inhalte durch eine *PHP-Skript* [ABD⁺10] automatisierte Erzeugung einer *X3D* Struktur [Int08], die von einem *BS Contact* Internetbrowser Plug-In [Bit10] interpretiert und visualisiert werden kann.

6.1 Multimodale Routenplanung

Die Lösung eines SPP mit spezieller Ausrichtung auf die kombinierte Nutzung zweier Verkehrsträger, wobei einer davon als zeitkritisch einzustufen ist, wird in diesem Abschnitt beschrieben. Im speziellen handelt es sich bei den zu nutzenden Verkehrsträgern zum einen um ein Wegenetz für Fußgänger und zum anderen um ein zeitabhängiges Liniennetz des öffentlichen Verkehrs. Innerhalb dieser Datenbasis soll ein optimal kürzester Weg nach Abschnitt 2.1.1 gefunden werden, wobei als Parameter einer monodirektionalen Suchbaumausbreitung (vgl. Abschnitt 5.1) der *Startort*, die *Startzeit* und der *Zielort* an den Algorithmus zu übergeben sind.

Ein operationelles System, das zum Teil vergleichbare Aufgabenstellungen bearbeitet, findet sich beispielsweise bei *AnachB.at* von [ITS10]. Jedoch unterscheidet sich der Einfluss der Startzeit bei *AnachB.at* zu der hier beschriebenen Lösung: Der hier entwickelte Algorithmus in Kombination mit der Datenbasis ist so ausgerichtet, dass bei zeitlich ungünstigem Erreichen einer Haltestelle ein Aufsuchen einer anderen Haltestelle derselben Linie – z.B. dem Bus entgegengehen – berücksichtigt wird.

6.1.1 Datengrundlage

Zur Lösung des multimodalen SPP werden zwei unterschiedliche Datenstrukturen in Kombination eingesetzt:

- Das Wegenetz für Fußgänger als zeitunabhängiges Knoten-Kanten Modell gemäß Abschnitt 3.1.
- Das Liniennetz des öffentlichen Verkehrs als zeitkritisches Kanten-Traversen Modell gemäß Abschnitt 3.2 unter Berücksichtigung der zeitlichen Variabilität der Elemente nach Abschnitt 4.3.

Diese beiden Datenstrukturen werden vor dem Hintergrund einer Suchbaumausbreitung aus einem Guss als gemischt-einfache Struktur (vgl. Abschnitt 3.3) für den Algorithmus vorgehalten. Der Übergang zwischen dem Wegenetz und dem zeitabhängigen Liniennetz – das Betreten und das Verlassen einer Haltestelle, wie in Abschnitt 4.3.3 beschrieben – erfolgt durch zusätzliche Verbindungselemente zwischen den Basiselementen beider Netze. Hierbei werden für das Betreten der Haltestelle ein Knoten des Wegenetzes n_i mit einer Kante des Liniennetzes a_j durch das Verbindungselement (n_i, a_j) , beziehungsweise für das Verlassen der Haltestelle eine Kante des Liniennetzes a_k mit einem Knoten des Wegenetzes n_l durch das Verbindungselement (a_k, n_l) in Beziehung gebracht.

Die Charakteristik einer gemischt-einfachen Struktur der Datengrundlage spiegelt sich in Abbildung 6.1 wider, in der ausschließlich die Basiselemente dargestellt werden. Der regelmäßige 9×5 Raster an Knoten – runde graue Punktsymbole – repräsentiert die Basiselemente der Knoten-Kanten Struktur und jene in grau dargestellten Linienelemente die Basiselemente der Kanten-Traversen Struktur. An den Stellen der Datengrundlage, an denen eine Haltestelle modelliert wurde, und ein Wechsel zwischen Wegenetz und Liniennetz

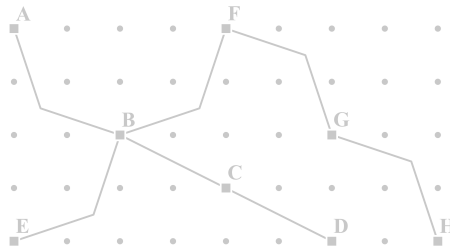


Abbildung 6.1: Basiselemente einer gemischt-einfachen Struktur bestehend aus Knoten des Wegenetzes für Fußgänger und Kanten des Liniennetzes des öffentlichen Verkehrs.

ermöglicht wird, sind graue Quadrate mit Haltestellenbezeichnung von A bis H ersichtlich. Somit können die einzelnen Linien des öffentlichen Verkehrs über die benutzten Haltestellen beschrieben werden (inkl. Richtungsinformation):

- Linie 1: $A \rightarrow B \rightarrow C \rightarrow D$.
- Linie 2: $E \rightarrow B \rightarrow F \rightarrow G \rightarrow H$.

Da bei Knoten-Kanten Strukturen das Verbindungselement – die Kante – eine geometrische Repräsentationsform besitzt, kann diese ebenfalls dargestellt werden. In Abbildung 6.2 sind zusätzlich zu allen Basiselementen der gemischt-einfachen Struktur die Verbindungselemente des Knoten-Kanten Modells als graue Linien dargestellt. Erst damit sind die Beziehungen zwischen den einzelnen Knoten erkennbar. Anm.: Die Schnittpunkte zwischen den Linien der Knoten-Kanten und der Kanten-Traversen Struktur besitzen keine topologische Bedeutung.

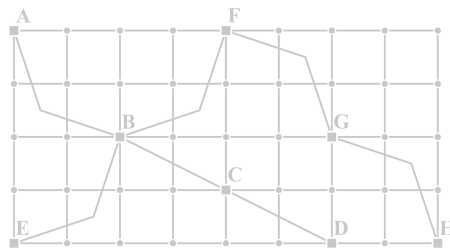


Abbildung 6.2: Basis- und Verbindungselemente einer gemischt-einfachen Struktur bestehend aus Knoten und Kanten des Wegenetzes für Fußgänger und Kanten des Liniennetzes des öffentlichen Verkehrs.

Die zeitlichen Beziehungen und Abfolgen des Liniennetzes des öffentlichen Verkehrs konnten in den vorangegangenen Abbildungen 6.1 und 6.2 auf Grund der grundrissartigen Situationsdarstellung nicht gezeigt werden. Diese Information über die raumzeitliche Ausbreitung des Liniennetzes wird in Abbildung 6.3 durch ein 2-dimensionales Raum/Zeit Diagramm aufbereitet. Hier sind entlang der Abszisse die einzelnen Haltestellen und entlang der Ordinate ein Ausschnitt des Betrachtungszeitraumes aufgetragen.

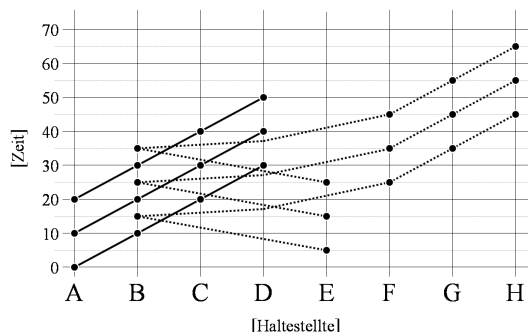


Abbildung 6.3: 2-dimensionales Raum/Zeit Diagramm des Liniennetzes des öffentlichen Verkehrs.

Die Bewegung des Verkehrsmittels ist für Linie 1 als durchgehende schwarze Liniensignatur und für die Linie 2 als schwarz-strichlierte Liniensignatur dargestellt. Den Anfang und das Ende eines jeden Liniensegmentes – der Ort, an dem eine Haltestelle verlassen oder erreicht wird – ist zusätzlich durch eine Kreissignatur ausgewiesen. Für eine bessere Übersichtlichkeit der diagrammatischen Abbildung wurde der dargestellte Linienverkehr auf drei Zyklen pro Linie beschränkt.

6.1.2 Anpassung des Algorithmus an die gegebene Problemstellung

Der in Abschnitt 5.1 beschriebene Algorithmus zur monodirektionalen Suchbaumausbreitung wird hier so erweitert, dass die zeitlichen Einschränkungen des Liniennetzes des öffentlichen Verkehrs Berücksichtigung finden. Dabei stellt, wie in Abschnitt 4.3.3 beschrieben, der Übergang zwischen dem zeitunabhängigen und dem zeitabhängigen Datensatz den einzigen zeitkritischen Vorgang in der gesamten Suchbaumausbreitung dar. Um diesen Übergang bewältigen zu können, bedarf es einer Erweiterung des Routing-Algorithmus: Ab dem Start der Ausbreitung des Suchbaumes wird zusätzlich zum Labelwert ${}^l\beta$, der die Erreichungskosten dokumentiert, ein zeitlicher Labelwert ${}^T\beta$ mitgeführt, der jenen Zeitpunkt beschreibt, an dem der Suchbaum dieses Basiselement verlässt. An Haltestellen, insbesondere beim Betreten der Haltestelle, muss in Abhängigkeit des zeitlichen Labelwertes jener Warte-Prozess als gültiger Nachbar identifiziert werden, über den der Suchbaum sich im zeitabhängigen Liniennetz des öffentlichen Verkehrs beginnt weiter auszubreiten.

Um diesen Identifikationsprozess ist der bestehende Algorithmus zu erweitern und kann als Zeitfang (*time snapping*) bezeichnet werden. Vergleichbare, auf räumliche Objekte anwendbare Funktionen können im Umfeld von Computer Aided Design (CAD) (vgl. [Aut10]) und geographischen Informationssystemen GIS (vgl. [Env10b]) gefunden werden. Die hier durchgeführte Umsetzung des Zeitfanges weist als Kernelement die Gegenüberstellung jenes Zeitpunktes ${}^T\beta_w$, an dem ein Warte-Prozess β_w angeschlossen werden würde, mit dem vom Fahrplan vorgegebenem Beginn und Ende eines Warte-Prozesses (${}^{start}\beta_w$ und ${}^{end}\beta_w$) aus der Menge der Basiselemente B . Diese Gegenüberstellung findet sich in Zeile 3, Algorithmus 8 wieder und dient als Kriterium, einen gültigen Warte-Prozess β_w als Basiselement für die weitere Suchbaumausbreitung auszuwählen. Zusätzlich wird nach erfolgreicher Iden-

tifikation eines gültigen Warte-Prozesses β_w der zeitliche Label T_{β_w} auf den identifizierten Warte-Prozess abgestimmt: $T_{\beta_w} := \text{ende}\beta_w$. Durch diese Wertzuweisung wird die aktuelle Zeit, resultierend aus der individuell gewählten Startzeit und den Benutzungszeiten der Wegeelemente, mit dem Zeitsystem des öffentlichen Verkehrs synchronisiert.

Algorithmus 8 Auswahl von gültigen Nachbarelementen beim Übergang von zeitunabhängigen zu zeitabhängigen Basisdaten: Zeitfang.

```

1: SELECT  $\beta_w$ 
2: FROM  $B$ 
3: WHERE  $T_{\beta_w}$  BETWEEN  $\text{start}\beta_w$  AND  $\text{end}\beta_w$ 

```

6.1.3 Ergebnisse

In den nachfolgenden zwei Szenarien werden Ergebnisse aus einer kürzesten Wegesuche präsentiert, die auf der zuvor beschriebenen Datenstruktur und dem erweiterten Algorithmus basiert. Im ersten Szenario werden unterschiedliche Routen durch Variation der Startzeit T_S bei gleichbleibendem Start- und Zielknoten erzeugt. Das zweite Szenario zeigt die Auswirkungen bei sich ändernden Umsteigekosten – eine Kombination aus Einsteige- und Aussteigekosten – auf. Hierbei bleiben Start- und Endknoten ebenfalls unverändert. Mit Hilfe dieser Szenarien soll die Mächtigkeit des Modellierungsverfahrens, die Nutzung einer gemischt-einfachen Struktur vor dem Hintergrund einer zeitkritischen multimodalen Anwendung, verdeutlicht werden.

Szenario 1

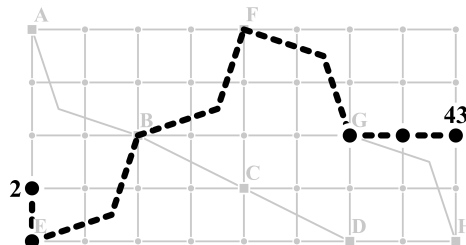


Abbildung 6.4: Multimodale Route zwischen Knoten 2 und Knoten 43 mit dem Startzeitpunkt $T_S = 3$.

In diesem Szenario werden drei berechnete Routen gegenübergestellt, die jeweils den kürzesten Weg – im Sinne eines kostengünstigsten Weges – zwischen dem Startknoten 2 und dem Zielknoten 43 zum Inhalt haben. In Abbildung 6.4 wird die berechnete Route zum Startzeitpunkt $T_S = 3$ als schwarz-strichlierte Linie dargestellt, die sich in folgende Segmente teilt:

1. Bewegung vom Startknoten 2 zur Haltestelle E der Linie 2 entlang des Wegenetzes für Fußgänger.

2. Warten auf Linie 2 an der Haltestelle E im Liniennetz des öffentlichen Verkehrs.
3. Fahrt mit Linie 2 von der Haltestellen E zur Haltestelle G entlang des Liniennetzes des öffentlichen Verkehrs.
4. Bewegung von der Haltestelle G zum Zielknoten 43 entlang des Wegenetzes für Fußgänger.

Die in Abbildung 6.5 gezeigte Route stellt die optimale Lösung zur Startzeit $T_S = 10$ dar. Hierbei wird im Verlauf der berechneten Route ein Umstieg der Linie des öffentlichen Verkehrs vollzogen. Im Detail betrachtet, setzt sich die Route aus den nachstehenden Segmenten zusammen:

1. Bewegung vom Startknoten 2 zur Haltestelle A der Linie 1 entlang des Wegenetzes für Fußgänger.
2. Warten auf Linie 1 an der Haltestelle A im Liniennetz des öffentlichen Verkehrs.
3. Fahrt mit Linie 1 von der Haltestelle A zur Umstiegshaltestelle B entlang des Liniennetzes des öffentlichen Verkehrs.
4. Warten auf Linie 2 an der Haltestelle B im Liniennetz des öffentlichen Verkehrs.
5. Fahrt mit Linie 2 von der Haltestelle B zur Haltestelle G entlang des Liniennetzes des öffentlichen Verkehrs.
6. Bewegung von der Haltestelle G zum Zielknoten 43 entlang des Wegenetzes für Fußgänger.

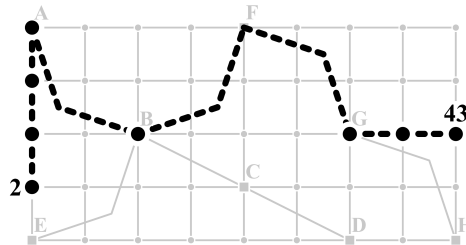


Abbildung 6.5: Multimodale Route zwischen Knoten 2 und Knoten 43 mit dem Startzeitpunkt $T_S = 10$.

Die letzte hier beschriebene Variante – Abbildung 6.6 – zeigt die resultierende Route bei einer Startzeit $T_S = 13$. Diese Route besteht aus einem umstiegsfreien Weg mit folgenden Segmenten:

1. Bewegung vom Startknoten 2 zur Haltestelle B der Linien 1 und 2 entlang des Wegenetzes für Fußgänger.
2. Warten auf Linie 2 an der Haltestelle B im Liniennetz des öffentlichen Verkehrs.
3. Fahrt mit Linie 2 von der Haltestellen B zur Haltestelle G entlang des Liniennetzes des öffentlichen Verkehrs.

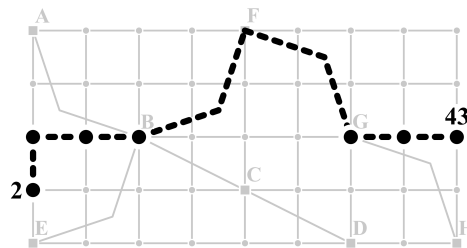


Abbildung 6.6: Multimodale Route zwischen Knoten 2 und Knoten 43 mit dem Startzeitpunkt $T_S = 13$.

4. Bewegung von der Haltestelle G zum Zielknoten 43 entlang des Wegenetzes für Fußgänger.

Der Vergleich dieser drei Ergebnisse stellt die Flexibilität des Algorithmus angewandt auf die Datenbasis dar: Erreicht der Suchbaum eine Haltestelle zu einem ungünstigen Zeitpunkt – ein teuer bewerteter Warte-Prozess muss in Kauf genommen werden, um an dieser Haltestelle in das öffentliche Verkehrsmittel einsteigen zu können – so wird eine Haltestelle mit einer geringeren Wartezeit aufgesucht. Dabei ist es belanglos, ob es sich um eine Haltestelle derselben Linie handelt, deren bisher untersuchte Haltestelle als zu teuer bewertet wurde.

Besonders deutlich wird der Einfluss teurer Warte-Prozesse bei der Gegenüberstellung der Routen aus den Abbildungen 6.4 und 6.5. Bezogen auf die Route aus Abbildung 6.5 ist die Summe der Kosten aus der Bewegung vom Startknoten 2 zur Haltestelle E , das Warten an Haltestelle E und die Fahrt zwischen Haltestelle E und Haltestelle B höher als der weitere Fußweg vom Startknoten 2 zur Haltestelle A , das Warten an der Haltestelle A , die Fahrt zwischen Haltestelle A und Haltestelle B und das Umsteigen an der Haltestelle B .

Vergleicht man die Routen aus Abbildung 6.4 und Abbildung 6.6, die in beiden Fällen ausschließlich die Linie 1 des öffentlichen Verkehrs nutzen, ist die unterschiedliche Einstiegs-haltestelle in die Linie 1 erkennbar. Trotz der räumlichen Nähe von Haltestelle E zum Startknoten 2, wird diese Haltestelle zeitlich betrachtet so ungünstig vom Suchbaum erreicht, dass der längere Fußweg zur Haltestelle B in Kauf genommen wird, um in die Linie 1 einsteigen zu können.

Szenario 2

Das zweite Szenario dokumentiert die Auswirkung von Einsteige- und Aussteigekosten auf die Berechnung einer Route. Mit Hilfe dieser Kosten – bei Kombination beider Kosten sind Umsteigekosten modellierbar – können Einschränkungen, wie etwa körperliche Gebrechen oder das Mitführen von Kinderwagen, bei der Nutzung des öffentlichen Verkehrs im Zuge der Routenberechnung berücksichtigt werden. Hierfür werden zwei Routen zwischen Knoten 4 und Knoten 43 berechnet, wobei im ersten Fall günstige Umsteigekosten und im zweiten Fall teure Umsteigekosten den Routenverlauf beeinflussen.

In Abbildung 6.7 wird jene Route dargestellt, der günstige Einsteige- und Aussteigekosten zugrunde liegen. Diese Route besteht aus folgenden Segmenten:

1. Bewegung vom Startknoten 4 zur Haltestelle A der Linie 1 entlang des Wegenetzes für Fußgänger.

2. Warten auf Linie 1 an der Haltestelle A im Liniennetz des öffentlichen Verkehrs.
3. Fahrt mit Linie 1 von der Haltestelle A zur Umstiegshaltestelle B entlang des Liniennetzes des öffentlichen Verkehrs.
4. Warten auf Linie 2 an der Haltestelle B im Liniennetz des öffentlichen Verkehrs.
5. Fahrt mit Linie 2 von der Haltestelle B zur Haltestelle G entlang des Liniennetzes des öffentlichen Verkehrs.
6. Bewegung von der Haltestelle G zum Zielknoten 43 entlang des Wegenetzes für Fußgänger.

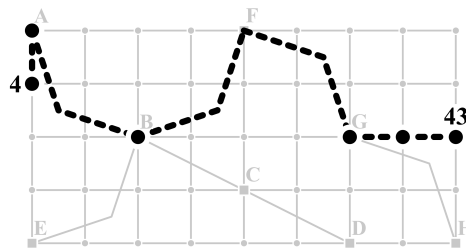


Abbildung 6.7: Multimodale Route zwischen Knoten 4 und Knoten 43 mit günstigen Einsteige- und Aussteigekosten in das Verkehrsmittel.

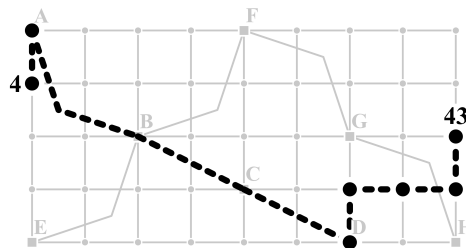


Abbildung 6.8: Multimodale Route zwischen Knoten 4 und Knoten 43 mit teuren Einsteige- und Aussteigekosten in das Verkehrsmittel.

Bei Erhöhung der Einsteige- und Aussteigekosten kann eine durchgehende Nutzung der Linie 1 erreicht werden (siehe Abbildung 6.8). Die erhaltene Route gliedert sich dabei folgendermaßen:

1. Bewegung vom Startknoten 4 zur Haltestelle A der Linie 1 entlang des Wegenetzes für Fußgänger.
2. Warten auf Linie 1 an der Haltestelle A im Liniennetz des öffentlichen Verkehrs.
3. Fahrt mit Linie 1 von der Haltestelle A zur Haltestelle D entlang des Liniennetzes des öffentlichen Verkehrs.

4. Bewegung von der Haltestelle D zum Zielknoten 43 entlang des Wegenetzes für Fußgänger.

Die Gegenüberstellung beider Routen aus den Abbildungen 6.7 und 6.8 verdeutlicht die Auswirkung der unterschiedlich hohen Einsteige- und Aussteigekosten auf das Berechnungsergebnis. Wird bei niedrigen Umsteigekosten der Wechsel von Linie 1 auf Linie 2 an Haltestelle B vollzogen (vgl. Abbildung 6.7), so wird bei hohen Umsteigekosten eine durchgängige Nutzung der Linie 1 bevorzugt (vgl. Abbildung 6.8). Hierbei wird die längere Strecke im Wegenetz für Fußgänger – von Haltestelle D bis zum Zielknoten 43 – vom Algorithmus als vorteilhafter beurteilt, als eine Lösung, die einem Umsteige-Prozess an Haltestelle B mit der anschließenden Nutzung der Linie 2 zugrunde liegt.

6.2 Short-Cut Routing

Dieser Abschnitt umfasst die Bearbeitung eines SPP innerhalb einer gemischt-komplexen Struktur. Diese Struktur setzt sich aus einem Kanten-Traversen Modell zur Abbildung eines Wegenetzes für Fußgänger und einem polygonalen Modell, womit freies Gelände beschrieben wird, zusammen. Anm.: Eine Realisierung des Wegenetzes für Fußgänger als Knoten-Kanten Modell in Anlehnung an Abschnitt 6.1 ist genauso durchführbar. Eine vergleichbare Zusammenstellung von Basisdaten zur Lösung logistischer Aufgabenstellungen findet sich unter anderem bei [Thi09]. Im Gegensatz zum multimodalen Anwendungskonzept in Abschnitt 6.1 stehen hier der Suchbaumausbreitung explizite Nachbarschaftsbeschreibungen aus dem Kanten-Traversen Modell impliziten Nachbarschaftsbeschreibungen aus dem polygonalen Modell gegenüber, die jedoch gleichberechtigt im Algorithmus zu berücksichtigen sind. Diese Suchbaumausbreitung erfolgt monodirektional gemäß Abschnitt 5.1 und benötigt den *Startort* und den *Zielort* als Eingangsparameter.

6.2.1 Datengrundlage

Die Datengrundlage des *Short-Cut Routing* Anwendungskonzeptes besteht aus zwei eigenständigen Strukturen:

- Das Wegenetz für Fußgänger als Kanten-Traversen Modell gemäß Abschnitt 3.2.
- Freies Gelände als polygonales Modell gemäß Abschnitt 3.4, das von einem Fußgänger überquert werden kann.

Räumlich betrachtet (vgl. Abbildung 6.9) überlagern sich beide Modelle, wodurch sie für ein *Short-Cut Routing* gemeinsam als gemischt-komplexe Struktur nach Abschnitt 3.6 genutzt werden können. Das Basiselement des Kanten-Traversen Modells wird in dieser Darstellung als schwarze Linie abgebildet, dessen Länge der doppelten Seitenkantenlänge eines Flächenelementes des polygonalen Modells entspricht. Das Basiselement des polygonalen Modells ist als graue quadratische Fläche mit weißen Seitenkanten ersichtlich. Dabei sind die einzelnen Flächen so ausgerichtet, dass sie einander lückenlos berühren. Erst durch diese lückenlose Berührung können Nachbarflächen erkannt und nach Abschnitt 6.2.2 für die Ausbreitung des Suchbaumes genutzt werden. Bezogen auf die Kostengestaltung der Flächenelemente wird hier prinzipiell ein individuelles Bewertungsverfahren eingesetzt (vgl. Tabelle 3.10 aus Abschnitt 3.4), jedoch kann speziell im Hinblick auf die ersten beiden Szenarien in Abschnitt 6.2.3 die Bewertung auch als äquimontär verstanden werden.

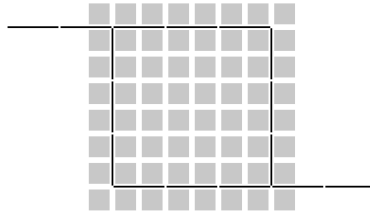


Abbildung 6.9: Basiselemente einer gemischt-komplexen Struktur bestehend aus Kanten des Wegenetzes und Polygone des freien Geländes.

Einen wesentlichen Bestandteil gemischt-komplexer Strukturen stellt nach Abschnitt 3.6 das RTE dar. Erst dadurch kann sich der Suchbaum von einem Datenmodell – z.B.: dem Kanten-Traversen Modell – in einem andern Datenmodell – z.B.: dem polygonalen Modell – weiter ausbreiten. Für dieses Anwendungskonzept werden zwei RTE-Paare definiert: Hierbei werden jeweils eine Kante des Kanten-Traversen Modells und ein Flächenelement des polygonalen Modells explizit als richtungsunabhängige Nachbarn ausgewiesen. Der Wechsel von der Kante zur Fläche und von der Fläche zur Kante werden dadurch ermöglicht. Diese speziellen Basiselemente sind in Abbildung 6.10 als schwarze Linie oder als schwarze Fläche dargestellt, alle übrigen Elemente sind grau eingefärbt.

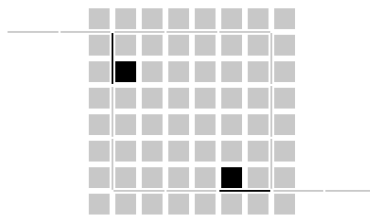


Abbildung 6.10: Definierte Übergänge zwischen Wegenetz und freiem Gelände (RTE). RTE-Paare im linken oberen und rechten unteren Datenbereich.

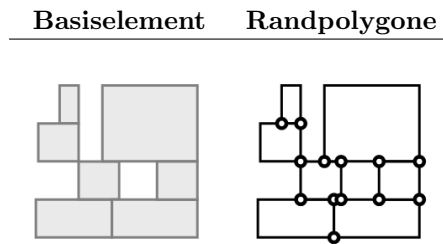
6.2.2 Anpassung des Algorithmus an die gegebene Problemstellung

Das Schwergewicht der Algorithmusanpassung liegt in der Erkennung und Bewertung von Nachbarschaften innerhalb des polygonalen Modells: Für eine Suchbaumausbreitung innerhalb flächenhafter Strukturen müssen jene Informationen, die in Knoten-Kanten oder Kanten-Traversen Modellen im Verbindungselement vorgehalten werden können, während des Ausbreitevorganges durch Interpretation der räumlichen Situation dynamisch gewonnen werden.

Zur Beschreibung von Nachbarschaftsverhältnissen erfolgt ein Übergang von Flächenelementen hin zu ihren Randpolygone (vgl. [Bar05, Abschnitt 3.1.2]). In Tabelle 6.1 werden einem Basisdatensatz eines polygonalen Modells die dazugehörigen Randpolygone gegenübergestellt. [Bar05] beschreibt das Thema der Randpolygone vor dem Hintergrund von

Ländergrenzen und Dreiländerecken. Diese topologischen Merkmale sind in Tabelle 6.1 graphisch als Linien oder als Kreissymbole zu erkennen.

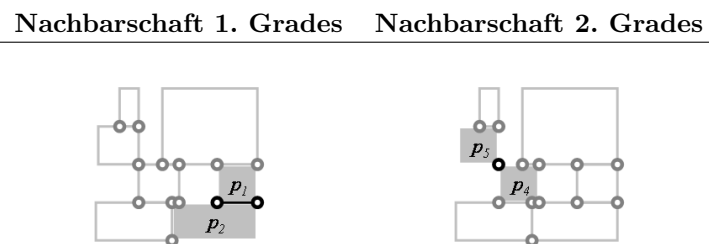
Tabelle 6.1: Datengrundlage für Nachbarschaftsanalyse.



Basierend auf der Thematik der Randpolygone können topologisch betrachtet zwei Grade von Nachbarschaften identifiziert werden, die assoziativ zu Ländergrenzen beziehungsweise zu Dreiländerecken zuzuordnen sind:

- Nachbarschaft 1. Grades: Die Randpolygone zweier Flächen, p_1 und p_2 aus Tabelle 6.2, berühren einander entlang eines gemeinsamen Liniensegmentes.
- Nachbarschaft 2. Grades: Die Randpolygone zweier Flächen, p_4 und p_5 aus Tabelle 6.2, berühren einander in einem einzelnen Punkt.

Tabelle 6.2: Grade der Nachbarschaft.



Diese Einteilung von Nachbarschaften ist mit der Einteilung in 4er- bzw. *Von-Neumann*-Nachbarschaften und 8er- bzw. *Moore*-Nachbarschaften aus der digitalen Bildverarbeitung vergleichbar, mit der die nachbarschaftlichen Beziehungen zwischen einzelnen Bildpunkten in Rastergraphiken beschrieben werden (vgl. [Lan09, 12.4.5.]). Die programmatische Umsetzung der Nachbarschaftsanalyse wird in diesem Anwendungskonzept, bedingt durch die vektorgeometrische Beschreibung der Flächenelemente durch Polygone, über eine in [Ope06b] beschriebene Analysefunktion gelöst, `st_touches(<geometry>, <geometry>)`, die bei Berührung der beiden Flächengeometrien den Wert `true`, ansonsten den Wert `false` liefert (vgl. [Ope06b]). Anm.: Diese Vorgehensweise dient ausschließlich der Erkennung von

Nachbarn, den Grad einer Nachbarschaft liefert diese Funktion jedoch nicht. Diese Vorgehensweise wird auch in Abschnitt 3.4 über die Umbildung eines polygonalen Modells in seinen Regionsadjazenzgraphen beschrieben (vgl. Algorithmus 4).

Ein anderer wesentlicher Bestandteil eines Verbindungselementes $\alpha_i = (\beta_k, \beta_l)$ stellt abgesehen von der Nachbarschaftsbeschreibung durch ${}^p\alpha_i$ und ${}^s\alpha_i$ die Kosteninformation für eine Bewegung vom Basiselement $\beta_k \equiv {}^p\alpha_i$ zum Basiselement $\beta_l \equiv {}^s\alpha_i$ dar: ${}^c\alpha_i$. Diese Bewegungskosten finden sich in einer polygonalen Modellierung nicht wieder. Räumliche Analysen, angewandt auf das zentrale Flächenelement p_c unter Einbeziehung der Nachbar-elemente p_i , bieten eine mögliche Variante zur Kostenbewertung von Bewegungen zwischen Basiselementen. In Abbildung 6.11 wird das hier verwendete Prinzip dargestellt, das nicht nur auf eine rasterartige Datenausprägung angewandt, sondern auch bei freier Formgebung des Basiselementes genutzt werden kann: Je länger die gemeinsame Grenze zwischen Zentralelement und Nachbar ist, umso günstiger ist eine Bewegung zwischen diesen.

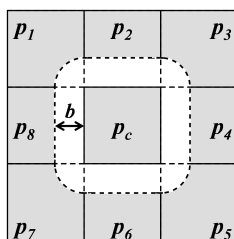


Abbildung 6.11: Modellierung von Übergangskosten ${}^c t$ zwischen dem Zentralpolygon p_c und den Nachbarpolygonen $\{p_1, \dots, p_8\}$ in Abhängigkeit der Nachbarschaft.

Unter Berücksichtigung der zur Verfügung stehenden Funktionen innerhalb des Datenbankmanagementsystems, in dem der Algorithmus umgesetzt wird, kann auf die Länge der gemeinsamen Grenze nicht direkt zugegriffen werden. Hier wird zur Bemessung der Nachbarschaft das zentrale Flächenelement p_c um einen Puffer mit Radius b erweitert – `st_buffer(<geometry>, <size>)` (vgl. [Ope06b]). Die dadurch entstehenden Überlappungen zwischen dem vergrößerten zentralen Flächenelement p_c und seinen Nachbarflächen p_i werden durch die Funktion `st_area(<geometry>)` (vgl. [Ope06b]) flächenmäßig erfasst, invertiert und als Kostenwert ${}^c t_{p_c p_i}$ nach Formel 6.1 für die Ausbreitung des Suchbaumes verwendet.

$${}^c t_{p_c p_i} = \frac{1}{\text{st.area}(\text{st.buffer}(p_c, b) \cap p_i)}. \quad (6.1)$$

6.2.3 Ergebnisse

Drei SPP werden in den nachfolgenden Szenarien gelöst, die jeweils die Kante 1 als Startelement und die Kante 16 als Zielelement aufweisen. Der Unterschied zwischen den einzelnen Szenarien findet sich in den Gestaltung der Benutzungskosten innerhalb der gemischt-komplexen Struktur: Zum einen variieren die Kosten zwischen dem Kanten-Traversen Modell und dem polygonalen Modell, und zum anderen erfolgt eine individuelle Bewertung innerhalb des polygonalen Modells. Die entstehenden Routen sollen die Kombinierbarkeit sich

stark unterscheidender Basisdaten – mit bzw. ohne Verbindungselement – aufzeigen, mit denen verschiedenste Transportträger bedarfsorientiert modelliert und gemeinsam in einer einzigen Suchbaumausbreitung genutzt werden können.

Szenario 1

Dieses Szenario behandelt die Aufgabenstellung, eine Route zwischen der Startkante 1 und der Zielkante 16 zu bestimmen. Die Kostengestaltung der Basis- und Verbindungselemente ist so gewählt, dass die Nutzung von Basiselementen aus dem polygonalen Modell als ungünstiger im Vergleich zu den Elementen aus dem Kanten-Traversen Modell bewertet wird. Innerhalb des Kanten-Traversen Modells sind alle Elemente mit denselben Kosten ausgewiesen.

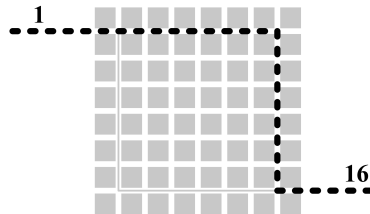


Abbildung 6.12: Short-Cut Route zwischen Kante 1 und Kante 16 bei günstigeren Kanten- und Traversennutzungskosten als Polygonnutzungskosten und Kostengleichheit innerhalb der Kanten-Traversen Struktur.

Das Ergebnis der Routenberechnung, als schwarz-strichlierte Linie in Abbildung 6.12 dargestellt, zeigt eine durchgehende Nutzung des Kanten-Traversen Modells – ein Wechsel auf die polygonalen Flächenelemente erfolgt nicht. Das einzige Routensegment setzt sich folgendermaßen zusammen:

1. Bewegung von der Startkante 1 zur Zielkante 16 entlang des Wegenetzes für Fußgänger.

Szenario 2

Beginnt man die Benutzungskosten der Flächenelemente aus Szenario 1 zu senken, erhöht man gleichsam die Attraktivität, Basiselemente des polygonalen Modells in der Route zu verwenden. Die Aufgabenstellung in Szenario 2 ist ident zu jener aus Szenario 1: Finde den kürzesten Weg zwischen Kante 1 und Kante 16. Durch die günstigere Bewertung der Flächenelemente beinhaltet die berechnete Route eine „Abkürzung“ über das freie Gelände und setzt sich aus folgenden Segmenten zusammen:

1. Bewegung von der Startkante 1 zum RTE entlang des Wegenetzes für Fußgänger.
2. Wechsel vom Kanten-Traversen Modell zum polygonalen Modell am RTE.
3. Bewegung innerhalb des freien Geländes zwischen den beiden RTE.
4. Wechsel vom polygonalen Modell zum Kanten-Traversen Modell am RTE.

5. Bewegung vom RTE zur Zielkante 16 entlang des Wegenetzes für Fußgänger.

Die berechnete Route wird in Abbildung 6.13 dargestellt, wobei die Routensegmente des Wegenetzes für Fußgänger schwarz-strichliert und die involvierten Flächenelemente schwarz gefüllt visualisiert werden. Die geometrische Form der Abkürzung – das Routensegment innerhalb des polygonalen Modells – ist dabei von der Bewertung der einzelnen Flächenelemente abhängig, die speziell in diesem Szenario kostengleich umgesetzt wurde. Deshalb verbindet die Abkürzung beiden RTE auf direktem Weg.

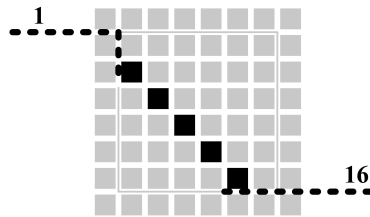


Abbildung 6.13: Short-Cut Route zwischen Kante 1 und Kante 16 bei günstigeren Polygonnutzungskosten als Kanten- und Traversennutzungskosten und Kostengleichheit innerhalb der polygonalen Datenstruktur.

Szenario 3

Im dritten Szenario wird genauso nach einer kürzesten Verbindung zwischen Kante 1 und Kante 16 gesucht. Wie in Szenario 2 sind die Flächenelemente des polygonalen Modells so bewertet, dass eine Nutzung dieser Elemente gewährleistet ist. Der Unterschied zu Szenario 2 liegt in der Bewertung der einzelnen Flächenelemente: Waren diese Kosten zuvor für alle Elemente gleich, so variieren diese im dritten Szenario untereinander.

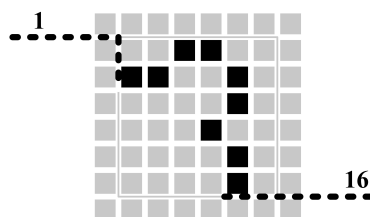


Abbildung 6.14: Short-Cut Route zwischen Kante 1 und Kante 16 bei günstigeren Polygonnutzungskosten als Kantennutzungskosten und ungleiche Kosten innerhalb der polygonalen Datenstruktur.

Diese Variation der Kosten wirkt sich unmittelbar auf das Ergebnis der Routenberechnung aus. Das Routensegment zwischen den beiden RTE innerhalb des polygonalen Modells verläuft nun nicht mehr auf direktem Weg, sondern spiegelt die Reaktion der Suchbaumausbreitung auf die individuelle Bewertung wider. In Abbildung 6.14 sind die Bestandteile

der Route in derselben thematischen Abfolge ersichtlich, wie sie bereits in Szenario 2 auftritt (Wegenetz für Fußgänger, freies Gelände und erneut Wegenetz für Fußgänger), nur die geometrische Form des Routensegmentes innerhalb des polygonalen Modells unterscheidet sich.

In weiterer Folge finden sich für unterschiedliche Anwendungsfälle des *Short-Cut Routings* passende Bewertungsgrundlagen für die flächenhaften Basiselemente des polygonalen Modells. Hierzu zählen digitale Geländehöhenmodelle und deren ableitbare Größen, wie beispielsweise die Steigung, hinsichtlich Landbedeckung und Landnutzung klassifizierte Fernerkundungsdaten oder spezielle Situationsaufnahmen von Plätzen und Hallen.

6.3 Cube Routing

Im Sinne einer konsequenten Fortsetzung der Anwendungsszenarien vor dem Hintergrund der eingesetzten geometrischen Repräsentationsformen der einzelnen Basiselemente, muss man sich zwangsläufig mit 3-dimensionalen Objekten auseinandersetzen. Wurden 0- und 1-dimensionale Daten in Abschnitt 6.1 und 1- und 2-dimensionale Daten in Abschnitt 6.2 in kombinierter Weise zur Routenberechnung eingesetzt, so bilden hier 3-dimensionale Basiselemente die Datengrundlage zur Lösung eines SPP. Auf eine gemeinsame Nutzung unterschiedlicher Basisdaten wird in diesem Abschnitt verzichtet, da auf die Besonderheiten der Suchbaumausbildung innerhalb gemischt-einfacher und gemischt-komplexer Strukturen in den Abschnitten 6.1 und 6.2 bereits eingegangen wurde. Die Ausbreitung des Suchbaumes ist wie in den vorangegangenen Abschnitten monodirektional nach Abschnitt 5.1 umgesetzt und wird durch Angabe eines *Startortes* und eines *Zielortes* parametrisiert.

6.3.1 Datengrundlage

Die Datenbasis zur Lösung eines SPP vor dem Hintergrund eines *Cube Routing* Anwendungskonzeptes besteht aus einer einzigen Struktur:

- Volumnhafte Situationsdarstellung anhand eines polyhedralen Modells gemäß Abschnitt 3.5.

Die einzelnen Basiselemente – Polyeder mit sechs quadratischen Seitenflächen – sind dabei so angeordnet, dass die Gesamtheit der Elemente einen großen Würfel bildet. Entlang einer Seitenkante dieses großen Würfels finden sich zehn Basiselemente – somit besteht die gesamte Datengrundlage aus 1000 Basiselementen. Betrachtet man die Anordnung der Basiselemente im Detail, so sind diese auf Grund der darauf anzuwendenden Nachbarschaftsanalyse berührend platziert. Ein Verbindungselement ist somit innerhalb der Datenbasis nicht vorzuhalten, wie es auch für das polygonale Modell innerhalb der gemischt-komplexen Struktur aus dem *Short-Cut Routing* Anwendungskonzept in Abschnitt 6.2 geschieht.

In Abbildung 6.15 wird diese Datengrundlage dargestellt, wobei eine farbliche Unterscheidung hinsichtlich der Benutzungskosten vorgenommen wurde: Hellgraue teiltransparente Würfel besitzen niedrige Benutzungskosten; dunkelgraue teiltransparente Würfel besitzen hohe Benutzungskosten. Anm.: Zu einem besseren Verständnis der Abbildung wird zusätzlich ein Achsenkreuz dargestellt, das jedoch nicht als Bestandteil der Basisdaten zu verstehen ist.

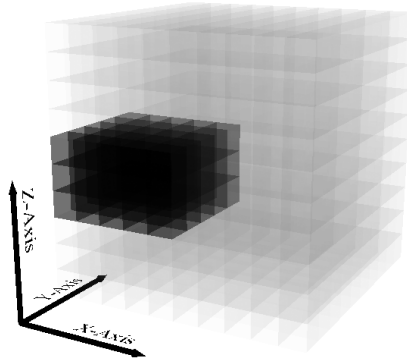


Abbildung 6.15: Basiselemente einer polyhedralen Struktur.

6.3.2 Anpassung des Algorithmus an die gegebene Problemstellung

Die Ausbreitung eines Suchbaumes innerhalb 3-dimensionaler Daten stellt im Vergleich zu einer 0-, 1- oder 2-dimensionalen Datenbasis besonders hohe Anforderungen an die eingesetzten Analysefunktionen. Zu dieser Thematik passend schreibt [Sv02] über die Anwendung bestehender Funktionen, die auf im \mathcal{R}^3 platzierte Elemente zwar angewendet werden können, aber oftmals die dritte Koordinatenkomponente nicht berücksichtigen: „[...] An experiment showed that the z-value, defined in the available geometry types is not used in spatial queries. [...]“.

Nachvollzogen kann diese Problematik durch einfache Flächenberechnung eines Polygons werden: Hierzu wird eine Geometrie in einer *PostgreSQL/PostGIS* Datenbank definiert, die einem Quadrat mit einer Seitenkantenlänge von 1[LE] entspricht und somit eine Fläche von 1[LE²] aufweist. Platziert man nun diese Fläche im \mathcal{R}^3 , sodass sie sich in der YZ-Ebene befindet, liefert die darauf angewandte `st_area(<geometry>)` Funktion das Ergebnis 0[LE²]. Durch die Platzierung in der YZ-Ebene erscheint das Polygon in der XY-Ebene projizierend – als Strecke – und besitzt somit kein Flächenmaß. Dieser beschriebene Fall wird in Algorithmus 9 in Form einer spatialSQL Anweisung angeführt.

Algorithmus 9 Flächenberechnung eines in der YZ-Ebene gelagerten Polygons.

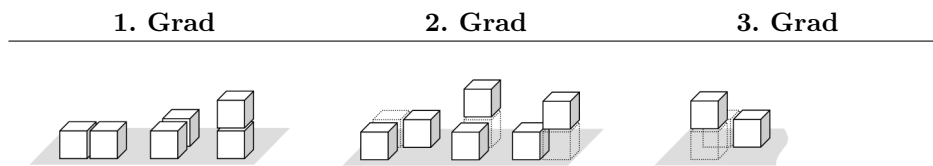
```
1: SELECT st_area(st_geometryfromtext('POLYGON((0 0 0,0 1 0,0 1 1,0 0 1,0 0 0))')
```

Diese Eigenheit, 3-dimensionale Objekte bisher nur eingeschränkt mit Standardfunktionen analysieren zu können, wirkt sich auf die Nachbarschaftsbeschreibung innerhalb des polyhedralen Modells aus: Speziell auf die Datenbasis angepasste Verfahren sind zu entwickeln. Die Grundlage dafür liefern Definitionen topologischer Beziehungen, wie sie beispielsweise bei [Zla00] und darauf aufbauend bei [CARZ08] beschrieben werden. Aus dieser Menge an Definitionen bieten sich für die Ausbreitung von Suchbäumen die *Meet* und die *Disjoint* Beziehung an, womit das Bestehen eines nachbarschaftlichen Verhältnisses zwischen zwei Volumenobjekten beurteilt werden kann. Vor dem Hintergrund der gewählten Struktur der

Datenbasis – eine lückenlose Aneinanderreihung von Würfeln – erfolgt in diesem Anwendungskonzept eine Klassifizierung der Nachbarschaften angelehnt an [GW09, 22.2], wobei diese graphisch in Tabelle 6.3 einander gegenübergestellt sind:

- Nachbarschaft 1. Grades: Die in nachbarschaftlicher Beziehung stehenden Objekte berühren einander an ihren Seitenflächen. Eine Ausbreitung des Suchbaumes erfolgt hier entlang einer einzigen Koordinatenachse: $X \vee Y \vee Z$. Anm.: Diese Nachbarschaftsbeschreibung entspricht jener 6er-Nachbarschaft bei [GW09].
- Nachbarschaft 2. Grades: Die in nachbarschaftlicher Beziehung stehenden Objekte berühren einander entlang ihrer Seitenkanten. Eine Ausbreitung des Suchbaumes erfolgt hier entlang zweier Koordinatenachsen: $(X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z)$. Anm.: Die bei [GW09] beschriebene 18er-Nachbarschaft umfasst Nachbarschaften 1. und 2. Grades.
- Nachbarschaft 3. Grades: Die in nachbarschaftlicher Beziehung stehenden Objekte berühren einander an ihren Ecken. Eine Ausbreitung des Suchbaumes erfolgt hier entlang aller drei Koordinatenachsen: $X \wedge Y \wedge Z$. Anm.: Die bei [GW09] beschriebene 26er-Nachbarschaft umfasst Nachbarschaften 1., 2. und 3. Grades.

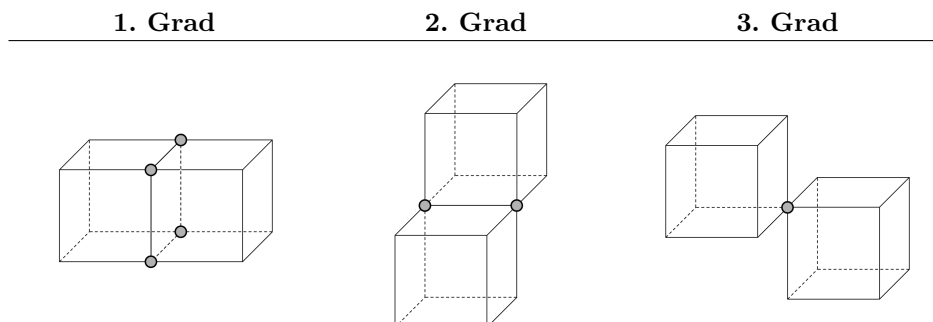
Tabelle 6.3: Gegenüberstellung der verschiedenen Nachbarschaftsklassen.



Die Identifikation dieser drei Nachbarschaftsklassen mit ihren geometrischen Eigenschaften – gemeinsame Seitenfläche, Seitenkante oder Ecke – fließt in die Entwicklung eines Algorithmus zur Nachbarschaftserkennung ein. Basierend auf der gewählten geometrischen Beschreibungsform der polyhedralen Basiselemente durch ihre Eckpunkte (vgl. Abschnitt 3.5) und deren spezielle Anordnung innerhalb der Datenbasis, sind die jeweiligen Bedingungen für einen bestimmten Grad einer Nachbarschaft zu formulieren (vgl. Tabelle 6.4): Zwei Basiselemente, die eine Nachbarschaft 1. Grades eingehen, besitzen vier koordinativ idente Eckpunkte; zwei Basiselemente, die eine Nachbarschaft 2. Grades eingehen, besitzen zwei koordinativ idente Eckpunkte; zwei Basiselemente, die eine Nachbarschaft 3. Grades eingehen, besitzen einen koordinativ identen Eckpunkt.

Die programmatische Umsetzung der Nachbarschaftsanalyse stützt sich in diesem Anwendungskonzept unter den gegebenen Rahmenbedingungen auf das Ergebnis eines Koordinatenvergleiches in jeder Achse. Dazu liefern die Funktionen `st_x(<geometry>)`, `st_y(<geometry>)` und `st_z(<geometry>)`, angewandt auf die Basiselemente, die notwendige numerische Informationsgrundlage. Im Vergleich zum *Short-Cut Routing* aus Abschnitt 6.2 und der dortig eingesetzten Methode zur Erkennung von Nachbarelementen innerhalb des polygonalen Modells mittels `st_touches(<geometry>, <geometry>)`, kann hier zusätzlich zum Erkennen von Nachbarn nicht nur der Grad der Nachbarschaft sondern auch die relative Lage zum

Tabelle 6.4: Übereinstimmende Eckpunkte für die Nachbarschaftsanalyse bei polyhedraler Modellierung.



Zentralelement identifiziert werden. Das Wissen über die relative Lage zweier benachbarter Basiselemente zueinander kann dabei für eine gerichtete Suchbaumausbreitung genutzt werden. Einem Basiselement wird dafür die Information über zulässige Nachbarschaften mitgegeben, z.B.: *Bei Basiselement c_i sind nur Nachbarschaften 1. und 2. Grades zulässig, bei denen sich der Suchbaum entgegengesetzt zur Z-Achse ausbreitet.* Diese Form einer gerichteten Datenbasis – das Basiselement β weist eine Richtungsinformation auf – unterscheidet sich von jener bei Knoten-Kanten oder Kanten-Traversen Modellen, bei denen das Verbindungselement α die Richtungsinformation beinhaltet.

6.3.3 Ergebnisse

In den drei nachfolgenden Szenarien werden SPP gelöst, deren Start- und Zielelement jeweils ident sind. Die Möglichkeit, zulässige Nachbarschaften für eine Suchbaumausbreitung zu definieren, wird in unterschiedlicher Weise in den einzelnen Szenarien genutzt: Werden im ersten Szenario nur Nachbarschaften 1. Grades zugelassen, so erfolgt eine Erweiterung auf 1. und 2. Grad im zweiten Szenario und eine Nutzung aller Nachbarschaftsgrade im dritten Szenario.

Diese Variante der Routenberechnung – die Nutzung 3-dimensionaler Basisdaten – wird hierbei beispielhaft wegen der Definition von Start- und Zielelement als SSSDP aufbereitet. Führt man eine Erreichbarkeitsanalyse im Sinne eines SSMDP durch, so eröffnen sich Anwendungsfelder, wie die Simulation von Emissionsprozessen oder Versickerungsvorgängen, die mit Werkzeugen der Navigation bedient werden können.

Szenario 1

Die Möglichkeit zur Einschränkung der zulässigen Nachbarschaftsverhältnisse zwischen einzelnen Basiselementen wird im ersten Szenario aufgezeigt. Hier wird eine kürzeste Verbindung zwischen zwei Würfelementen gesucht, wobei ausschließlich Nachbarschaften des 1. Grades für eine Suchbaumausbreitung zu nutzen sind.

Die Abbildung 6.16 zeigt das Resultat dieser Verbindungssuche, worin die berechnete Route in einem satten Grauton dargestellt wird. Die Auswirkung der Nachbarschaft-

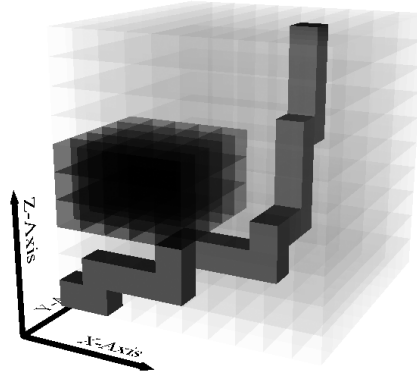


Abbildung 6.16: Route in einer 3-dimensionalen Datenbasis bei eingeschränkten Nachbarschaftsverhältnissen: Nur Nachbarschaften 1. Grades.

seinschränkung sind deutlich erkennbar: Die Würfelemente entlang der Route liegen Seitenfläche an Seitenfläche zu ihren jeweiligen Nachbarn. Abgesehen von der Bedingung, einem definierten Nachbarschaftsgrad zu entsprechen, wirkt sich die Bewertung der einzelnen Basiselemente auf den Verlauf der Route aus. Da die Datenbasis nur zwei unterschiedliche Bewertungskosten aufweist (vgl. Abschnitt 6.3.1) und sich die berechnete Route zur Gänze innerhalb der günstig bewerteten Basiselemente befindet, ist die geometrische Form der Route vor dem Hintergrund einer *Manhattan Metrik* zu sehen – in diesem Szenario, einer 3-dimensionalen *Manhattan Metrik*: Gleichgültig wie viele Kurven eine Route aufweist, solange sie sich in Richtung Ziel bewegt, ist sie als eine kürzeste Verbindung zu sehen.

Szenario 2

Eine Lockerung der Nachbarschaftseinschränkung erfolgt im zweiten Szenario. Zusätzlich zum 1. Grad einer Nachbarschaft – Basiselemente berühren einander an ihren Seitenflächen – wird der 2. Grad einer Nachbarschaft – Basiselemente berühren einander an ihren Seitenkanten – für die Ausbreitung eines Suchbaumes zugelassen. Dadurch wird das Spektrum an Ausbreitungsvarianten um Bewegungen, die sich diagonal in XY-, XZ- oder YZ-Ebene erstrecken, erweitert.

Das Ergebnis eines Routenplanungsprozesses, dem dasselbe Start- und Zielelement aus Szenario 1 zugrunde liegt, wird in Abbildung 6.17 aufgezeigt. Die darin beschriebene Route weist zwei Segmente auf, die durch den Grad der nachbarschaftlichen Beziehung zwischen den einzelnen Basiselementen voneinander unterschieden werden können:

1. Segment, bestehend aus Basiselementen mit Nachbarn 1. Grades. Dieses Segment befindet sich im linken unteren Bereich der Routendarstellung.
2. Segment, bestehend aus Basiselementen mit Nachbarn 2. Grades. Dieses Segment erstreckt sich über den restlichen Routenverlauf. Im Detail betrachtet wechselt hier die Orientierung der gemeinsamen Seitenkante benachbarter Basiselemente von Y-Achsen-parallel zu X-Achsen-parallel und wieder zu Y-Achsen-parallel. Anm.: Die Reihenfolge

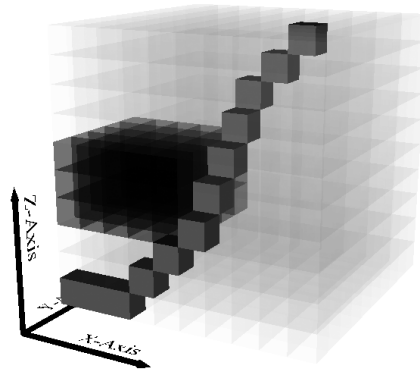


Abbildung 6.17: Route in einer 3-dimensionalen Datenbasis bei eingeschränkten Nachbarschaftsverhältnissen: Nur Nachbarschaften 1. und 2. Grades.

der Parallelitäten bezieht sich auf die Ausrichtung des Segmentes und beginnt mit dem Anschluss an das vorangegangene Segment und endet im rechten oberen Bereich der Routendarstellung.

Szenario 3

Im dritten und letzten Szenario werden alle verfügbaren Nachbarschaftsgrade für eine Suchbaumausbreitung genutzt. Durch die Nutzung des 3. Grades einer Nachbarschaft können zusätzlich zum vorhergegangenen Szenario Bewegungen diagonal durch den Raum genutzt werden.

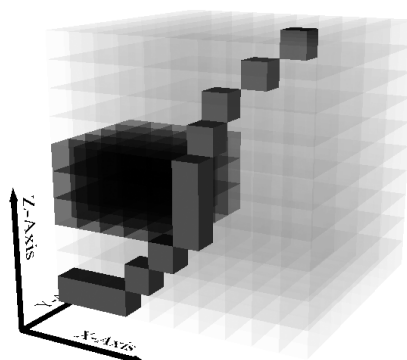


Abbildung 6.18: Route in einer 3-dimensionalen Datenbasis ohne Einschränkungen der Nachbarschaftsverhältnisse: Nachbarschaften 1., 2. und 3. Grades.

Die berechnete Route, unter Verwendung derselben Start- und Zielelemente aus den Szenarien 1 und 2, wird in Abbildung 6.18 dargestellt. Bezogen auf die zum Einsatz kommenden Nachbarschaften entlang der Route, können hier fünf Segmente identifiziert werden, wobei sich der Anfang der Route im linken unteren und das Ende im rechten oberen Bereich der Routendarstellung befindet:

1. Segment, bestehend aus Basiselementen mit Nachbarn 1. Grades.
2. Segment, bestehend aus Basiselementen mit Nachbarn 2. Grades.
3. Segment, bestehend aus Basiselementen mit Nachbarn 1. Grades.
4. Segment, bestehend aus Basiselementen mit Nachbarn 2. Grades.
5. Segment, bestehend aus Basiselementen mit Nachbarn 3. Grades.

Die hier angeführten Szenarien veranschaulichen das Ergebnis einer funktionellen Erweiterung der Routenplanung für den 3-dimensionalen Raum. Die Aufgabe, SPP im \mathfrak{R}^3 zu lösen, ist als logische Konsequenz zu den vorangegangenen Anwendungskonzepten – Multimodale Routenplanung und *Short-Cut Routing* – zu sehen. Dabei ist von Anwendungskonzept zu Anwendungskonzept ein Ansteigen der Komplexität des jeweiligen SPP zu erkennen, die sich jedoch auf Datenmodellierung und Nachbarschaftsanalyse beschränkt. Die Methode der Suchbaumausbreitung hingegen bleibt in all diesen Anwendungskonzepten unverändert und beruht ausschließlich auf einer monodirektionalen Suchbaumausbreitung nach *Dijkstra* (vgl. Abschnitt 2.2 bzw. 5.1).

Kapitel 7

Zusammenfassung, Diskussion und Ausblick

In diesem Kapitel werden die wesentlichen Erkenntnisse und Kernaussagen der Arbeit in Abschnitt 7.1 zusammengefasst und anschließend einer Diskussion unterzogen, in der diese Resultate kritisch hinterfragt werden (vgl. Abschnitt 7.2). Den Abschluss bildet ein Ausblick, Abschnitt 7.3, der für anknüpfende Arbeiten und Entwicklungen als Nukleus zu verstehen ist.

7.1 Zusammenfassung

Die Suche nach kürzesten Wegen stellt ein Forschungsfeld dar, das in der Literatur bereits von verschiedensten Ausgangsstellungen betreten und untersucht wurde. Wie in Kapitel 2 beschrieben, entstanden im Laufe der Geschichte eine Fülle an Algorithmen, die zur Lösung eines SPP ausgewählt werden können. Eine Unterscheidung zwischen den einzelnen Algorithmen kann hierbei in den seltensten Fällen anhand der prinzipiellen Lösungsstrategie getroffen werden (z.B.: Algorithmus nach *Floyd*, [Flo62] und Algorithmus nach *Dijkstra*, [Dij59]). Im Gegensatz dazu lassen sich in den Details der jeweiligen Algorithmusausführung sehr wohl Unterschiede erkennen (z.B.: *bottom-up* und *top-down* Suchbaumausbreitung bei hierarchischen Algorithmen), die zur Bedienung spezieller Anwendungen konzipiert sind.

Genau mit solchen Spezialanwendungen und Detailerweiterungen bestehender Methoden befasst sich diese Arbeit, wobei dieses teilweise auf sehr konzeptioneller Ebene geschieht. Dabei wird ein Bogen vom Aufbau der Datengrundlage in den Kapiteln 3 und 4 über die Mechanismen der Suchbaumausbreitung im allgemeinen in Kapitel 2 und in detaillierter Weise in Kapitel 5 bis hin zu beispielhaften Anwendungskonzepten in Kapitel 6 gespannt. Entlang dieses Bogens lassen sich wesentliche Bestandteile wie folgt finden:

7.1.1 Definition gemischter Datenstrukturen

Die Datengrundlage einer Suchbaumausbreitung besteht traditioneller Weise aus Basis- und Verbindungselementen, die geometrisch betrachtet jeweils einem einzigen Typ zugeordnet werden können: Ein Knoten-Kanten Modell, dessen Basiselemente durch Knoten und dessen Verbindungselemente durch Kanten repräsentiert werden; oder ein Kanten-Traversen

Modell, dessen Basiselemente durch Kanten und dessen Verbindungselemente durch Traversen repräsentiert werden.

In dieser Arbeit werden Verfahren aufgezeigt, mit denen eine Datengrundlage, bestehend aus Basis- und Verbindungselementen verschiedener Geometrietypen, gestaltet werden kann. Lassen sich für die gesamte Datengrundlage Basis- und Verbindungselemente finden, sodass eine explizite Nachbarschaftsbeschreibung vorliegt, wird diese Datengrundlage als *gemischt-einfache Struktur* bezeichnet (vgl. Abschnitt 3.3). Diese entsteht beispielsweise bei einer Mischung von Knoten-Kanten und Kanten-Traversen Modellen, wie sie im Anwendungskonzept *Multimodale Routenplanung* in Abschnitt 6.1 zum Einsatz kommt.

Nützt man Daten einer expliziten Nachbarschaftsbeschreibungsform in Kombination mit Daten einer impliziten Nachbarschaftsbeschreibungsform für eine gemeinsame Suchbaumausbreitung als *gemischt-komplexe Struktur* (vgl. Abschnitt 3.6), so sind jene Basiselemente gesondert zu kennzeichnen, die Übergangsstellen zwischen den unterschiedlichen Datenformen bilden: *Road Transition Element*. Diese Kennzeichnung geschieht im Hinblick auf die Methode der Nachbarschaftssuche während der Suchbaumausbreitung. Diese Art einer gemischten Datenstruktur bildet in Abschnitt 6.2 für ein *Short-Cut Routing* durch Kombination eines Kanten-Traversen Modells mit expliziter Nachbarschaftsbeschreibung mit einem polygonalen Modell mit impliziter Nachbarschaftsbeschreibung die Grundlage.

7.1.2 Modellierung zeitabhängiger Prozesse

Als eine Sonderform der Datenmodellierung wird in Kapitel 4 auf die *Beschreibung zeitabhängiger Vorgänge* vor dem Hintergrund öffentlicher Linienfahrpläne eingegangen. Der Zugang zum Modellierungsverfahren erfolgt hierbei graphisch orientiert (vgl. Abschnitt 4.2) und baut konzeptionell auf Bildfahrplänen des öffentlichen Verkehrs auf. Die Graphenstruktur eines Bildfahrplanes wird in einem 3-dimensionalen Kanten-Traversen Modell abgebildet, wobei die Rauminformation in der XY-Ebene und die Zeitinformation als *Julianisches Datum*, ein Dezimalsystem, entlang der Z-Achse geführt wird.

Das gewählte Kanten-Traversen Modell wird dabei so genutzt, dass zeitveränderliche Prozesse, wie das Warten an einer Haltestelle oder das Fahren mit einem Verkehrsmittel, als Kanten und der Wechsel zwischen diesen Prozessen, wie das Aussteigen aus dem Verkehrsmittel oder das Einsteigen in ein Verkehrsmittel, als Traversen beschrieben werden (vgl. Abschnitt 4.3). Besonders einfach ist dabei das Weiterfahren mit demselben Verkehrsmittel an einer Haltestelle modellierbar: Hier werden zwei zeitlich nicht direkt anschließende Kanten durch eine Traverse miteinander verbunden. Eine Unterscheidung zwischen dieser Prozesskombination und einem gleichzeitigen Warten an der Haltestelle ist dabei durchführbar. Zusätzlich sind alle Teilschritte eines Umstieges an einer Haltestelle modellierbar: Das Aussteigen aus dem ersten Verkehrsmittel, das Warten an der Haltestelle und das Einsteigen in das zweite Verkehrsmittel. Mit Hilfe dieser prozessscharfen Datengrundlage und der damit verbundenen Bewertungsmöglichkeit können Konsumenteninteressen bedarfsorientiert Berücksichtigung finden.

Ein solches Kanten-Traversen Modell zur Modellierung raumzeitlicher Beziehungen vor dem Hintergrund eines Fahrplannetzwerkes bildet für ein Anwendungskonzept – *Multimodale Routenplanung* – in Abschnitt 6.1 einen Teil der Datenbasis. Die unterschiedlichen Bewertungsmöglichkeiten verschiedener Prozesse und deren Kombinationen werden in diesem Anwendungskonzept als einzelne Szenarien abgebildet.

7.1.3 Multi Level Hopping

Ausgehend von Standardalgorithmen der hierarchischen Routenberechnung, beschrieben in Abschnitt 5.3, wird eine spezielle Situation im Detail betrachtet: Start und Ziel einer Routenberechnung, beides Elemente einer hierarchisch strukturierten Datengrundlage, sind nicht über eine durchgehende Verbindung in der höchsten Hierarchieebene miteinander verbunden. Als Beispiel einer höchsten Hierarchieebene, die keine durchgehende Verbindung aufweist, wird das österreichische Autobahn- und Schnellstraßennetz angeführt (vgl. Abbildung 5.9).

In Abschnitt 5.4 wird unter der Bezeichnung *Multi Level Hopping* ein Algorithmus vorgestellt, der auf diese nicht zusammenhängenden Elemente der höchsten Hierarchie für die Ausbreitung eines Suchbaumes Rücksicht nimmt. Diese Lösung bedingt ein Kanten-Traversen Modell nach Abschnitt 3.2 als Datengrundlage, wodurch eine eindeutige Zuweisung des Basiselementes zu einer hierarchischen Ebene erfolgen kann. In einem Knoten-Kanten Modell ist die Zuweisung eines Kreuzungsknotens zu einer hierarchischen Ebene kritisch zu sehen, sobald Straßenkanten unterschiedlicher Hierarchien von diesem Kreuzungsknoten abgehen.

Aufbauend auf diese eindeutige Zuordnung von Basiselementen zu einer hierarchischen Ebene, werden im Verlauf der Suchbaumausbreitung zusätzlich hierarchisch niedrigere Kanten vermerkt, die von einer aktuell zu untersuchenden Kante abgehen. Sobald der Algorithmus das Fehlen eines Zusammenhanges auf der aktuellen Hierarchieebene erkennt, und ein Aufstieg in eine höhere hierarchische Ebene nicht möglich ist, werden diese vorgemerkten Kanten (*Level Exit Arcs*) als neuer Ausgangspunkt für die Fortführung der Suchbaumausbreitung genutzt. Somit entfallen algorithmusbedingte Datenaufbereitungsprozesse, wie beispielsweise die Definition von *pseudo major roads* nach [Liu97].

7.1.4 Anwendungskonzepte für gemischte Datenstrukturen

Zwei der drei Anwendungskonzepte in Kapitel 6 beschreiben Bereiche, in denen eine einzelne Modellierungsvariante der Datenbasis nicht der gewünschten Funktionalität der Anwendung genügt. Im ersten Anwendungskonzept in Abschnitt 6.1 über die *Multimodale Routenplanung* wird ein Knoten-Kanten Modell zur Beschreibung eines Fußgängerwegenetzes mit einem Kanten-Traversen Modell zur Beschreibung eines Liniennetzwerkes des öffentlichen Verkehrs in kombinierter Weise eingesetzt. Bei beiden Modellen werden die Nachbarschaften explizit durch ihre Verbindungselemente beschrieben, weshalb keine gesonderte Anpassung des Suchbaumausbreitungsmechanismus zu erfolgen hat, lediglich die Übergangselemente zwischen den Datenmodellen müssen als zusätzliche Verbindungselemente definiert werden.

In Abschnitt 6.2 wird ein Anwendungskonzept vorgestellt, das 1-dimensionale Basiselemente mit Verbindungselementen (Kanten-Traversen Modell) und 2-dimensionale Basiselemente ohne Verbindungselement (polygonales Modell) für eine gemeinsame Suchbaumausbreitung nutzt: *Short-Cut Routing*. Durch diese gemischte Datengrundlage mit expliziten und impliziten Nachbarschaftsbeschreibungen muss die Nachbarschaftssuche innerhalb des Routing-Algorithmus so gestaltet sein, dass zum einen unter Verwendung des Verbindungselementes direkt Nachbarelemente identifiziert und zum anderen über spezielle räumliche Analysefunktionen – Berührung von Geometrieobjekten – Nachbarelemente gefunden werden können. Der Wechsel zwischen diesen Suchverfahren wird durch *Road Transition Elemente* gesteuert, die sowohl zur prinzipiellen Indikation eines Datenmodellüberganges dienen, als auch eine Information beinhalten, welches Basiselement des jeweils anderen Datenmodells zu erreichen ist.

Zusätzlich wird in diesem Abschnitt ein Bewertungsverfahren vorgestellt, mit dem Übergangskosten zwischen den einzelnen Basiselementen des polygonalen Modells berechnet werden können. Dadurch wird das Informationsdefizit im Vergleich zu den Verbindungselementbesitzenden Modellen kompensiert, denen Kostenwerte in ihren Kanten, bei Knoten-Kanten Modellen, beziehungsweise in ihren Traversen, bei Kanten-Traversen Modellen, zur Verfügung stehen.

7.1.5 Routenberechnung im Raum

Als konsequente Fortführung der Datenmodellierung bezogen auf die geometrische Repräsentationsform der jeweiligen Basiselemente, wird in Abschnitt 3.5 ein *polyhedrales Modell* vorgestellt. In weiterer Folge wird dieses in Abschnitt 6.3 als Datengrundlage für eine Routenberechnung eingesetzt. Dabei begründet sich die dafür gewählte Bezeichnung *³routing* (sprich: *Cube Routing*) in der geometrischen Form des 3-dimensionalen Basiselementes der Datengrundlage: dem Würfel.

Die Ausbreitung des Suchbaumes gestaltet sich komplexer als die Analyse eines ebenfalls verbindungselementefreien polygonalen Modells, da geometrische Analysefunktionen innerhalb der gewählten Geodateninfrastruktur, einem postgreSQL/PostGIS Datenbankmanagementsystem, größtenteils Koordinatenwerte entlang der Z-Achse nicht berücksichtigen. Aus diesem Grund wurde eine eigene Methode entwickelt, Nachbarelemente eines Zentralelementes aus der Gesamtmenge der Basiselemente zu identifizieren, seine relative Position zum Zentralelement zu beschreiben und für eine Fortführung der Suchbaumausbreitung bereit zu stellen.

Während der Suchbaumausbreitung kann das Wissen über die relative Position von Nachbarelementen selektiv genutzt werden: Dazu ist das Basiselement mit der Information, welche relative Positionen von Nachbarelementen zuzulassen sind, zu orientieren. Dieser Steuermechanismus während der Suchbaumausbreitung ist mit einer orientierten Kante, die Teil eines Knoten-Kanten Modelles ist, vergleichbar, mit dem Unterschied, dass das Basiselement und nicht das Verbindungselement die Richtungsinformation besitzt.

7.2 Diskussion

Der Diskussion ist folgendes vorwegzunehmen: Die hier gewählte Strategie zur Lösung eines SPP spiegelt nur eine von vielen wider. Das Spektrum an Lösungsstrategien ist mannigfaltig, kann aber größtenteils in zwei grundsätzlichen Herangehensweisen eingeteilt werden:

- Aufgabenbezogene Modellierung der Datenbasis: Die Funktionalität der Routenberechnung wird durch eine detaillierte Modellierung der Situation gewährleistet, auf der ein Standardalgorithmus angewandt werden kann.
- Problembezogene Komplexitätssteigerung des Algorithmus: Der Algorithmus wird durch Regelmechanismen so erweitert, dass dieser, angewandt auf eine allgemein konzipierte Datenbasis, dennoch ein ansprechendes Ergebnis liefert.

In beiden Fällen ist die Berechnung eines kürzesten Weges ein Balanceakt zwischen der Geschwindigkeit, mit der eine Lösung zu ermitteln ist, und der Funktionalität, mit der auf besondere Anforderungen des Benutzers Rücksicht zu nehmen ist. In den nachfolgenden Diskussionspunkten werden diese konträren Zugänge zur Routenberechnung konkret aufgezeigt.

7.2.1 Einsatz gemischt-komplexer Datenstrukturen

In Abschnitt 3.6 wird eine Kombination verschiedenartiger Datengrundlagen vorgestellt, die zum einen explizite Nachbarschaftsbeschreibungen und zum anderen implizite Nachbarschaftsbeschreibungen aufweisen. Diese Form der Datenbasis wird in Abschnitt 6.2 für das Anwendungskonzept *Short-Cut Routing* genutzt. Kernelement dieses Anwendungskonzeptes ist der basisdatenabhängige Ausbreitungsmechanismus des Suchbaumes: Entweder wird das Verbindungselement oder eine geometrische Analyse für das Finden von Nachbarelementen eingesetzt.

Eine alternative Lösungsstrategie zu jener aus Abschnitt 6.2 unter Verwendung einer gemischt-komplexen Datenstruktur nach Abschnitt 3.6 stellt eine Umbildung der polygonalen Basiselemente in ihren Regionsadjazenzgraph dar. Dieser Umbildungsprozess wird in Abschnitt 3.4 beschrieben und kann methodisch in gleicher Weise auf 3-dimensionale Basisdaten angewendet werden. Das Resultat dieser Umbildung besteht aus zwei Datensätzen: Einem 0-dimensionalen Basisdatensatz, der die vorherigen Basisdaten (Polygone oder Würfel) repräsentiert und einem 1-dimensionalen Verbindungsdatensatz, der richtungsabhängige Nachbarschaften beschreibt.

Stellt man beide Lösungsstrategien gegenüber, so findet man auf der Seite der gemischt-komplexen Datenstrukturen eine kleinere Datengrundlage, aber einen komplexeren Algorithmus und auf der Seite der Regionsadjazenzgraphenlösung eine größere Datengrundlage, aber eine einfachere Ausführung des Algorithmus. Der Unterschied zwischen einem komplexen und einem einfachen Algorithmus bezieht sich hier auf die Suche nach Nachbarelementen. Muss im komplexen Fall der Algorithmus entscheiden, ob ein Verbindungselement genutzt werden kann oder eine räumliche Analyse zur Nachbarschaftserkennung einzusetzen ist, stellt sich im einfachen Fall diese Frage nicht: Hier stehen ausschließlich Verbindungselemente zur Verfügung.

7.2.2 Multi Level Hopping

Die prinzipielle Strategie hierarchischer Suchbaumausbreitungsalgorithmen beruht auf einer stufenweisen, hierarchiebezogenen Ausdünnung der Datenbasis. Je nach hierarchischer Ebene, in der sich der Suchbaum aktuell ausbreitet, werden Basiselemente gleicher oder höherer Hierarchieebene für eine Fortführung der Ausbreitung eingesetzt.

Im Gegensatz dazu baut die Funktionalität des *Multi Level Hopping* Algorithmus (vgl. Abschnitt 5.4) auf der zusätzlichen Nutzung von Kanten niedrigerer Hierarchie auf: den *Level Exit Arcs*. Diese speziellen Kanten werden für eine situationsgesteuerte Fortführung der Suchbaumausbreitung in niedrigeren Hierarchieebenen benötigt. Der entstehende Funktionalitätsgewinn hat ein Anwachsen der Basisdatenmenge zur Folge, die im Zuge der Suchbaumausbreitung zu analysieren ist: Zusätzlich zu den Elementen gleicher und höherer Hierarchie, die in einem herkömmlichen hierarchischen Verfahren zu berücksichtigen sind, kommen hier noch die erwähnten *Level Exit Arcs* hinzu.

Dem Nachteil, mehr Daten analysieren zu müssen und somit mehr Zeit für eine Routenberechnung zu benötigen, stehen Vorteile aus dem Bereich der Datenmodellierung gegenüber. Einerseits müssen Elemente nicht Hierarchien zugeordnet werden, denen sie ursprünglich nicht angehören (vgl. *pseudo major roads*) und andererseits können bei kurzfristiger Einschränkung hierarchisch höherer Elemente, wie zum Beispiel bei einer unfallbedingten Sperre eines Autobahnabschnittes, lokale Umfahrungen ohne Manipulation der Datengrundlage gefunden werden.

7.2.3 Einsatz 3-dimensionaler Datenbasen zur Routenberechnung

In Abschnitt 3.5 werden 3-dimensionale Basisdaten beschrieben, die in weiterer Folge in Abschnitt 6.3 für ein spezielles Anwendungskonzept eingesetzt werden. Um dabei das Ziel, eine Route durch den \mathfrak{R}^3 vor dem Hintergrund der eingesetzten Infrastruktur (PostgreSQL/PostGIS) berechnen zu können, muss auf eine kompromissbereite Variante zur Modellierung der 3-dimensionalen Objekte eingegangen werden. Hierbei werden die würfelförmigen Volumenobjekte behelfsmäßig durch ihre acht Eckpunkte definiert, die nach [Pos10, Abschnitt 4.1.2] als Extended Well Known Binary (EWKB) beziehungsweise als Extended Well Known Text (EWKT) 3-dimensional beschrieben und gespeichert werden können. Zusätzlich muss in dieser Variante die Topologie dieser Eckpunkte für aufbauende Analysen und Visualisierungen gesondert vorgehalten werden.

Im Gegensatz dazu stellen Definitionen von Volumenobjekten, wie *POLYHEDRON* oder *IndexedFaceSet* (vgl. Abschnitt 3.5), topologisch konsistente Varianten dar, die jedoch laut [Pos10] im Datenbankmanagementsystem PostgreSQL/PostGIS als Geometrietypen nicht realisiert sind. Ab dem Zeitpunkt, ab dem solche Geometrietypen zur Verfügung stehen, werden komplexe Formen der Basiselemente zum Thema. Gehen mit dem die Entwicklung 3-dimensionaler Analysefunktionen einher, so kann für das Ausbreiten von Suchbäumen im \mathfrak{R}^3 nach einer Nachbarschaftsanalyse analog zur 2-dimensionalen Methode aus Abschnitt 6.2.2 verfahren werden.

7.3 Ausblick

Die erarbeiteten Modellierungserfahren und Algorithmen können als Ausgangspunkte zukünftiger Forschungsaktivitäten dienen. Die Erfahrungen aus dieser Arbeit, eingebracht als Wissensvorsprung in die untersuchten Bereiche der Modellierung und Algorithmen, bieten die Grundlage für einen Regelkreis, der eine iterative Weiterentwicklung und Verfeinerung der Themen unterstützt. Nachfolgend sind drei Bereiche aufgelistet, in denen Potential für weiterführende Entwicklungen identifizierbar ist:

- Konzeptionelle Weiterentwicklung.
- Technische Weiterentwicklungen.
- Anwendungsbezogene Weiterentwicklungen.

7.3.1 Konzeptionelle Weiterentwicklung

Das größte konzeptionelle Weiterentwicklungspotential findet sich im Bereich *Cube Routing* (vgl. Abschnitt 6.3). Hierzu wird bereits in Abschnitt 7.2.3 das Modellierungsverfahren der würfelförmigen Volumenobjekte und das Verfahren zur Nachbarschaftsanalyse angesprochen. Weiterentwicklungen sind dann unumgänglich, sobald passende Geometrietypen für Volumenobjekte und vollwertige 3-dimensionale Analysefunktionen zur Verfügung stehen. Es ist davon auszugehen, dass bei einer entsprechenden Umstellung der Datenvorhaltung und der eingesetzten Analyseverfahren eine spürbare Verbesserung in der Berechnungszeit zu erreichen ist.

7.3.2 Technische Weiterentwicklung

Eine technische Weiterentwicklung bezieht sich großteils auf die programmatische Umsetzung der Algorithmen. Vor dem Hintergrund der bisher eingesetzten Programmiersprache – pl/pgSQL (vgl. [The10b, Kapitel 38]) – sind für eine Verkürzung der Berechnungszeiten Alternativen zu untersuchen. Vorausgesetzt, die Infrastruktur (PostgreSQL/PostGIS) wird beibehalten, stehen laut [The10b, Kapitel 34, 39, 40 u. 41] folgende (z.T. prozedurale) Programmiersprachen zusätzlich zu pl/pgSQL zur Auswahl: C, pl/Tcl, pl/Perl und pl/Python. Die Aussage von [CARZ08, Kapitel 3], „[...] *Data types can be started also using high-level language PL/SQL but usually these implementations have a bad performance. [...]*“, legt es nahe, diesen Wechsel hin zu einer anderen Programmiersprache durchzuführen.

7.3.3 Anwendungsbezogene Weiterentwicklung

Bezugnehmend auf Kapitel 6 werden in dieser Arbeit ausschließlich Anwendungskonzepte auf konzeptioneller Ebene angeführt. Ziel einer anwendungsbezogenen Weiterentwicklung ist es, konkrete Anwendungsfelder zu identifizieren. Vor allem das *Cube Routing* in Kombination mit Strategien zur Lösung von SSMDP öffnet der Navigation das breit gefächerte Anwendungsfeld der Geosimulation speziell für 3-dimensionale Fragestellungen.

Speziell bei 3-dimensionalen Routenberechnungen muss die dritte Koordinatenkomponenten nicht ausschließlich zur Beschreibung von Raum genutzt werden. Beispielsweise wird in Kapitel 4 entlang dieser dritten Achse Zeitinformation aufgetragen, womit Fahrpläne modelliert werden. Diese thematische Freistellung der dritten Koordinatenachse der Datenbasis ist auf das *Cube Routing* übertragbar. Werden Zeitintervalle als eine denkbare Variante zur Nutzung der dritten Achse hierbei eingesetzt, so nähert man sich funktionell den *zellularen Automaten* an.

Für jegliche zukünftige Forschungstätigkeit im Umfeld dieser Arbeit, egal ob konzeptionell, technisch oder anwendungsspezifisch, ist folgende Erkenntnis nicht außer Acht zu lassen: Obwohl die Routenberechnung neben Positionierung und Zielführung als ein zentrales Thema der Navigation gilt, ist sie thematisch sehr eng mit der Geoinformatik verbunden. Würde man die Berechnung des kürzesten Weges als eine Form der räumlichen Analyse werten, so könnte man die Routenberechnung auch als eine Kernaufgabe der Geoinformatik sehen. Jedenfalls ist eine strenge disziplinäre Trennung von Navigation und Geoinformatik für Fragestellungen, wie sie in dieser Arbeit behandelt werden, nicht zielführend. Gleichgültig in welcher Form eine Weiterentwicklung erfolgt, es sind beide Disziplinen stets einander ergänzend und sich gegenseitig unterstützend einzusetzen.

Literaturverzeichnis

- [ABD⁺10] ACHOUR, M., F. BETZ, A. DOVGAL, N. LOPES, H. MAGNUSSON, G. RICHTER, D. SEGUY und J. CRANA: *PHP-Handbuch*. 2010. <http://www.php.net/manual/de/>; zuletzt besucht: 07.10.2010.
- [ACD⁺03] ADRIAENSEN, F., J. P. CHARDON, G. DE BLUST, E. SWINNEN, S. VILLALBA, H. GULINCK und E. MATTHYSEN: *The application of 'least-cost' modelling as a functional landscape model*. *Landscape and Urban Planning*, 64:233–247, 2003.
- [AK96] AURENHAMMER, F. und R. KLEIN: *Voronoi Diagrams*. Technical Report 198, FernUniversität Hagen, Department of Computer Science, Germany, 1996.
- [AP08] ABDUL-RAHMAN, A. und M. PILOUK: *Spatial Data Modelling for 3D GIS*. Springer Berlin Heidelberg New York, 2008.
- [ASv05] ARENS, C., J. STOTER und P. VAN OSTEROM: *Modelling 3D spatial objects in a geo-DBMS using a 3D primitive*. *Computer and Geosciences*, 31(2):165–177, 2005.
- [Aut10] AUTODESK: *AutoCAD 2010 User Documentation; Using the snapping environment*. 2010. <http://docs.autodesk.com/ACD/2010/ENU/AutoCAD%202010%20User%20Documentation/index.html?url=WS1a9193826455f5ffa23ce210c4a30acaf-714b.htm,topicNumber=d0e52357>; zuletzt besucht: 11.10.2010.
- [Bar91] BARTELME, N.: *Datenmodelle für Netzwerk-Applikationen in GIS*. Wiener Schriften zur Geographie und Kartographie - GIS und Kartographie, 6:17–22, 1991.
- [Bar05] BARTELME, N.: *Geoinformatik – Modelle, Strukturen, Funktionen*. Springer Verlag, 4., vollst. überarb. Auflage, 2005.
- [Bar08] BARROW, J. D.: *Einmal Unendlichkeit und zurück*. Rowohlt Taschenbuch Verlag, 2008.
- [BDS⁺10] BAUER, R., D. DELLING, P. SANDERS, D. SCHIEFERDECKER, D. SCHULTES und D. WAGNER: *Combining Hierarchical and Goal-Directed Speed-up Techniques for Dijkstra's Algorithm*. *Journal of Experimental Algorithmics*, 15:69–88, 2010.
- [Beh06] BEHRINGER, W.: *Kartenwelten – Der Raum und seine Repräsentation in der Neuzeit*, Kapitel Mit der Karte auf Grand Tour. *Infrastrukturbildung und Mental Mapping in der Frühen Neuzeit*, Seiten 77–93. Primus Verlag Darmstadt, 2006.

- [BG09] BOGUSLAWSKI, P. und C. GOLD: *3D Geo-Information Sciences*, Kapitel Construction Operators for Modelling 3D Objects and Dual Navigation Structures. Springer Berlin Heidelberg, 2009.
- [Bit10] BITMANAGEMENT SOFTWARE GMBH: *Bitmanagement - Interactive Web3D Graphics*. 2010. <http://www.bitmanagement.com/>; zuletzt besucht: 07.10.2010.
- [BK09] BORTENLÄNGER, C. und U. KIRSTEIN: *Börse für Dummies*. Wiley-Vch Verlag GmbH u. Co. KGaA, 2. Auflage, 2009.
- [BNK09] BECKER, T., C. NAGEL und T. H. KOLBE: *3D Geo-Information Sciences*, Kapitel A Multilayered Space-Event Model for Navigation in Indoor Spaces. Springer Berlin Heidelberg, 2009.
- [BRVI00] BOGAERT, J., R. ROUSSEAU, P. VAN HECKE und I. IMPENS: *Alternative area-perimeter ratios for measurement of 2D shape compactness of habitats*. Applied Mathematics and Computation, 111:71–85, 2000.
- [Cal61] CALDWELL, T.: *On Finding Minimum Routes in a Network With Turn Penalties*. Communications of the ACM, 4(2):107–108, 1961.
- [Car97] CAR, A.: *Hierarchical Spatial Reasoning: Theoretical Consideration and its Application to Modeling Wayfinding*. Doktorarbeit, Technical University Vienna, Austria, 1997.
- [CARZ08] CHEN, T. K., A. ABDUL-RAHMAN und S. ZLATANOV: *3D Spatial Operations for Geo-DBMS: Geometry vs. Topology*. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 37(B2):549–554, 2008.
- [CF94] CAR, A. und A. U. FRANK: *General principles of hierarchical spatial reasoning—the case of way-finding*. In: *Proceedings of the sixth international symposium on spatial data handling*, Band 1, Seiten 646–664, 1994.
- [CGR93] CHERKASSKY, B. V., A. V. GOLDBERG und T. RADZIK: *Shortest Paths Algorithms: Theory And Experimental Evaluation*. Mathematical Programming, 73:129–174, 1993.
- [CH91] CLARK, J. und D. A. HOLTON: *A First Look at Graph Theory*. World Scientific, 1991.
- [CTB01] CAR, A., G. TAYLOR und C. BURNSDON: *An analysis of the performance of a hierarchical wayfinding computational model using synthetic graphs*. Computers, Environment and Urban Systems, 25:69–88, 2001.
- [De 83] DE CHAMPEAUX, D.: *Bidirectional heuristic search again*. Journal of the Association for Computing Machinery, 30(1):22–32, 1983.
- [Dia69] DIAL, B. R.: *Algorithm 360: shortest path forest with topological ordering*. Communications of the Association for Computing Machinery, 12(11):632–633, 1969.
- [Dij59] DIJKSTRA, E. W.: *A Note on Two Problems in Connexion with Graphs*. Numerische Mathematik, 1:269–271, 1959.

- [DN94] DILLENBURG, J. und P. NELSON: *Perimeter search*. Artificial Intelligence, 65(1):165–178, 1994.
- [DN95] DILLENBURG, J. F. und P. C. NELSON: *Improving Search Efficiency Using Possible Subgoals*. Mathematical and Computer Modelling, 22(4-7):397–414, 1995.
- [Dom07] DOMSCHKE, W.: *Logistik: Transport; Grundlagen, lineare Transport- und Umladeprobleme*. R. Oldenburg Verlag München, Wien, 5. Auflage, 2007.
- [DP84] DEO, N. und C. PANG: *Shortest-Path Algorithms: Taxonomy and Annotation*. Networks, 14:275–323, 1984.
- [Dre69] DREYFUS, E. S.: *An appraisal of some shortest path algorithms*. Operations Research, 17:395–412, 1969.
- [DS77] DE CHAMPEAUX, D. und L. SINT: *An improved bi-directional heuristic search algorithm*. Journal of the Association for Computing Machinery, 24(2):177–191, 1977.
- [Dun10] DUNGL, M.: *Historischer Grenzverlauf Österreichs*. E-mailkorrespondenz am 19.08.2010, 2010.
- [EI56] EINSTEIN, A. und L. INFELD: *Die Evolution der Physik; Von newton bis zur Quantentheorie*. Rowohlt, 1956.
- [Eng10] ENGELS, B.: *The Transitive Minimum Manhattan Subnetwork Problem in 3 dimensions*. Discrete Applied Mathematics, 158:298–307, 2010.
- [Env10a] ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (ESRI): *ArcGIS Desktop 9.3 Help; Cost Distance algorithm*. 2010. http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?id=5946&pid=5941&topicname=Cost_Distance_algorithm; zuletzt besucht: 16.04.2010.
- [Env10b] ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE (ESRI): *ArcGIS Desktop 9.3 Help; Using the snapping environment*. 2010. http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=Using_the_snapping_environment; zuletzt besucht: 11.10.2010.
- [ESM93] EGENHOFER, M., J. SHARMA und D. MARK: *A Critical Comparison of the 4-Intersection and 9-Intersection Models for Spatial Relations: Formal Analysis*. In: *Proceedings of the International Symposium On Computer-Assisted Cartography – Auto-Carto XI*, Seiten 1–11, 1993.
- [Flo62] FLOYD, R. W.: *Algorithm 97: Shortest path*. Communications of the ACM, 5(6):345, 1962.
- [FNSE10] FRIESE, E., L. NIERADZIK, A. STRUNK und H. ELBERN: *Eyjafjalla – Ausbreitung der Aschewolke über Europa*. 2010. <http://db.eurad.uni-koeln.de/promote/RLAQS/volcano.html>; zuletzt besucht: 21.04.2010.
- [FSR06] FU, L., D. SUN und L. R. RILETT: *Heuristic shortest path algorithms for transportation applications: State of the art*. Computers and Operations Research, 33:3324–3343, 2006.

- [Fu96] FU, L.: *Real-time vehicle routing and scheduling in dynamic and stochastic traffic networks*. Doktorarbeit, University of Alberta, Canada, 1996.
- [Gam10] GAMILLSCHEG, H.: *Vulkanausbruch: Tausende Flüge weltweit abgesagt*. Die Presse, Print-Ausgabe vom 16.04.2010, 2010. <http://diepresse.com/home/panorama/welt/558680/index.do>; zuletzt besucht: 21.04.2010.
- [GB05] GRITZMANN, P. und R. BRANDENBERG: *Das Geheimnis des kürzesten Weges; Ein mathematisches Abenteuer*. Springer Verlag Berlin, Heidelberg, 3., überarbeitete Auflage, 2005.
- [GKP85] GLOVER, F., D. KLINGMAN und N. A. PHILIPS: *A new Polynomial bounded shortest path algorithm*. Operations Research, 33:65–73, 1985.
- [GLN01] GUDMUNDSSON, J., C. LEVOPOULOS und G. NARASIMHAN: *Approximating a Minimum Manhattan Network*. Nordic Journal of Computing, 8(2):219–232, 2001.
- [GM07] GUTIÉRREZ, E. und A. L. MEDAGLIA: *Labeling algorithm for the shortest path problem with turn prohibitions with application to large-scale road networks*. Annals of Operations Research, 157:169–182, 2007.
- [Gro10] GROSSGLOCKNER HOCHALPENSTRASSEN AKTIENGESELLSCHAFT: *Großglockner Hochalpenstraße; Service*. 2010. <http://www.grossglockner.at/de/service/>; zuletzt besucht: 19.08.2010.
- [GW09] GUO, J. und L. WU: *3D Geo-Information Sciences, Kapitel A Topological Analysis Method for 3D Geo-Entities Structured as Hexahedron Tessellations*. Springer Berlin Heidelberg, 2009.
- [Hal97] HALSTED, G. B.: *Biography James Joseph Sylvester, LL. D., F. R. S*. The American Mathematical Monthly, 4(6):159–168, 1897.
- [HJR97] HUANG, Y.-W., N. JING und E. A. RUNDENSTEINER: *A Hierarchical Path View Model for Path Finding in Intelligent Transportation Systems*. Geoinformatica, 1(2):125–159, 1997.
- [HLC01] HOFMANN-WELLENHOF, B., H. LICHTENEGGER und J. COLLINS: *GPS; Theory and Practice*. Springer Wien, New York, 5. revised Auflage, 2001.
- [HLW03] HOFMANN-WELLENHOF, B., K. LEGAT und M. WIESER: *Navigation; Principles of positioning and guidance*. Springer Wien, New York, 2003.
- [HNR68] HART, E. P., N. J. NILSSON und B. RAPHAEL: *A formal basis for the heuristic determination of minimum cost paths*. IEEE Transaction, System Science and Cybernetics, SSC-4(2):100–107, 1968.
- [HTB95] HRIBAR, M., V. E. TAYLOR und D. E. BOYCE: *Choosing a Shortest Path Algorithm*. Technischer Bericht, 1995.
- [Int01] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION: *Geographic information – Spatial schema*. ISO/DIS 19107, ISO/TC 211, 2001.

- [Int02] INTERNATIONAL ORGNIZATION FOR STANDARDIZATION: *Geographic information – Temporal schema*. ISO/DIS 19108, ISO/TC 211, 2002.
- [Int04] INTERNATIONAL ORGNIZATION FOR STANDARDIZATION: *Data elements and interchange formats – Information interchange – Representation of dates and times*. ISO 8601, ISO/TC 154, 2004.
- [Int08] INTERNATIONAL ORGNIZATION FOR STANDARDIZATION: *Computer graphics and image processing – Extensible 3D (X3D)*. ISO/IEC 19775-1, 2008.
- [ITS10] ITS VIENNA REGION: *AnachB.at*. 2010. <http://www.anachb.at/>; zuletzt besucht: 08.10.2010.
- [JSQ02] JAGADEESH, G. R., T. SRIKANTHAN und K. H. QUEK: *Heuristic Techniques for Accelerating Hierarchical Routing on Road Networks*. IEEE Transactions on Intelligent Transportation Systems, 3(4):301–309, 2002.
- [Jun05] JUNGNICHEL, D.: *Graphs, Networks and Algorithms*. Springer Verlag Berlin Heidelberg New York, 2 Auflage, 2005.
- [Kad07] KADA, M.: *Zur maßstabsabhängigen Erzeugung von 3D-Stadtmodellen*. Doktorarbeit, Universität Stuttgart, Deutschland, 2007.
- [Kar96] KARIMI, H. A.: *Real-time optimal route computation: a heuristic approach*. Journal of Intelligent Transportation Systems, 3(2):111–127, 1996.
- [KCY93] KAUFMANN, A., D. COHEN und R. YAGEL: *Volume Graphics*. IEEE Computer, 26(7):51–64, 1993.
- [Kis09] KISSLING, S.: *Angewandte Geoinformatik 2009*, Kapitel GIS-basierte Konnektivitätsanalysen für die Biotopverbundplanung, Seiten 549–557. Wichmann, Heidelberg, 2009.
- [KK93] KÖLL, A. und H. KAINDL: *Bidirectional best-first search with bounded error: Summary of results*. In: *IJCAI-93 Proceedings*, Seiten 217–223, 1993.
- [KK97] KAINDL, H. und G. KAINZ: *Bidirectional Heuristic Search Reconsidered*. Journal of Artificial Intelligence Research, 7:283–317, 1997.
- [KM08a] KONRAD, H. und W. MADERTHANER (Herausgeber): *... der Rest ist Österreich: das Werden der Ersten Republik*, Band 1. Gerold, 2008.
- [KM08b] KONRAD, H. und W. MADERTHANER (Herausgeber): *... der Rest ist Österreich: das Werden der Ersten Republik*, Band 2. Gerold, 2008.
- [Kor85] KORF, R. E.: *Depth-first iterative deeping: an optimal admissible tree search*. Artificial Intelligence, 27(1):97–109, 1985.
- [Kor87] KORF, R. E.: *Planning as search: a quantitative approach*. Artificial Intelligence, 33(1):65–68, 1987.
- [Kwa89] KWA, J. B. H.: *BS*: An admissible bidirectional staged heuristic search algorithm*. Artificial Intelligence, 38(1):95–109, 1989.

- [Lan09] LANGLOIS, P.: *The Modelling Process in Geography*, Kapitel Cellular Automata for Modeling Spatial Systems. ISTE Ltd. and John Wiley and Sons, Inc., 2009.
- [Lee04] LEE, J.: *Geographic Information Science*, Band 3234/2004, Kapitel 3D GIS for Geocoding Human Activity in Microscale Urban Environments. Springer Berlin Heidelberg, 2004.
- [Liu96] LIU, B.: *Intelligent route finding: combining knowledge, cases and an efficient search algorithm*. In: *12th European Conference on Artificial Intelligence*, Seiten 380–384, 1996.
- [Liu97] LIU, B.: *Route finding by using knowledge about the road network*. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 27(4):436–448, 1997.
- [Lys95] LYSGAARD, J.: *A two-phase shortest path algorithm for networks with node coordinates*. European Journal of Operational Research, 87(2):368–374, 1995.
- [Mac95] MACFARLANE, A.: *Biography Arthur Cayley*. The American Mathematical Monthly, 2(4):99–102, 1895.
- [Man95] MANZINI, G.: *BIDA*: an improved perimeter search algorithm*. Artificial Intelligence, 75(2):347–360, 1995.
- [Mas07] MASTIN, L. G.: *A user-friendly one-dimensional model for wet volcanic plumes*. Geochemistry, Geophysics, Geosystems, 8(3):1–24, 2007.
- [Moo57] MOORE, E. F.: *The shortest path through a maze*. In: *Proceedings of the International Symposium on Theory of Switching*, Seiten 285–292. Harvard University Press Cambridge, 1957.
- [Mun95] MUNKRES, J. R.: *Elements of Algebraic Topology*. Perseus Books, New Ed Auflage, 1995.
- [NH06] NUHN, H. und M. HESSE: *Verkehrsgeographie*. Ferdinand Schöningh Verlag Paderborn, München, Wien, Zürich, 2006.
- [Nic66] NICHOLSON, J. A. T.: *Finding the shortest route between two points in a network*. Computer Journal, 6:275–280, 1966.
- [Nil71] NILSSON, N. J.: *Problem-solving methods in artificial intelligence*. McGraw-Hill, New York, 1971.
- [Nil80] NILSSON, N. J.: *Principles of Artificial Intelligence*. Tioga, Palo Alto, CA, USA, 1980.
- [OBS94] OKABE, A., B. BOOTS und K. SUGIHARA: *Nearest neighbourhood operations with generalized Voronoi diagrams: A review*. International Journal of Geographical Information Science, 8(1):43–71, 1994.
- [Oes10] OESTERREICHISCHE NATIONALBIBLIOTHEK: *Sammlungen von Handschriften und alten Drucken*. 2010. <http://www.onb.ac.at/about/11777.htm>; zuletzt besucht: 16.03.2010.

- [OGP10] OGP SURVEY AND POSITIONING COMMITTEE: *Survey and Positioning*. 2010. <http://www.epsg.org>; zuletzt besucht: 27.08.2010.
- [Ope06a] OPEN GEOSPATIAL CONSORTIUM: *OpenGIS Implementation Specification for Geographic information – Simple feature access – Part 1: Common architecture*. OGC 06-103r3, Version 1.2.0, 2006.
- [Ope06b] OPEN GEOSPATIAL CONSORTIUM: *OpenGIS Implementation Specification for Geographic information – Simple feature access – Part 2: SQL option*. OGC 06-104r3, Version 1.2.0, 2006.
- [Pal84] PALLOTTINO, S.: *Shortest-Path Methods: Complexity, Interrelations and New Propositions*. *Networks*, 14:257–267, 1984.
- [Pap74] PAPE, U.: *Implementation and Efficiency of Moore Algorithms for the Shortest Root Problem*. *Mathematical Programming*, 7:212–222, 1974.
- [Pea84] PEARL, J.: *Heuristics: intelligent search strategies for computer problem solving*. Addison-Wesley Publishing Company, 1984.
- [PK09] PIETSCH, M. und M. KRÄMER: *Angewandte Geoinformatik 2009*, Kapitel Analyse der Verbundsituation von Habitaten und -strukturen unter Verwendung graphentheoretischer Ansätze am Beispiel dreier Zielarten, Seiten 564–573. Wichmann, Heidelberg, 2009.
- [Poh71] POHL, I.: *Bi-directional search*. *Machine Intelligence*, 6:127–140, 1971.
- [Pos10] POSTGIS: *PostGIS 1.5.1 Manual*. 2010. <http://postgis.refractor.net/download/postgis-1.5.1.pdf>; zuletzt besucht: 20.04.2010.
- [PP84] POLITOWSKI, G. und I. POHL: *D-node retargeting in bidirectional heuristic search*. In: *AAAI-84 Proceedings*, Seiten 274–277, 1984.
- [PP06] PIJLS, W. und H. POST: *Bidirectional A*: Comparing balanced and symmetric heuristic methods*. Technischer Bericht EI 2006-41, Econometric Institut, 2006.
- [RR97] RITTER, N. und M. RUTH: *The GeoTIFF data interchange standard for raster geographic images*. *International Journal of Remote Sensing*, 18(7):1637–1647, 1997.
- [Ruc10] RUCH, M.: *Hochwaldbahn*. 2010. http://www.hochwaldbahn.info/hochwaldbahn_info/fahrplaene/fahrplaene.html; zuletzt besucht: 16.08.2010.
- [Sac74] SACERDOTI, E. D.: *Planning in a hierarchy of abstraction spaces*. *Artificial Intelligence*, 5(2):115–135, 1974.
- [Sch05] SCHULTE, C.: *Logistik; Wege zur Optimierung der Supply Chain*. Verlag Franz Vahlen, München, 4. Auflage, 2005.
- [Stü93] STÜCKLBERGER, A.: *Optimierungs- und Generalisierungsstrategien in digitalen Straßendatenbanken*, 1993. Diplomarbeit an der Technischen Universität Graz, Österreich.

- [Sv02] STOTER, J. und P. VAN OOSTEROM: *Incorporating 3D geo-objects into a 2D geo-DBMS*. Proceedings FIG, ACSM/ASPRS, Washington D.C. 2002, 2002.
- [SW93] STÜCKELBERGER, A. und M. WIESER: *GIS und „Multilevel“-Routensuche*. In: *Grazer Geoinformatik-Tage '95 – GIS in Transport und Verkehr. Mitteilungen der geodätischen Institute der Technischen Universität Graz*, Band 80, Seiten 9–16, 1993.
- [TEF97] TIMPF, S., M. EGENHOFER und A. FRANK: *Ein konzeptionelles Modell für Navigation in Fernstraßennetzen*. *Zeitschrift für Vermessungswesen*, 122(2):55–77, 1997.
- [Tel07] TELE ATLAS: *Tele Atlas MultiNet Shapefile 4.3.2.2 Format Specifications*, 2007. Version 1.1.
- [The10a] THE MAPSERVER TEAM: *MapServer Documentation, Release 5.6.5*. 2010. <http://mapserver.org/MapServer.pdf>; zuletzt besucht: 07.10.2010.
- [The10b] THE POSTGRESQL GLOBAL DEVELOPMENT GROUP: *PostgreSQL 8.4.3 Documentation*. 2010. <http://www.postgresql.org/files/documentation/pdf/8.4/postgresql-8.4.3-A4.pdf>; zuletzt besucht: 20.04.2010.
- [Thi09] THILL, J.-C.: *Manual of Geographic Information Systems*, Kapitel Transportati-on Applications of Geographic Information Systems. American Society for Photogrammetry and Remote Sensing, 2009.
- [Wie91] WIESER, M.: *Wesen und Nutzen von Optimierungsstrategien am Beispiel des AIS*. *Mitteilung der geodätischen Institute der Technischen Universität Graz*, 70:57–68, 1991.
- [Wik10] WIKIPEDIA: *Österreich*. 2010. <http://de.wikipedia.org/wiki/Österreich>; zuletzt besucht: 19.08.2010.
- [Wil64] WILLIAMS, J. W. J.: *Algorithm 232: Heapsort*. *Communications of the Association for Computing Machinery*, 7(6):347–348, 1964.
- [Win02] WINTER, S.: *Modeling Costs of Turns in Route Planning*. *GeoInformatica*, 6(4):345–361, 2002.
- [YWH06] Y. CHEN, B. W. WAH und C.-W. HSU: *Temporal Planning using SubgoalPartitioning and Resolution in SGPlan*. *Journal of Artificial Intelligence Research*, 26:323–369, 2006.
- [Zha97] ZHAO, Y.: *Vehicle Location and Navigation Systems*. Artech House, Inc. Boston, London, 1997.
- [Zla00] ZLATANOVA, S.: *3D GIS for Urban Development*. Doktorarbeit, Graz University of Technology, Austria, 2000.
- [ZN98] ZHAN, F. B. und C. E. NOON: *Shortest Path Algorithms: An Evaluation using Real Road Networks*. *Transportation Science*, 32(1):65–73, 1998.

Abbildungsverzeichnis

2.1	Knoten-Kanten Struktur mit darin ausgebreitetem Suchbaum zur Ermittlung einer Route zwischen dem Startelement n_S und dem Zielelement n_D	12
3.1	Verbindungselement zwischen expliziten und impliziten Datenmodellen.	30
4.1	Zeitabhängige Darstellung von Finanz- und Anlagebedarf in einem 2-dimensionalen Thema/Zeit Diagramm. Quelle: [BK09, S. 331].	38
4.2	Zeitabhängige Darstellung eines Fallexperimentes in einem 2-dimensionalen Raum/Zeit Diagramm. Quelle: [EI56, S. 139].	39
4.3	Zeitabhängige Darstellung über die Entwicklung des Universums in einem 3-dimensionalen Raum/Zeit Diagramm. Quelle: [Bar08, S. 112].	39
4.4	Ausschnitt eines Bildfahrplanes der Deutschen Bundesbahn, Bundesbahndirektion Saarbrücken aus dem Jahr 1970/71. Quelle: [Ruc10].	40
4.5	Situationsdarstellung einer einfachen Haltestelle H_n	41
4.6	Situationsdarstellung einer komplexen Haltestelle H_n	45
4.7	Situationsdarstellung einer Haltestelle H_n mit anschließendem Individualverkehrsnetz über den Wegeknoten N_n	47
5.1	Ausgangsdaten – Kanten einer Kanten-Traversen Struktur.	52
5.2	Monodirektionale Suchbaumausbreitung vom Startelement 230 zum Zielelement 300.	52
5.3	Monodirektionale Suchbaumausbreitung vom Zielelemente 300 zum Startelement 230.	53
5.4	Abbruchkriterium bei bidirektionaler Suchbaumausbreitung – Überlagerung des Startsuchbaumes um Kante 230 und des Zielsuchbaumes um Kante 300 bei Kante 293.	56
5.5	Bidirektionale Suchbaumausbreitung vom Startelement 230 zum Zielelement 300.	56
5.6	Querschnitt eines hierarchisch strukturierten Datensatzes.	59
5.7	Ausgangsdaten – Kanten einer Kanten-Traversen Struktur mit hierarchischer Zusatzinformation.	60
5.8	Hierarchische Suchbaumausbreitung vom Startelement 230 zum Zielelement 300.	62
5.9	Österreichisches Autobahn- und Schnellstraßennetz. Datengrundlage: Tele Atlas Straßennetz (_nw) beschränkt auf $FRC = 0$ – Stand 2008; Tele Atlas Administrative Grenzen (_a0) – Stand 2008; NASA SRTM-3 – Stand 2000.	64

5.10	Route zwischen Kante 175 und Kante 186 zur Erläuterung der <i>Level Exit Arcs</i> .	67
5.11	Ausgangsdaten – Kanten einer Kanten-Traversen Struktur mit hierarchischer Zusatzinformation. Einschränkung der Daten auf einen definierten Bereich, wodurch eine Verbindung der Elemente innerhalb der höchsten Hierarchieebene nicht mehr gegeben ist.	68
5.12	Multi Level Hopping Suchbaumausbreitung von Startelement 230 zum Ziel-element 300 ohne Verwendung eines LEA Elementes in der resultierenden Route.	69
5.13	Multi Level Hopping Suchbaumausbreitung von Startelement 1261 zum Ziel-element 1380 mit Verwendung zweier LEA Elemente in der resultierenden Route.	69
6.1	Basiselemente einer gemischt-einfachen Struktur bestehend aus Knoten des Wegenetzes für Fußgänger und Kanten des Liniennetzes des öffentlichen Verkehrs.	73
6.2	Basis- und Verbindungselemente einer gemischt-einfachen Struktur bestehend aus Knoten und Kanten des Wegenetzes für Fußgänger und Kanten des Liniennetzes des öffentlichen Verkehrs.	73
6.3	2-dimensionales Raum/Zeit Diagramm des Liniennetzes des öffentlichen Verkehrs.	74
6.4	Multimodale Route zwischen Knoten 2 und Knoten 43 mit dem Startzeitpunkt $T_S = 3$	75
6.5	Multimodale Route zwischen Knoten 2 und Knoten 43 mit dem Startzeitpunkt $T_S = 10$	76
6.6	Multimodale Route zwischen Knoten 2 und Knoten 43 mit dem Startzeitpunkt $T_S = 13$	77
6.7	Multimodale Route zwischen Knoten 4 und Knoten 43 mit günstigen Einsteige- und Aussteigekosten in das Verkehrsmittel.	78
6.8	Multimodale Route zwischen Knoten 4 und Knoten 43 mit teuren Einsteige- und Aussteigekosten in das Verkehrsmittel.	78
6.9	Basiselemente einer gemischt-komplexen Struktur bestehend aus Kanten des Wegenetzes und Polygone des freien Geländes.	80
6.10	Definierte Übergänge zwischen Wegenetz und freiem Gelände (RTE). RTE-Paare im linken oberen und rechten unteren Datenbereich.	80
6.11	Modellierung von Übergangskosten ${}^c t$ zwischen dem Zentralpolygon p_c und den Nachbarpolygonen $\{p_1, \dots, p_8\}$ in Abhängigkeit der Nachbarschaft. . . .	82
6.12	Short-Cut Route zwischen Kante 1 und Kante 16 bei günstigeren Kanten- und Traversennutzungskosten als Polygonnutzungskosten und Kostengleichheit innerhalb der Kanten-Traversen Struktur.	83
6.13	Short-Cut Route zwischen Kante 1 und Kante 16 bei günstigeren Polygonnutzungskosten als Kanten- und Traversennutzungskosten und Kostengleichheit innerhalb der polygonalen Datenstruktur.	84
6.14	Short-Cut Route zwischen Kante 1 und Kante 16 bei günstigeren Polygonnutzungskosten als Kantennutzungskosten und ungleiche Kosten innerhalb der polygonalen Datenstruktur.	84
6.15	Basiselemente einer polyhedralen Struktur.	86

6.16	Route in einer 3-dimensionalen Datenbasis bei eingeschränkten Nachbarschaftsverhältnissen: Nur Nachbarschaften 1. Grades.	89
6.17	Route in einer 3-dimensionalen Datenbasis bei eingeschränkten Nachbarschaftsverhältnissen: Nur Nachbarschaften 1. und 2. Grades.	90
6.18	Route in einer 3-dimensionalen Datenbasis ohne Einschränkungen der Nachbarschaftsverhältnisse: Nachbarschaften 1., 2. und 3. Grades.	90

Tabellenverzeichnis

2.1	Relationale Darstellung der Eigenschaften von Basiselementen.	9
2.2	Relationale Darstellung der Eigenschaften von Verbindungselementen.	10
3.1	Basis- und Verbindungselement in einer Knoten-Kanten Struktur.	16
3.2	Relationale Darstellung der Eigenschaften von Basiselementen in einer Knoten-Kanten Struktur.	16
3.3	Relationale Darstellung der Eigenschaften von Verbindungselementen in einer Knoten-Kanten Struktur.	17
3.4	Basis- und Verbindungselement in einer Kanten-Traversen Struktur.	18
3.5	Relationale Darstellung der Eigenschaften von Basiselementen in einer Kanten-Traversen Struktur.	18
3.6	Relationale Darstellung der Eigenschaften von Verbindungselementen in einer Kanten-Traversen Struktur.	19
3.7	Basis- und Verbindungselement in einer gemischt-einfachen Struktur.	19
3.8	Basis- und Verbindungselement in einer polygonalen Struktur.	21
3.9	Relationale Darstellung der Eigenschaften von Basiselementen in einer polygonalen Struktur.	21
3.10	Bewertungsverfahren bei polygonalen Struktur.	22
3.11	Basis- und Verbindungselement in einer Knoten-Kanten Struktur abgeleitet aus einer polygonalen Struktur.	22
3.12	Basis- und Verbindungselement in einer polyhedralen Struktur.	24
3.13	Modellierung von polyhedralen Elementen.	26
3.14	Gegenüberstellung von Nomenklaturen zu Beschreibung eines Würfels.	27
3.15	Einheitswürfel zur Gegenüberstellung der Nomenklaturen.	27
3.16	Relationale Darstellung der Eigenschaften von Basiselementen in einer polyhedralen Struktur unter der Verwendung von Eckpunkten.	28
3.17	Basis- und Verbindungselement in einer Knoten-Kanten Struktur abgeleitet aus einer polyhedralen Struktur.	29
3.18	Basis- und Verbindungselement in einer gemischt-komplexen Struktur.	30
4.1	Beziehung zwischen zwei Zeitpunkten.	34
4.2	Beziehung zwischen einem Zeitpunkt und einer Zeitspanne.	35
4.3	Beziehung zwischen zwei Zeitspannen	36
4.4	Modellierung einer einfachen Haltestelle – Warten auf das Verkehrsmittel an der Haltestelle.	42

4.5	Modellierung einer einfachen Haltestelle – Einsteigen in das Verkehrsmittel an der Haltestelle.	43
4.6	Modellierung einer einfachen Haltestelle – Aussteigen aus dem Verkehrsmittel an der Haltestelle.	44
4.7	Modellierung einer einfachen Haltestelle – Fahrt mit dem Verkehrsmittel an der Haltestelle fortsetzen.	45
4.8	Modellierung einer komplexen Haltestelle – Umsteigen an der Haltestelle von einem Verkehrsmittel der Linie L_1 in ein Verkehrsmittel der Linie L_2	46
4.9	Modellierung einer Haltestelle – Bewegung im Netz des Individualverkehrs an der Haltestelle fortsetzen.	48
4.10	Modellierung einer Haltestelle – Betreten der Haltestelle.	49
4.11	Modellierung einer Haltestelle – Verlassen der Haltestelle.	50
5.1	Statistische Kennwerte des österreichischen Straßennetzes nach Tele Atlas in hierarchisch absteigender Reihung (Stand 2008).	57
5.2	Relationale Darstellung der Eigenschaften von Basiselementen in einer hierarchisch strukturierten Kanten-Traversen Struktur.	60
5.3	Aufbau der Ausgangsdaten.	61
5.4	Hierarchische Suchbaumausbreitung vom Startelement 291 zum Zielelement 300 mit gesonderter Ausweisung einer Übersprungstelle eines Suchbaumes.	63
5.5	Hierarchische Suchbaumausbreitung bei unterschiedlichem räumlichen Kontext.	63
5.6	Gegenüberstellung der permanenten und temporären Kantenmengen der Start- und Zielsuchbäume bei der Routenberechnung zwischen Kante 175 und Kante 186.	67
6.1	Datengrundlage für Nachbarschaftsanalyse.	81
6.2	Grade der Nachbarschaft.	81
6.3	Gegenüberstellung der verschiedenen Nachbarschaftsklassen.	87
6.4	Übereinstimmende Eckpunkte für die Nachbarschaftsanalyse bei polyhedraler Modellierung.	88

Algorithmenverzeichnis

1	Allgemeine Suchbaumausbreitung nach Dijkstra.	11
2	Extraktion der Route.	13
3	Erzeugen des Basiselementes des Regionsadjazenzgraphen aus einer polygonalen Struktur.	23
4	Erzeugen des Verbindungselementes des Regionsadjazenzgraphen aus einer polygonalen Struktur.	23
5	Bidirektionale Suchbaumausbreitung.	55
6	Hierarchische Suchbaumausbreitung.	62
7	Multi Level Hopping.	66
8	Auswahl von gültigen Nachbarelementen beim Übergang von zeitunabhängigen zu zeitabhängigen Basisdaten: Zeitfang.	75
9	Flächenberechnung eines in der YZ-Ebene gelagerten Polygons.	86