TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Kartographie

Data Model and Algorithms for Multimodal Route Planning with Transportation Networks

Lu Liu

Vollständiger Abdruck der von der Fakultät für Bauingenieur-und Vermessungswesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Dr. h.c. Reinhard Rummel

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Liqiu Meng
2. Univ.-Prof. Dr. phil. Alexander Zipf, Ruprecht-Karls-Universität Heidelberg

Die Dissertation wurde am 29.11.2010 bei der Technischen Universität München eingereicht und durch die Fakultät für Bauingenieur-und Vermessungswesen am 16.02.2011 angenommen.

Abstract

Determining a best route in highly developed complex transportation networks is not a trivial task, especially for those who are unfamiliar with the local transportation system. To assist the mobility of people by taking advantage of the multimodal transportation infrastructure is the main goal of intelligent multimodal navigation services. Multimodal route planning that aims to find an optimal route between the source and the target of a trip while utilizing several transportation modes including car driving, public transportation, cycling, walking, etc. is essential to intelligent multimodal navigation services. Although the task originates from the field of transportation, it can be abstracted as a general form independent of the domain-specific details on the underlying data model and algorithms. This research work is therefore dedicated to a general approach of modeling the multimodal network data and performing optimal path queries on it. The approach is approved in the application field of urban transportation.

The bottleneck in the development of a multimodal route planning service is reflected in two aspects: one is the lack of a high-quality dataset; the other is the lack of effective modeling and path-finding approach. With an integrated navigation dataset produced from an automated data-matching process as the desirable test bed, this research is focused on the second aspect.

The weighted digraph structure can well represent the fundamental static networks. For each mode, there is one corresponding mode graph. These graphs constitute the Multimodal Graph Set as a key component of the overall multimodal network data model. In comparison with the traditional mono-modal problem, another key component necessary in the modeling of multimodal route-planning problem is mode-switching actions. In this work, such actions are described by Switch Points which are somewhat analog to plugs and sockets between different mode graphs. Consequently, it is possible to plug-and-play a Multimodal Graph Set by means of Switch Points.

On the basis of the multimodal network data model, the multimodal route-planning problem is categorized into two types and formalized as the multimodal shortest path problem on the Plug-and-Play Multimodal Graph Set. The first type where the mode sequence is given in the input is described as to find a shortest path from a given source to a destination across the modes in the sequence one after another. This type of problem can be solved within a general algorithmic framework. For the second type where the mode sequence cannot be determined beforehand, the multimodal path-finding algorithm can make good use of the traditional mono-modal shortest path algorithms together with the SCM-PLUG operation. It turns out that the solutions for these two types of problem are equivalent if the input mode list for the first type is transformed into its matrix expression. When applying the general multimodal route-planning approach to a specific application domain, a rule-based inferring process is necessary to determine whether a mode sequence is reasonable or not.

Performance evaluations on the integrated navigation dataset have verified the efficiency of the proposed approach. A web-based prototype system demonstrates the whole workflow of the multimodal route-planning function which is missing in any other existing systems. Case studies based on the prototype system show that all feasible routing plans and the corresponding optimal paths can be automatically created for users who just need to tell the system about their preferences on the usage of transportation modes.

Zusammenfassung

Die beste Route in einem hochentwickelten komplexen Verkehrsnetzen zu bestimmen ist keine triviale Aufgabe vor allem für diejenigen, die mit der lokalen Verkehrsinfrastruktur noch nicht vertraut sind. Das Ziel einer intelligenten multimodalen Navigation ist es den Menschen die bestmögliche Mobilität in einer multimodalen Verkehrsinfrastruktur anzubieten. Multimodale Routenplanung ist unentbehrlich beim Aufbau von intelligent multimodalen Navigationsdiensten. Deren Zweck ist es einen Benutzer den optimalen Weg zwischen Start und Ziel einer Reise anzubieten, wenn dabei die Strecke mehrere Verkehrsträger, einschließlich PKW, öffentliche Verkehrsmittel, Fahrrad, etc. oder auch Strecken zu Fuß, beinhaltet. Obwohl dieses Problem ursprünglich aus dem Transportwesen stammt, kann es in allgemeiner Form durch die Entkopplung der Domain-spezifischen Details aus dem zugrunde liegenden Datenmodell und Algorithmen abstrahiert werden. Diese Forschungsarbeit widmet sich dem allgemeinen Ansatz der Modellierung eines multimodalen Datenmodells, den Abfragen von optimalen Routen in diesem Modell, sowie deren Anwendung in einer urbanen Verkehrsinfrastruktur.

Der Engpass bei der Entwicklung eines multimodalen Routenplanungsdienstes lässt sich in zwei Aspekte aufteilen: Zum einen das Fehlen eines hochwertigen Datenbestandes, zum anderen, dass keine effektive Modellierung und kein Wegfindungsansatz existiert. Unter Verwendung eines integrierten Navigationsdatensatzes, erzeugt aus einem automatisierten Daten-Matching-Verfahren, das als Testumgebung zur Verfügung steht, konzentriert sich diese Forschung auf den zweiten Aspekt.

Die gewichtete Digraphstruktur ist zur Repräsentation der fundamentalen statischen Verkehrsnetze geeignet. Für jeden Modus gibt es einen entsprechenden Modusgraph. Diese Graphen konstituieren das Multimodale Graphenset, dass eine Komponente des gesamten multimodalen Datenmodells darstellt. Im Vergleich mit dem traditionellen mono-modalen Problem ist die primäre Herausforderung bei der Modellierung einer multimodalen Routenplanung, dass Modus-Umschaltungs-Aktionen zusätzlich zu den Basisnetzen berücksichtigt werden müssen. Solche Maßnahmen werden mit der Einführung des Konzepts Switch-Point, dass unter bestimmten Bedingungen mit Stecker und Buchsen zwischen verschiedenen Modusgraphen verglichen werden können, modelliert. Folglich kann der Multimodale Graphenset nach „Plug-and-Play-Art“, mit der Unterstützung der anderen Komponente des Datenmodells, nämlich den Switch-Points, verwendet werden.

Auf Basis des multimodalen Datenmodells ist die multimodale Routenplanungsaufgabe in zwei Typen kategorisiert und wird als multimodales kürzester Pfad Problem auf dem Plug-and-Play Multimodalen Graphenset formalisiert. Das Typ-I-Problem, bei dem die Modus-Sequenz dem Input hinzugefügt wird, kann durch die Suche des kürzesten Weges von einem gegeben Ursprung zu einem Zielpunkt, über die verschiedenen Modi der Sequenz laufend, beschrieben werden. Dieser Problemtyp kann durch ein allgemeines algorithmisches System gelöst werden. Für das Typ-II-Problem, bei dem die Modus-Sequenz nicht im Voraus bestimmt werden kann, lassen sich konventionelle mono-modale Algorithmen, mit der

vorherigen Anwendung eines SCM-PLUG-Verfahrens, gut für den multimodalen Wegfindungsalgorithmus verwenden. Es stellt sich heraus, dass die Lösungen beider Probleme äquivalent sind, wenn die Modus-Sequenz des Typ-I-Problems in eine Matrixform transformiert wird. Bei der Anwendung des allgemeinen multimodalen Routenplanungsansatzes in einem speziellen Domain ist ein regelbasiertes Ableitungsverfahren notwendig, um zu bestimmen ob eine Modus-Sequenz möglich ist oder nicht.

Leistungsbewertungen auf Grundlage des integrierten Navigationsdatensatzes haben die Effizienz des vorgeschlagenen Ansatzes bestätigt. Ein Web-basiertes Prototypensystem demonstriert den gesamten Workflow der multimodalen Routenplanungsfunktion, das, soweit bekannt, kein anderes, bestehendes System leistet. Fallstudien zum Prototypensystem zeigen, dass alle sinnvollen Routingpläne sowie die korrespondierenden optimalen Pfaden dem Benutzer, der lediglich im System seine Präferenzen bei der Nutzung der Verkehrsträger festlegen muss, automatisch generiert werden können.

Table of Contents

List of Figures

List of Tables

Abbreviations

APSP	all-pairs shortest-paths problem
ATKIS	Amtliches Topographisch-Kartographisches Informationssystem, the official topographic cartographic information system in Germany
DBMS	Database Management System
DLM	Digital Landscape Model
GIS	Geographic Information Science
GIS-T	the application of geographic information systems in transportation
GPS	Global Positioning System
ITS	Intelligent Transportation System
MMGS	Multimodal Graph Set
MMSP	multimodal shortest path
MMSPA(s)	multimodal shortest path algorithm(s)
MMTQ	Multimodal TWO-Q algorithm
P2P	point-to-point shortest-path problem
P2PMMSP-I	point-to-point MMSP problem with deterministic mode sequence, or Type I point-to-point MMSP problem
P2PMMSP-II	point-to-point MMSP problem with non-deterministic mode sequence, or Type II point-to-point MMSP problem
PND(s)	Personal Navigation Device(s)
PnPMMGS	Plug-and-Play Multimodal Graph Set
PNP-MMTQ	MMTQ working on PnPMMGS
POI(s)	Point(s) of Interest
SCM-PLUG	Switch Condition Matrix-based Plug
SPA(s)	shortest path algorithm(s)
SSMMSP-I	single-source MMSP problem with deterministic mode sequence, or Type I single-source MMSP problem
SSMMSP-II	single-source MMSP problem with non-deterministic mode sequence, or Type II single-source MMSP problem

SSSP	single-source shortest-paths problem
STM	Switch Type Matrix
STM-T	STM in transportation
VGI	Volunteered Geographic Information

Chapter 1

Introduction

1.1 Motivation

In a modern society, transportation networks become more and more complex with the development of infrastructure construction for the purpose of facilitating mobility. However, high-complexity, high-density, multi-layer and multi-modal transportation networks may not automatically bring convenience to users. They may rather confuse users' planning to go from one place to another. Figure 1.1 shows a map with a bunch of rendered transportation networks. The densely plotted lines with different styles and attached labels are intertwined with each other. There is so much information contained in the highly developed transportation networks that many people may find it difficult to figure the best routes out. Consequently, for the sake of helping people efficiently find the best paths through the transportation infrastructures, route planning is gaining more and more importance in the fields of logistics and transportation.

Figure 1.1 Confused user in front of a labyrinth-like complex transportation network.

In recent years, personal navigation service is becoming more and more widespread. It benefits from the rapid development of supporting technologies including high-accuracy positioning, high-speed data communication, powerful mobile computation, openness and standardization of navigation data, efficient data processing and analysis, etc. Although a personal navigation service or application contains multiple navigation-related functions, and has a complicated underlying workflow, it basically answers not more than three questions: *where I am, where the destination is and how to get there*. Route planning, as an important function of the service, is responsible for a part of the third question based on the results of the first two. The other part of the question, which is the turn-by-turn instructions given on the fly, ought to be answered by route guidance function. Such sort of service is available in some commercial products such as hand-held GPS systems, in-vehicle navigation devices, smart phones enhanced with navigation function, etc. Systems with only route planning function are more and more common as well. Quite a few public transit operating organizations offer some kind of route planning facilities in the form of web-based information systems. Other well-known online route planning systems include: Google Maps[1], Bing Maps[2] and MapQuest[3] working on commercial navigation dataset, OpenRouteService[4] and CloudMade Routing[5] working on Volunteered Geographic Information (VGI) dataset OpenStreetMap[6]. These products or systems work pretty well, but only within their respective domain.

In the real life, people may find difficulties when they try to plan routes in a complex transportation network across multiple different transport means with the abovementioned products or systems. For instance, a car navigator does not work in public transit system. And a public transit route planner cannot tell a user who has a car or bicycle available on departure how to get to a feasible station with such private traffic tools. Although some of the route planning systems are making efforts to integrate more transportation modes, e.g. Google Maps added “Walking”, “By public transit” and “Bicycling” options besides “By car” in its Get Directions function for some areas as shown in Figure 1.2, the route planning is performed totally separately for each mode, i.e. one mode at a time.

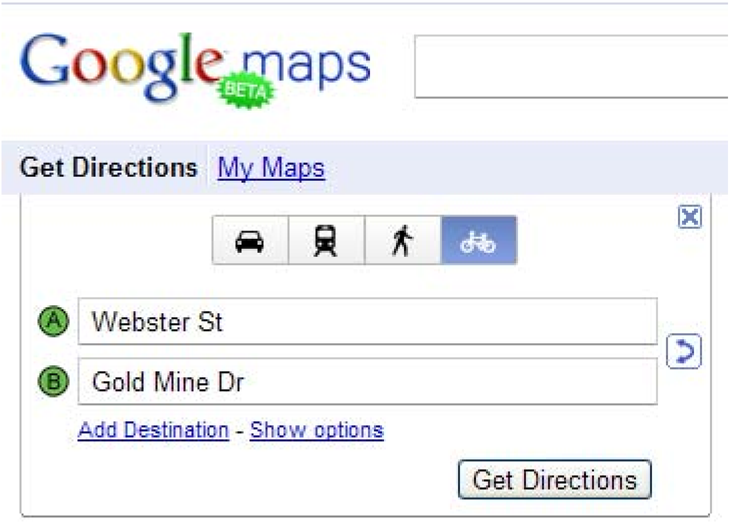


Figure 1.2 Four separation options for Get Directions in Google Maps.

In transportation field, the term *mode* is defined as a transportation means or a tool, e.g. private car, bicycle, wheelchair, bus, underground train, suburban train, tram, inter-city shuttle bus, inter-city train, aircraft, ship, etc. A transportation mode can be private or public. More

1 http://maps.google.com
2 http://www.bing.com/maps/
3 http://www.mapquest.com/
4 http://www.openrouteservice.org/
5 http://maps.cloudmade.com/
6 http://www.openstreetmap.org/

precisely, transportation modes are the means by which people and freight achieve mobility (Rodrigue et al. 2009). Besides the meaning of transportation means, a mode also refers to the type or functional class of physical transportation network, e.g. motorized road, pedestrian way, bus line, inner-city railway, inter-city railway, airline, sea line, etc. Throughout this work, both implications are adopted. The concrete indication is based on the context. For example, it refers to the network type when discussing the modeling issue of multimodal transportation network or to the transportation means when inferring the feasible mode combination of the result.

In fact, better solutions can usually be yielded if two or more modes are taken into consideration when planning routes. Furthermore, some seemingly unsolvable routing problems in mono-modal situation do have solutions if reconsidered from the multimodal point of view. For example in Munich, if a traveler wants to find a path from Albrechtstraße 37 to a spot near Scholss Nymphenburg 205 which is in a pedestrian-only area, there is definitely no direct motor way in between (Figure 1.3). Though a pure pedestrian route is possible, it is apparently time-consuming. However, a double-modal and faster route is easily found by driving to the parking lot near the east gate of Schloßpark Nymphenburg first, parking the car there and walking to the destination as demonstrated in Figure 1.4.

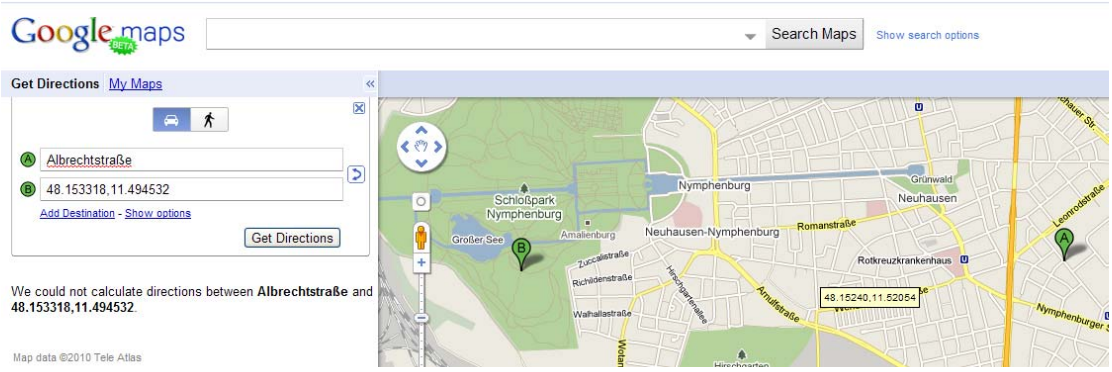


Figure 1.3 Routing result – no direct path given by Google Maps.

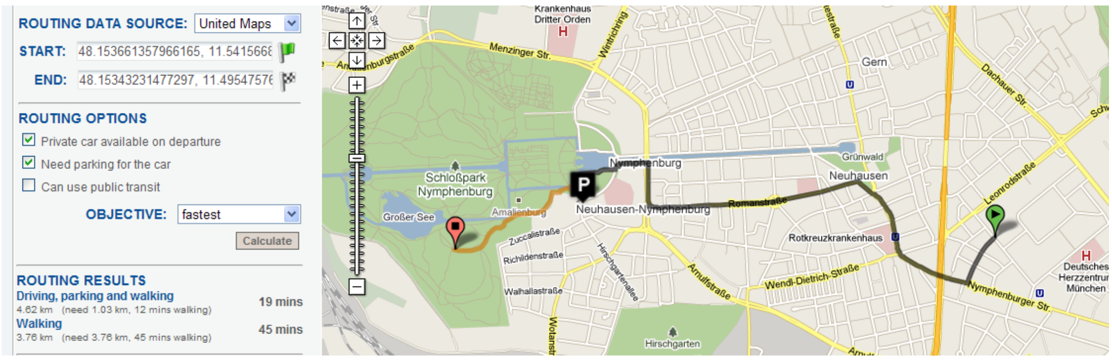


Figure 1.4 A feasible double-modal route between Albrechtstraße 37, Munich and the destination in the pedestrian area.

There are of course more complicated multimodal route planning cases in our everyday life. Unfortunately, automatic multimodal routing solutions are not available in conventional route-planning systems. With the increasing availability and integration of various types of road networks, it can be anticipated that the multimodal route planning will soon become a popular service. This work addresses the topic of multimodal route planning which involves several research fields ranging from cartography, Geographic Information Science (GIS), transportation to computer science. This work keeps its focus on the network data model and path finding algorithms of multimodal route planning in transportation field for people living in urban area.

1.2 Goal

In a nutshell, *multimodal route planning* refers to the problem of route planning involving different transportation modes. In some literatures from the field of transportation research, it is also called intermodal route planning. The authors refer the underlying networks to be multimodal, while a trip across different modes of network to be intermodal. In this work, the term multimodal is adopted whenever it refers to the networks or a trip for consistency. From the data modeling and algorithmic point of view, the main bottleneck of solving route planning problem in the context of multimodal transportation is reflected in two aspects:

- **Multi-source navigation data integration.** Different modes correspond to various navigation datasets which are acquired, stored and managed by different public or private organizations. A multimodal routing application requires these datasets be interoperable with each other. This employs a challenging task of integrating different datasets into one containing all the necessary geometric, topological and semantic information for multimodal route planning. A previous thesis work from our department was devoted to this topic (Zhang 2009).

- **Multimodal network modeling and optimal path-finding approach.** A raw navigation dataset cannot be directly used for route planning even though it is produced by a high-quality data integration process. It needs a procedure of network modeling (or graph construction) that extracts necessary information and builds appropriate data structures for path-finding algorithms to perform on. The network-modeling method and path-finding algorithm design are closely related to each other. In the multimodal context, the network-modeling and path-finding problem cannot be simply solved by adopting approaches that work well in mono-modal situation.

This thesis is dedicated to the second aspect, and intended to find feasible and efficient approaches to the problem of multimodal network modeling and optimal path finding. The approaches can provide the next generation of personal navigation service – multimodal navigation service – with theoretical and technical support. To achieve this goal, a ready-to-use navigation dataset containing sufficient information for multimodal route planning is necessary. The following four research tasks are involved:

- **Representation of multimodal transportation networks.** This is the main task of data modeling where the appropriate data structures of the networks are investigated by taking their multimodal characteristics into account. Additionally, the representation of the spots where people can change from one mode to another, which is called *Switch Point* in this work, plays a significant role as bridges between different modal networks.

- **Formalization of multimodal route-planning problem.** On the basis of an appropriate data model that can well depict both of multimodal transportation networks and mode-switching actions, it is necessary to formalize the problems and clarify what the inputs are and what results are supposed to look like. The purpose of formalization is to abstract and generalize the problems that all cases that occur in the reality are represented.

- **Design and analysis of multimodal route-planning algorithms.** The algorithms are the solutions to the problems defined in the second task. They should work on the multimodal network data model and are able to create all feasible mono-modal or multimodal optimal routes. Their correctness needs to be confirmed and their computing complexities need to be analyzed.

- **Evaluation and demonstration of the proposed approach.** The theoretical foundation should be flanked by empirical work. What do the proposed approaches behave with respect to the performance? Can they be applied in practice and solve real-world multimodal route-planning problems? To answer these questions, extensive experiments including performance evaluations, demo application development and case studies are necessary.

This dissertation covers all these aspects and strives for a general, effective and efficient approach to help people find optimal routes in urban transportation networks with accessible modes and reasonable mode combinations.

1.3 Structure of this thesis

After this chapter including the motivation and goal of the research, the rest of this dissertation is organized as follows.

The technical and methodological fundamentals of this work are introduced in Chapter 2. In addition, the state of the art in the field of optimal path finding, especially multimodal path finding, is summarized and reviewed.

The modeling method of multimodal transportation networks including road and public transit networks for both private and public travel modes is elaborated in Chapter 3. The core concept – Switch Point – for the network modeling as well as the algorithm designing is defined and analyzed in detail. The time dependency and constraints applicable on Switch Point are introduced for the purpose of taking timetables and other possible restrictions into account when planning a multimodal route. Plug-and-Play Multimodal Graph Set (PnPMMGS) and the associated operation for the construction of this data structure called Switch Condition Matrix-based Plug (SCM-PLUG) are proposed on the basis of Switch Point concept.

With the support of the multimodal graph data model, multimodal shortest path algorithms are designed. Chapter 4 introduces the algorithms generalized from both label-setting and label-correcting methods. The problems are categorized into two types depending on whether the mode sequence can be known as a part of input or not. For the situation that mode sequence can be predetermined, a generic algorithmic framework is proposed, and the mode sequence inferring process is introduced; while for the situation that the sequence cannot be given in advance, a PnPMMGS is constructed with SCM-PLUG before the shortest path algorithm can be carried out. The relations between these two solutions are analyzed.

The experiments and implementation details are covered in Chapter 5. After the evaluation of the algorithms in terms of performance on real-world transportation network dataset, the architecture of an online prototype system for the Multimodal Route Planning is described. Two case studies in Munich and Berlin with the help of the prototype system are introduced and evaluated.

The final chapter summarizes the main contributions followed by an outlook about the emerging research issues under this topic.

Chapter 2

Fundamentals and Related Work

2.1 Human mobility in a multimodal context

As mentioned in Chapter 1, the original problem that motivates the author to do this work is from transportation field. In general, the goal of transportation is to transform the geographical attributes of freight, people or information, from an origin to a destination, conferring them an added value in the process (Rodrigue et al. 2009).

Many applications in transportation involve optimization, or the design of solutions to meet specified objectives. These objectives might include the minimization of travel distance, time or any cost accumulating along the path. One of the most important areas of optimization is related to route planning, or decisions about the optimum tracks followed by people who have mobility requirement. This is an interesting issue about human mobility within urban areas. The term *mobility*, from the perspective of transportation geography, refers to the ability to move between different activity sites (Hanson 1995). It can have different levels linked to the speed, capacity and efficiency of movements (Rodrigue et al. 2009). As one of the basic human needs, mobility is central for the modern social life. Many human mobility activities such as commuting, shopping, business traveling, attending some event etc. are driven by a strong purpose in terms of efficiency, economy or convenience.

To assist people's mobility in transportation networks, Intelligent Transportation Systems (ITS) are developed with navigation services as an indispensible part. People need the assistance of navigation to plan and modify routes in response to new information. At present, personal navigation service is mostly offered by Personal Navigation Device (PND) which is a portable electronic product combining a positioning capability and navigation functions. According to the marketing analysis of the firm *Berg Insight*, there were more than 150 million turn-by-turn navigation systems worldwide in mid 2009, including about 35 million embedded car navigation systems, over 90 million PNDs and an estimated 28 million navigation-enabled smart phones with GPS (Malm 2009).

Traditionally, navigation is the process of reading and controlling the movement of a craft or vehicle from one place to another. This definition clearly indicates that navigation is vehicle-oriented. However, as the related technology has rapidly evolved since 1980s, more and more devices aiming at personal navigation requirement have appeared on the market. Figure 2.1 shows examples of PND and navigation-enabled mobile handset products, some of which were released very recently.

(*a*) (*b*) (*c*) (*d*)

Figure 2.1 Typical PNDs in the market. (*a*). TomTom Urban Rider for bikers[7]; (*b*). Google Maps Navigation on Nexus One[8]; (*c*). TeleNav application on iPhone[9]; (*d*). Nokia Ovi Maps[10].

Different types of navigation solutions and devices are likely to co-exist in the future. Vehicle-oriented navigation systems often provide a better user experience than many smart phone-based navigation services because the underlying technologies are more mature. Moreover, navigation services on smart phones are also well suited as a complement to other solutions, especially for use outside the car. Pedestrian navigation features gradually being introduced include improved map data and multimodal navigation. They assist users in planning routes while taking into account all available transportation modes including private cars, bicycles, walking, public transits, etc. Almost all of the existing navigation services such as car navigation, pedestrian navigation, public transportation information systems which can do route planning for a specific network are mono-modal. Some public transportation information systems can be envisaged as an embryonic form of multimodal route planning systems as they can serve the user with an estimated travel time and an overview map that sketches the walking routes from the original locations to various stations.

The emerging issue of multimodal navigation service that is not confined to car driving has drawn an increasing attention from both academic and industrial fields since the beginning of 21st century. People need such kind of service wherever they are during a trip, especially in an unfamiliar area. Since the human mobility within an urban area actually always happens in a multimodal transportation network, the personal navigation service should also be able to work inter-modally. As indicated by Goodchild, one of the complications that navigation-supported systems have to deal with is the requirement of multimodal usage. At the time when Goodchild published the reviewing paper on the application of geographic information systems in transportation (GIS-T), "few efforts have been made to create databases that combine modes, by representing both road and rail networks and their interconnections, for example, but these would be essential for multimodal routing." (Goodchild 2000)

Among all the supporting technologies of transportation and navigation, GIS always plays

[7] http://www.tomtom.com/en_gb/products/bike-navigation/urban-rider-europe/

[8] http://www.google.com/phone/detail/nexus-one

[9] http://www.telenav.com/gps-navigator/apple/iphone/

[10] http://maps.nokia.com/services-and-apps/ovi-maps/ovi-maps-main

an important role. Route planning, as one of the classic problems in network analysis, is widely investigated in GIS, computer science and operational research. The problem of route planning can be modeled by finding a shortest path on a weighted directed graph. Lots of approaches developed so far address different objectives both in theory and practice. The classic solutions to conventional routing problems have been included in most of commercial GIS software. Although these theories, methodologies and tools haven't yet taken into account the integration of multiple transportation modes, they provide fundamental support for personal navigation service and possibilities of enhancing human mobility in the context of urban multimodal transportation.

2.2 Graph theory and shortest path problem

Behind all routing or route planning problems is the concept of the *shortest path* – the path(s) through the network from a known starting point to an optional ending point that minimizes distance, or some other measure based on distance, such as travel time. This is one of the most classic problems in *graph theory*, and has been investigated extensively since half a century. This section contains a brief introduction to graph theory and the shortest path problem because they form the methodological foundation of the thesis work.

A graph $G = (V, E)$ is an ordered pair of sets. Elements of *V* are called *vertices*, and elements of $E \subseteq V \times V$ are called *edges*. For consistency, the term vertices and edges are adopted respectively throughout the dissertation. *V* refers to the vertex set of *G*, while *E* the edge set. In the case that any direction of the edges is disregarded, *G* is referred to as an *undirected graph*.

One can label a graph by attaching labels to its vertices. If $(u, v) \in E$ is an edge of a graph $G = (V, E)$, *u* and *v* are said to be *adjacent* vertices. The edge (u, v) is also said to be *incident* with the vertices *u* and *v*. A *directed edge* is an edge such that one vertex incident with it is designated as the head vertex and the other as the tail vertex. A directed edge (u, v) is said to be directed from its tail *u* to its head *v*. A *directed graph* or *digraph G* is a graph with directed edges. The *indegree*/*outdegree* of a vertex $v \in V(G)$ counts the number of edges such that *v* is the head/tail of those edges. Similarly, the *incoming*/*outgoing* edges of a vertex $v \in V(G)$ are a set of edges such that *v* is the head/tail of those edges. Furthermore, the set of *in-neighbors*/*out-neighbors* of $v \in V(G)$ consists of all those vertices that contribute to the indegree/outdegree of *v*. A *multigraph* is a graph in which there are multiple edges between a pair of vertices. A *multi-undirected* graph is an undirected multigraph. Similarly, a *multidigraph* is a directed multigraph.

A graph is said to be *weighted* when a numerical label or weight is assigned to each of its edges. There might be a cost involved in traveling from a vertex to one of its neighbors, in which case the weight assigned to the corresponding edge can represent such a cost. Sometimes, the weight is expressed by a cost function $c : E \to R$ mapping edges to real-valued weights. Hence, the term *cost* is adopted instead of weight in this work.

Based on the concept of weighted graphs, a *shortest-path problem* can be defined. Being given a weighted digraph $G = (V, E)$ with edge cost function $c : E \to R$, the cost of path *p* from *v* to *w* is the sum of the edge costs for edges in *p*, denoted by $c(p)$. The minimal value of $c(p)$ for all paths from *v* to *w* is the *shortest-path cost* denoted by $\delta(v, w)$. A *shortest path* from vertex *v* to vertex *w* is then defined as any path *p* with cost $c(p) = \delta(v, w)$.

A few variations related to the shortest-path problem are:

- **Single-source shortest-paths problem (SSSP):** find a shortest path from a given source vertex $s \in V(G)$ to each of other vertices $v \in V(G)$.
- **Point-to-Point shortest-path problem (P2P):** find a shortest path from *v* to *w* for a pair of given vertices *v* and *w*. If an SSSP with source vertex *v* is solved, the P2P is solved as well. The algorithm for P2P can be more efficient in practice by setting new terminating conditions to SSSP algorithm.
- **All-pairs shortest-paths problem (APSP):** find a shortest path from *v* to *w* for every pair of vertices *v* and *w*.

More detailed fundamental knowledge of graph theory and the description of the classic shortest path problem is referred to the textbooks *Graph Theory* (Diestel 2006), *Introduction to Algorithms* (Cormen et al. 2009) or the up-to-date electronic book under GNU Free Documentation License[11] *Algorithmic Graph Theory* (Joyner et al. 2010).

With regards to the solution to the shortest path problem, or the *shortest-path algorithm* (SPA), a large number of related works are available. Researchers typically classify algorithmic approaches into two groups: *label setting* and *label correcting* (Ahuja et al. 1993, Klunder and Post 2006). Both approaches are iterative. They assign tentative distance labels to vertices at each step; the distance labels are estimates of (i.e., upper bounds on) the shortest-path distances. The approaches vary in how they update the distance labels from step to step and how they “converge” toward the shortest-path distances.

Label-setting methods determine an exact (permanent) distance label of one node per iteration. Unfortunately, they can handle only a restricted set of instances, for example, acyclic graphs with arbitrary edge costs or arbitrary graphs with nonnegative edge costs. The label-correcting methods are more flexible and, in general, they do not have such restrictions regarding their input. A label-correcting algorithm may change all distance labels multiple times and only after the final step they all become permanent. However, the label-correcting algorithms are in general less efficient with respect to worst-case running time compared to label-setting ones. A good overview and discussion on both groups of labeling algorithms can be found in (Ahuja et al. 1993).

It has been 50 years since the most classic shortest-path-finding algorithms were proposed by Dijkstra (1959), Bellman (1958) and Ford (1956, 1962). Their works lay the foundation for the research of general SPA in theory. Therefore, the two SPAs based on labeling method, Dijkstra’s algorithm and Bellman-Ford algorithm, are named after these three pioneers in theoretical computing. From then on, the research work around this topic grows considerably

[11] http://www.gnu.org/copyleft/fdl.html

with the fast development of transportation technology, electronic engineering and the Internet within last decades. About 820,000 or 188,000 relevant items[12] pop up if one performs a Google Web or Google Scholar search with the keywords "*shortest path*" given respectively. More than 1500 registered patents related to shortest paths bear witness to the popularity of shortest-path problems according to the statistics by Santos (2009) who also reported the active application areas in which SPA plays important role.

Most of the research work concentrates on the improvement of the running time of the classic SPA. On one hand, efforts have been made to reduce the theoretical worst-case running time by introducing better data structures. On the other hand, researchers introduced speed-up techniques for specific inputs, especially for transportation networks. Both of these improvements are applicable to the label-setting and label-correcting methods.

More specifically, efforts devoted to data structures of the candidate vertex queue include:

- **Label-setting method.** In the original form of Dijkstra's algorithm, a simple list is utilized to express the vertex min-priority queue which reveals an overall running time of $O(|V|^2)$. This bound has been improved several times by adopting more elaborate priority queues, from a binary heap leading to a running time of $O(|E|\log|V|)$ to a Fibonacci heap implementation of $O(|E|+|V|\log|V|)$. If the edge weights are restricted to integers, the theoretical complexity can be improved further. Thorup (2004) implemented the method with the running time improved to $O(|E|+|V|\log\log C)$ where *C* is the upper bound of all edge weights.
- **Label-correcting method.** The earliest label-correcting algorithm, Bellman-Ford algorithm uses the most trivial first-in, first-out (FIFO) queue. Pape (1974) improved the queue upon vertex entrance by checking if it has ever been in the queue before – if yes, insert it at the top; otherwise at the bottom. Pallottino (1984) went one step further by organizing the candidate list into two distinct queues, thus called Two-Q algorithm. Besides, the methods (Glover et al. 1986, Bertsekas 1993, Bertsekas et al. 1996, Cherkassky et al. 1996) were motivated by a strategy of placing small labels near the top of the queue, so as to emulate approximately the minimum label selection policy of label-setting method at a much smaller computational cost (Klunder and Post 2006).

The speed-up techniques can be categorized into three different types as well as their combination:

- **Goal-directed search.** The classic SSSP algorithms generally do not consider the location of the target. Vertices are scanned in the order of their proximity to the source, independent of the direction of the target. If only a shortest path between two locations has to be determined, SSSP algorithms can be speeded up by taking into account a heuristic estimation of the cost from a location to the target which is smaller than or equal to the actual cost. It is possible to reduce the searching scope if the algorithm

[12] Data acquired on August 18, 2010, 15:13

searches more in the direction of the target. The algorithm that applies such heuristics is called A* algorithm named by Hart et al. (1968), and went through an in-depth investigation by Dechter and Pearl (1985). Since then on, A* algorithm has been improved by many other researchers for the application in transportation realm (Hasselberg 2000, Jagadeesh et al. 2002, Goldberg and Harrelson 2005, Hahne et al. 2008, Zeng and Church 2009).

- **Bidirectional search.** The idea of bidirectional search is to accelerate P2P path finding by starting a second simultaneous search from the target (Lenie and Dennis De 1977, Dennis De 1983, Luby and Ragde 1989, Hermann and Gerhard 1997). The backward search operates on the same graph with all the edges reversed. Because the searched scope now consists not of one big “sphere” whose “radius” equals to the distance from the source to the target, but of two smaller “spheres” each with a radius that amounts half of the distance from the source to the target, the number of scanned vertices will become roughly a half as before.

- **Network partitioning.** The underlying networks for route calculation in nationwide continent-wide transportation datasets usually contain millions of vertices and edges. Planning an optimum route on such a real-world network takes too long time if a standard SPA is used. Therefore, approaches are developed to reduce the size of the network for planning routes by *network partitioning* including horizontal partitioning that divides the network into several disjoint sub-graphs and vertical partitioning namely hierarchical organization. Such sort of approaches has been widely used during a preprocessing step in commercial car navigation systems where the hardware resource is very limited. This approach has been studied by several authors (Huang et al. 1996, Jing et al. 1998, Berry and Goldberg 1999, Jagadeesh et al. 2002, Holzer 2003, Flinsenberg 2004, Möhring et al. 2006, Geisberger et al. 2008).

- **Combinations.** The speed-up techniques could complememt each other to gain additional computational efficiency. The combination of goal-directed and hierarchical approaches was studied by (Vitter et al. 1999, Jagadeesh et al. 2002, Bauer et al. 2010), while bidirectional A* by (Pijls and Post 2006, Nannicini et al. 2008). In (Holzer et al. 2005, Holzer et al. 2009), all previous techniques were systematically combined with each other. Delling et al. (2010) concluded that different combinations yield different improved performances, depending on the graph type.

More detailed reviews on the speed-up techniques of SPA can be found in (Engineer 2001, Barrett et al. 2002, Fu et al. 2006, Klunder and Post 2006, Wagner and Willhalm 2007, Huang et al. 2007, Sanders and Schultes 2007, Delling et al. 2009b).

In addition to theoretical study of the improvements on SPA, the practical performances of the proposed SPA have also been reviewed. Cherkassky et al. (1996) conducted the first comprehensive evaluation work on the performances of 17 algorithms on simulated network datasets. Their study showed that there is no single best code for all classes of shortest path problems, but the double bucket implementation of Dijkstra’s algorithm (abbr. DIKBD) for problems with nonnegative edge weights and their simplified implementation of the Goldberg-Radzik algorithm (Goldberg and Radzik 1993) for problems with negative edge weights were suggested respectively. Compared with the work by Cherkassky et al., the evaluations done on real transportation networks (Zhan and Noon 1998, Jacob et al. 1999, Klunder and Post 2006, Barrett et al. 2007, Sanders and Schultes 2007, Delling et al. 2009b) are more meaningful for route planning algorithms in transportation field in spite of their more or less diverse conclusions. The later the evaluation being conducted, the better results would be acquired because of two main reasons: 1). newly improved algorithms were included in the test and had better practical performances; 2). the hardware environments of the tests keep evolving all the time and provide more and more powerful computing capabilities. To summarize: the improved algorithms from both groups of label-setting and label-correcting methods have been verified to be efficient and recommendable.

2.3 State of the art

One of the largest application areas of SPA is route finding in transportation networks with minimum cost, and has been examined for years (Pallottino and Scutella 1997, Klunder and Post 2006, Holzer et al. 2005, Bast et al. 2007). The research results have been applied in trip planning for passengers or tourists (Nguyen et al. 2001), vehicle navigation systems (Guzolek and Koch 1989, Flinsenberg 2004, Garaix et al. 2010), package delivery or freight shipment systems (Crainic and Rousseau 1986, Boardman et al. 1997, Kim et al. 1999, Southworth and Peterson 2000, Song and Chen 2007, Cho et al. 2010), etc.

Before applying the routing algorithms, transportation networks have to be modeled as an edge/vertex structure, i.e. a graph. This is not a trivial process as the real transportation networks are often highly complicated. Network data model is a special type of the topological feature model. This general model comprises a collection of nodes and lines, as well as the topological relationships between them. Published literatures for specific and concrete modeling cases serve as the foundations for the path-finding algorithmic studies. The discussions on modeling and algorithm development are usually interlaced with each other.

For the road network, Fohl et al. (1996) adopted a fully non-planar representation in order to portray human perceptions of real world transportation networks more realistically. Goodchild (1998) proposed a number of extensions of the basic network data models in addition to the non-planner property. David and Michael (2008) studied read-world road networks from an algorithmic perspective, and provided strong empirical evidence that road networks are quite non-planar. With regards to the efficient transportation network data organization, various methods of network partitioning are applied both vertically and horizontally. Vertical partitioning which is also known as hierarchical network organization method is studied in (Car and Frank 1994, Sanders and Schultes 2005, Geisberger et al. 2008), while the horizontal partitioning including graph decomposition and clustering is investigated in (Huang et al. 1996, Flinsenberg 2004). Another interesting topic with respect to road network modeling is the expression of turn impedance, i.e. turning restrictions for cars and traffic rules in transportation networks (Winter 2002, Volker 2008, Gutiérrez and Medaglia 2008). On the basis of modeling studies, there are quite a few literatures on path-finding approach in road networks. The novel progresses on this topic have been reviewed in (Pallottino and Scutella 1997, Barrett et al. 2002, Flinsenberg 2004, Fu et al. 2006, Klunder and Post 2006).

In the public transit networks, the aforementioned modeling methods for road network need some adaptation to the fixed-route and scheduled system. Schulz et al. (2002) empirically investigated a hierarchical decomposition approach based on multi-level graphs for the railway transport application scenario, and their results exhibit a considerable improvement in shortest-path computation performance. Liu et al. (2001, 2002) proposed a method for best-path planning in public transit network based on a variation of transition matrices. Rüetschi and Timpf (2005) modeled way-finding in the public transportation infrastructure by proposing scene space other than traditional network space, and analyzed their properties in detail. More recently, Bauer et al. (2011) checked whether results from road networks are transferable to timetable information systems. They presented an extensive experimental study of the most prominent speed-up techniques on inputs deriving from different applications. Besides, modeling methods of public transit network with the help of GIS in specific cities can be found in (Filipov and Davidkov 2006, Zeng et al. 2010). An overview of the modeling of urban structure and dynamics that takes into account urban public transportation system was given by Jiang and Yao (2010).

For even more complicated cases, dynamic and real-time information has to be integrated in the process of transportation network modeling and optimal route planning. To tackle this issue, Huang and Peng (2008) presented an object-oriented data model that coherently represents space, time and dynamics of transit networks and supports schedule-based route planning. Both time-expanded and time-dependent approaches were considered in (Pyrga et al. 2008). In the time-expanded model, a group of vertices will be created for one public transit station at each departure or arrival time during the digraph construction. The vertices are ordered according to their time values, and sequentially connected. The edges connecting them are stay edges expressing the waiting times. On the other hand, the time-dependent modeling approach creates only one vertex per station, and there is an edge between every two physically connected stations. The cost of an edge depends on the time at which this particular edge will be used by a path-finding algorithm. Pyrga et al. provided extensive experimental comparison of the two approaches, and drew a conclusion that the time-expanded approach is more robust while the time-dependent one shows a better performance. The approach proposed in (Fawcett and Robinson 2000) could help drivers find routes with minimal delays resulting from congestion on UK roads. Sung et al. (2000) suggested a model for time-dependent networks where the flow speed of each link depends on the time interval, and a modified Dijkstra's algorithm as their solution. Ahuja et al. (2003) studied minimum-time and minimum-cost walking problems of SSSP in a network where edge travel times change dynamically. Huang et al. (2007) proposed an incremental search approach with heuristics based on a variation of the A* algorithm for determining the shortest path between a moving object and its destination. By remodeling unimportant stations in time-expanded model for timetable information, Delling et al. (2008) were able to obtain faster path-query times with less space consumption. Nannicini et al. (2008) applied speed-up strategies by combining bidirectional and A* with landmarks into time-dependent graphs, and achieved effective performance in practice. Moreover, the parallel computing solutions for this problem was discussed in (Ghiani et al. 2003).

With regard to the multimodal context, Couckuyt et al. (2006) from Microsoft patented some basic concepts and functions of a multimodal navigation system, e.g. the multimodal navigation system which includes multimodal route data, a route presentation module, a cost determination module, a routing module and an optional external interface. Rehrl et al. (2007) described the requirements of a multimodal transportation routing system in more details. A multimodal route planning system is essentially rooted on the multimodal network modeling methods and optimal path-finding algorithms.

Boile (2000) presented a framework for integrating intermodal network equilibrium models with GIS. Hoel et al. (2005) from ESRI proposed their approach for efficient modeling of the multimodal network, which was implemented as a tool in ArcGIS Network Analysis toolbox. The modeling issue of a multimodal freight transportation network was discussed in (Southworth and Peterson 2000). Their proposed data models can be regarded as multi-layer graphs connected by transfer nodes or arcs, while (Modesti and Sciomachen 1998, Boussedjra et al. 2004) set up a single graph containing all the information of different transportation modes. The network designing issue for multimodal package delivery or freight shipment was elaborated by researchers from transportation science (Kim and Barnhart 1997, Kim et al. 1999, Macharis and Bontekoning 2004). Van Nes (2002) presented a strategy to design physical multimodal transport network based on the concept of hierarchical network level. Wang et al. (2009) improved the hierarchical multimodal network model, and described the relationship between different levels. Pereira and Vidal (2001) proposed a multimodal model from a pedestrian's behavioral perspective for public transportation in Barcelona. At the conceptual level, the object-oriented paradigm was utilized to describe the objects in multimodal networks with behaviors and semantics (Chiu et al. 2005, Bielli et al. 2006, Yang 2008). And in the field of geographical data standardization, the support for multimodal transport including pedestrian navigation is one of the foci in the latest version of Geographical Data Files (GDF) (Van Essen and Hiestermann 2005). Mouncif et al. (2006) modeled the multimodal transportation network by a multimodal graph, where each mode network is represented by one mono-modal subgraph. Friedrich (1998) presented a transport model consisting of a demand model, a network model and a set of impact models from the transportation simulation point of view, which forms the theoretical basis of an online system BayernInfo (Neuherz et al. 2000, Keller et al. 2001). Joel et al. (2009) introduced how the trips and the network can be represented as both a graph and a relational model. They defined a set of operators to work over the transportation concepts and integrated them within an SQL-like syntax to express queries over the uncertain transportation network.

The fast development of transportation infrastructure and multimodal facilities has brought about the emerging requirements of multimodal transporting passengers and freights. The problem of finding the optimal path in a multimodal network has therefore attracted worldwide attention of researchers, especially since the beginning of this century. The proposed approaches have substantially benefited from the existing research results of SPA in computer science. Ziliaskopoulos and Wardell (2000) abstracted the problem with non-hierarchical multimodal graph and proposed several pre-calculated data structures and tables in order to optimize the computation. Lozano and Storchi (2001, 2002) computed the multimodal shortest paths by simultaneously taking into account the viability of mode combinations. Boussedjra et al. (2004) adopted a label-correcting approach that updates some labels associated with the graph nodes to find the intermodal shortest paths in a multi-label graph. Bielli et al. (2006) considered the problem of multimodal transport between towns. They calculated the multimodal shortest path based on pre-computed shortest paths inside low-level subgraphs. Zografos and Androutsopoulos (2008) formulated the problem as finding shortest paths with time windows on a multimodal time-schedule network that lexicographically optimizes the en-route time, the number of interchanges, and the total walking and waiting time. The time constraints were also considered in (Marathe 2002, Chang et al. 2007, Lope 2009). The method in (Cho et al. 2007) was based on dynamic programming and has been applied in real international logistics of container cargo shipment. Frank (2008) proposed a novel approach applying the traditional SPA in the product of the navigation and business graphs. (Bousquet et al. 2009, Bousquet 2009) investigated the two-way viable multimodal shortest path problem based on label-setting method. Gräbener et al. (2009) presented a representation model for multimodal shortest path problem intended for mobile platform. In addition, Barrett et al. proposed a multimodal path-finding method on the basis of a (regular-) language-constrained SPA (Barrett et al. 2008, Pajor 2009) which in turn was generalized from (formal-) language-constrained algorithm (Barrett et al. 2000) for applications in transportation field. The regular language is a formal language that can be described by a regular expression. The time-dependent case was discussed in (Hanif et al. 2003). Recently, this approach was further improved in (Delling et al. 2009a) by taking advantage of Transit-Node Routing (Bast et al. 2007). As remarked in (Delling et al. 2009b), however, using a fast routing algorithm in such a label-constrained scenario is rather complicated. Ayed et al. (2008, 2009) introduced their multimodal route planning approach based on a transfer graph model, especially in a distributed environment – European Carlink platform – where the approach is integrated (Galvez-Fernandez et al. 2009). In their most recent paper (2010), Ayed et al. proposed a hybrid approach for optimizing the time-dependent multimodal transport problems, and revealed a good balance between computation time and memory space by empirical studies. The *k*-shortest paths problem in multimodal transportation networks was discussed in (Li and Kurt 2000, Lin et al. 2009, Kheirikharzar 2010). The attempts to solve the problem based on genetic algorithm were discussed in (Abbaspour and Samadzadegan 2009, Yu and Lu 2010). The discussion on the multi-objective version of the problem, which is much more complicated, can be found in (Modesti and Sciomachen 1998, Li and Kurt 2000, Ziliaskopoulos et al. 2007). Hong et al. (2005) employed a so-called state augmented multimodal network and pre-calculating all-pairs of shortest paths. Chiu et al. (2005) presented a multi-agent approach to support multiple modes of public transportation services as well as mobile vehicles and commuters. Yang (2008) utilized the combinations of speed-up techniques for Dijkstra’s algorithm to calculate the optimal paths in multimodal networks. Su and Chang (2010) developed a heuristic algorithm by considering the transfer and other characteristics between one or multiple travel modes.

Further methodologies for multimodal route planning can also be found in the literatures under the topic of pedestrian navigation and bicycle navigation (May et al. 2003, Baumann et al. 2004, Hochmair 2004, Li and Tsukaguchi 2005, Letchner et al. 2006, Tscheligi and Sefelin 2006), or the integration of indoor/outdoor navigation (Rehrl et al. 2005, Becker et al. 2009, Mandloi and Thill 2010). The issue of multimodal choice decision modeling has also been investigated (Kenyon and Lyons 2003, Li and Tsukaguchi 2005, Molin and Van Gelder 2008).

A growing number of online mapping services (e.g. Google Maps, Bing Maps) provide the "Get Directions" function for driving, walking or taking public transit separately with different geographical coverage, but no combinations of these modes. In other words, these well-known mapping services are unable to provide multimodal results. These direction-getting functions are mainly performed on commercial navigation database bought from professional data providers (see Figure 2.2). As these commercial databases are created by automobiles equipped with positioning devices, there is a serious lack of the pedestrian paths and public transit lines data.

(*a*) (*b*)

(*c*) (*d*)

Figure 2.2 Navigation data providers for some online mapping services. (*a*). Tele Atlas for Google Maps; (*b*). NAVTEQ for Yahoo Maps; (*c*). NAVTEQ for MapQuest; (*d*). NAVTEQ and AND for Microsoft Bing Maps.

Since the public transit data including the lines, stations and timetables information are maintained by local governments or operators, the service of route planning within public transportation system is often developed as a standalone application. In Munich, for example, the Münchner Verkehrs- und Tarifverbund (MVV) provides all kinds of information service with respect to the public transportation system in its official website[13], while BayernInfo[14] is a web portal publishing real-time travel and traffic information as well as multimodal route planning within the state of Bavaria. Such information systems can be found in many other European countries like Britain (Transport Direct[15], Journey On[16] and Transport for London[17]), Finland (Helsinki Region Transport[18]), France (RATP[19]), Italy (ATM[20], Actv[21]), Netherland (9292 door-to-door journey planner[22]), Belgium (De Lijn[23]), etc. and in Japan (Tokyo Transfer Guide[24]) since the public transit systems there are highly developed. In North America, there are some similar systems (RTAchicago[25] and LA Metro[26] in the US, TransLink in Canada[27]), but the service is less popular than in Europe and Japan because its motorized way system is more developed. Many of the system providers claim that they provide multimodal routing results, in essence, however, they mean a set of different mono-modal paths.

Apart from the routing systems focused on road or public transit network, many prototype systems that demonstrate the idea of multimodal route planning have been developed, e.g. RADS in Singapore (Meng et al. 1999), TRANSIMS (Barrett et al. 2002), SMART TRAVELER in Los Angeles (Cashin et al. 2002), LITRES-2 (Horn 2002), EasyTransport (Maria 2005), eFinder (Hong et al. 2005) and MIRAS (Chiu et al. 2005) in Hong Kong, Vienna-SPIRIT and Open-SPIRIT in Austria (Bruntsch and Rehrl 2005, Rehrl et al. 2007), MTIS (Tan et al. 2007), ENOSIS in Greece (Zografos et al. 2009) and MTPS in Taiwan (Su and Chang 2010).

Industry has become increasingly aware of the navigation applications that are targeted beyond vehicle drivers. The data providers, application developers and device manufacturers have noticed the rapidly growing market of navigation for pedestrian, bicycler, public transit commuter and especially their combination – multimodal transportation user. As one of the

[13] http://www.mvv-muenchen.de/
[14] http://www.bayerninfo.de/
[15] http://www.transportdirect.info/
[16] http://www.journeyon.co.uk/
[17] http://www.tfl.gov.uk/
[18] http://www.hsl.fi/
[19] http://www8.ratp.info/
[20] http://www.atm-mi.it/
[21] http://www.actv.it/
[22] http://journeyplanner.9292.nl/
[23] http://www.delijn.be/
[24] http://www.tokyo-subway.net/english/
[25] http://rtachicago.com/
[26] http://www.metro.net/
[27] http://tripplanning.translink.ca/

worldwide dominating navigational data providers, NAVTEQ launched Discover Cities^{TM} project in 2008 with the intention to enable multimodal routing by combining pedestrian thoroughfares with time-sensitive public transit system information. In 2009, Nokia released free navigation – Ovi Maps – for both drivers and pedestrians on some of their own smart phones. Both Garmin and TomTom added the pedestrian mode in their navigation devices in the last two years. NAVITIME from Japan can provide multimodal routing solutions on both web-based and mobile platforms (Arikawa et al. 2007). Graphserver and OpenTripPlanner (2009) are active open-source projects under this topic. Nevertheless, the common drawback of these practical solutions is the lack of detailed modeling of intermodal facilities in transportation networks, which has restricted the ability of providing multimodal route planning service.

Chapter 3

Multimodal Modeling of Urban Transportation Network

This chapter is dedicated to the modeling method of multimodal transportation networks within urban area. The road and public transit railway networks are further classified according to their functional types. Different functional types of network correspond well to the diverse transportation modes, and are separately modeled as a set of standalone graphs. A proper description of the action of transferring from one mode to another is treated as one of the key issues in multimodal route planning problem. The approach in this work is to model such kind of actions as Switch Point with a series of attached properties. In order to keep the basic network model as simple as possible, the time-dependent or dynamic elements are separated from the static network topology. The dynamic parts of the transportation network, e.g. timetables, are modeled within Switch Point. The standalone underlying networks become conditionally pluggable to each other by means of Switch Point, which leads to the concept of Plug-and-Play Multimodal Graph Set.

3.1 Navigation data for route planning

A high-quality navigation dataset is indispensible for both mono-modal and multimodal route planning. Here the "high-quality" refers to the following aspects.

- **High accuracy**. The dataset should be accurate both geographically and topologically. Besides, the data has to be up-to-date. In other words, it must be spatiotemporally consistent with its representing real world as much as possible to ensure the routing results are correct and reliable.
- **High coverage**. The geographical coverage of the dataset is supposed to extend to every routable corner in reality of its covering area, so that it is possible to support the route planning between any origin and destination. To support door-to-door navigation, the road dataset should be as complete as possible.
- **Rich attribute**. The dataset should contain all the navigable attributes that allow users to access features such as expressway ramps, one-way streets, legal turn restrictions, as well as physical and painted lane dividers. The level of attribute richness will bring significant impact on the quality of route planning.

The above quality criteria differentiate the navigation data from normal digital line graphs or other kinds of vector dataset. In addition, the dataset must contain different types of content to support the whole navigation workflow. The content can be categorized into four main components according to their different purposes:

- **For geocoding/reverse geocoding**. Geocoding is the process of finding associated geographic coordinates from other geographic data, such as street addresses, zip codes, or names of Point of Interest (POI). And the reverse geocoding is the opposite. A navigation system should maintain such a dictionary-like mapping database to determine where the user exactly wants to go, or what a specific point on the map is. Although there are more and more geocoding (and reverse geocoding) systems available[28] on the Internet, this database has to be integrated in a navigation system if it cannot be accessed online.

- **For route planning**. As it has been introduced in fundamentals, a graph-based data structure consisting of vertices and edges is essential for route planning algorithms to determine the optimal paths. This basic data structure as well as all the routing-relevant attributes constitute the component for route planning, also known as the navigation view (Goodchild 2000).

- **For visualization**. The data for cartographic presentation ranging from the styles of spatial features and symbols, the fonts of labels, generalized maps for different scales to the auxiliary semantic information are necessary for route rendering, or visualization. Usually a static background map is overlaid with some dynamic information such as planned routes, real-time traffic status, etc.

- **For route guidance**. A navigation system should be able to guide the user from his/her origin to the destination step by step according to his/her current position by providing some text-based, graphic-based or voice-based instructions. The data used for this purpose is termed as route guidance.

The relation among the above four components is illustrated in Figure 3.1.

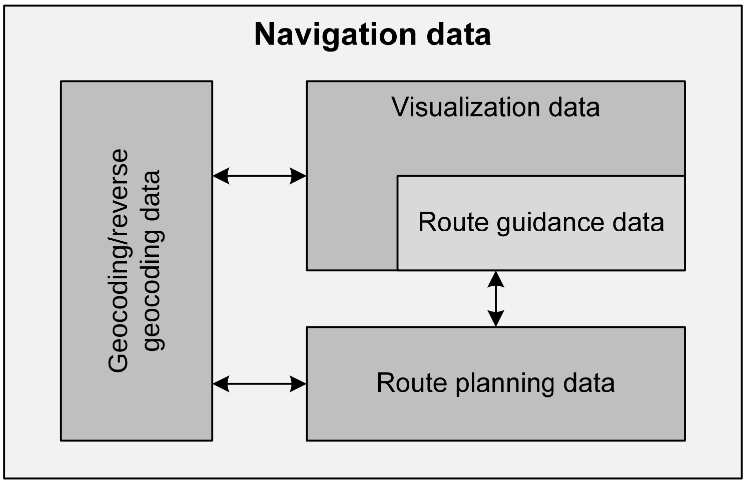


Figure 3.1 Navigation data and its components.

The spatial data type in navigation dataset is traditionally accessible in vector formats. With the emerging new technologies in recent years, other types of spatial data like geo-raster data (e.g. high-resolution satellite imagery), 3D data (e.g. 3D model of city buildings) or even panoramic photographic data (e.g. Google Street View, Bing Streetside) have been integrated

[28] http://en.wikipedia.org/wiki/Geocoding#List_of_some_geocoding_systems

into some latest navigation systems.

It is not trivial to produce a navigation dataset with such high quality and rich contents. The data provider has to traverse the whole workflow from data acquisition, processing, enrichment, production to updating, which is tedious and costly. Well-known navigation data providers in industrial field are NAVTEQ, Tele Atlas and AND. They have been providing navigation data for car-navigation systems since years.

With regard to the formats of navigation data offered by commercial data providers, there are industrial or de facto standards. In which format the data should be organized depends on which service stage the data is intended for as well as the hardware environment. The stage is classified into three types:

- **Operation**. At this stage, the data is on standby to support the navigation functions. All the internal data structures are ready for use by the algorithms including route planning, map rendering, etc. All the data preprocessing steps are completed. However, the data organization varies with different environments. For example, in a resource-constrained device like mobile or embedded system, the data organization has to be as compact as possible to make full use of the storage space. While in a server- or cluster-based high performance computing environment, some spatial database management systems are qualified for this task.

- **Archive**. This stage is mainly for massive storage, backup, archiving the out-of-date dataset or sometimes delivery. At this stage, navigation data are usually organized as files on disks in general formats. All the internal data structures which can be derived from these files are unnecessary to reserve.

- **Exchange**. As an intermediate stage, navigation data are sometimes exported to or imported from such a form when there is a format-mismatch problem between a data producer and a consumer. It requires the data at this stage to be in some standard format which makes it easy to interchange, understand and extend.

Some spatial data formats in common use at different stages are listed in Table 3.1.

Table 3.1 Spatial data formats for navigation data.

Format	Abbr.	Description
ESRI Shapefile	shapefile	Shapefile is a geospatial vector data format for GIS software. It is developed and regulated by ESRI as a mostly open specification for data interoperability among ESRI and other software products. ref. (ESRI 1998).
Kiwi Format	KIWI	KIWI is an open specification of map disc format stored for car navigation system. It is candidate for the standard in ISO TC204/WG3. The most recent edition of KIWI is Ver1.22[29].

[29] http://www.kiwi-w.org/format_english/format_kihon.html

SDAL Format	SDAL	SDAL is a proprietary map format published in 1999 by NAVTEQ, which released it royalty free in the hope that it would become an industry standard for digital navigation maps. The format has not been very widely adopted by the industry.
Physical Storage Format	PSF	The PSF initiative is an industry grouping of car manufacturers, navigation system suppliers and map data suppliers whose objective is to standardize the data format used in car navigation systems, as well as to allow a capability of map updating.
Geographic Data Files	GDF	GDF is an interchange file format for geographic data. In contrast with generic GIS formats, GDF provides an extensive catalog of standard features, attributes and relationships. The most recent edition of the GDF standard is GDF5.0, approved as Draft International Standard (ISO/DIS14825) in June 2010[30]. ref. (Van Essen and Hiestermann 2005)

As mentioned above, different data formats are suitable for different stages. Among them, shapefile is predominantly used for archiving; KIWI, SDAL and PSF are for operation in PND; and GDF is used for exchange.

This work is concentrated on the data component for route planning in a navigation dataset. The navigation dataset used for this work was provided by United Maps GmbH[31] that integrates the data from NAVTEQ and ATKIS (Amtliches Topographisch-Kartographisches Informationssystem, the official topographic cartographic information system in Germany)[32], and produces enriched and navigable vector maps at large map scales up to 1:500. Since the format of navigation data provided by United Maps is shapefile, it is necessary to extract the information and construct the data structures for multimodal route planning. This process is defined as multimodal network data modeling.

3.2 Modeling urban transportation networks

For the sake of consistency, the usage of the terms network, graph, as well as their components are distinguished. As Figure 3.2 shows, *network* refers to the data model consisting of *nodes* and *lines*, as well as their topological relationships, geographic positions and shapes. This implication is consistent with the concept of network data model in GIS (Longley et al. 2010); *graph* refers to the *vertex*/*edge* model in graph theory, which can be extracted from a network and constructed for route planning.

[30] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=54610

[31] http://unitedmaps.net/

[32] http://www.atkis.de

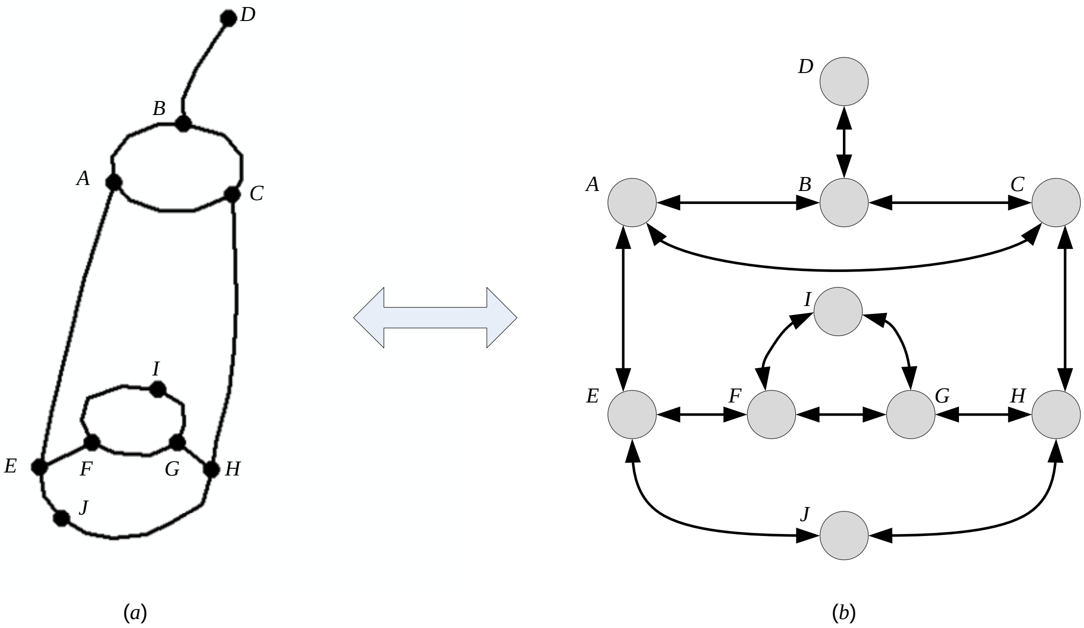


Figure 3.2 Network and graph. (*a*). A sample network with 10 nodes and 13 lines; (*b*). corresponding graph of the sample network with 10 vertices and 13 double-way edges.

The transportation modes considered in this work are motorized road network for private car driving, pedestrian way network for walking and public transit networks including underground, suburban and tram lines for passenger transporting (the cells in italic in Table 3.2).

Table 3.2 Transportation modes and networks.

Mode type	**Mode (abbr.)**	**Transportation network**	
		Functional type	**Carrier type**
Private	*Walking* (*W*)	*Pedestrian-allowed*	Road
	Car driving (*D*)	*Private car-allowed*	
	Bicycle riding	Bicycle-allowed	
	Motorbike	Motorbike-allowed	
	Taxi	Taxi-allowed	
Public	Bus taking	Bus line	
	By underground train (*U*)	*Underground line*	Railway
	By suburban train (*S*)	*Suburban line*	
	By tram train (*T*)	*Tram line*	

It can be observed from Table 3.2 that there is one-to-one relationship between transportation mode and the functional type of transportation network. So it turns out that given a specific mode, a network with identical functional type can be obtained. Such a network is termed as a mode graph according to the following definition.

Definition 1 Mode Graph.
For a given mode m_i*, the corresponding network is modeled as graph* G_i*.* G_i *is called the mode graph of* m_i*.*

Generally speaking, the modeling methods are distinguished mainly between private and public modes. When a subject is moving in a public mode, his movement is restricted by fixed geographic routes and timetables that are not under his control, while he does not have such restrictions in private modes. Therefore, different modeling methods for the two mode classes are deployed.

3.2.1 Road network for private modes

The road network is modeled into several separated mode graphs corresponding to different transportation modes. The reason is that the networks for different modes usually are different in topology and edge cost functions. Figure 3.3 shows a sample road network of Munich in which the city center area is bounded by red dashed line. Both of the motorized and pedestrian networks in the bounded area are shown in Figure 3.4.

Figure 3.3 Sample road network of Munich, the area bounded by red dashed lines is shown in more detail in Figure 3.4.

Figure 3.4 Motorized and pedestrian networks for the same area of center of Munich. (*a*). Motorized network; (*b*). Pedestrian network.

Specifically, the road network is modeled into motorized and pedestrian graphs for the modes car-driving and walking respectively. In some of the existing approaches, the multimodal network is modeled by creating multi-edges, i.e. constructing multigraph. In the implementation, this work sticks to using one edge for each street line in one direction. Therefore, it is not necessary to support multigraph in the data models and routing algorithms, which makes it easier to apply the classic non-multigraph-oriented path finding algorithms on the data model.

In the raw navigation dataset, one physical street line feature may correspond to several modes for saving the storage space. The mode(s) of one road segment can be identified by its attached attributes in the raw dataset. Taking the commercial navigation database NAVSTREETS provided by NAVTEQ for example. Table 3.3 lists nine data fields related to the modes of transportation on road segments. These fields identify the types of traffic allowed on a street line. In the navigation dataset provided by United Maps, these attributes are available because the dataset is a result of integration of NAVTEQ and ATKIS. Among these attributes, the AR_Auto and AR_Pedest are the most useful two for the multimodal network model since they are related to the addressed transport modes.

Table 3.3 Accessible attributes of Streets in NAVSTREETS database.

Attribute	**Field name**	**Type**	**Value**
Access Automobiles	AR_Auto		
Access Buses	AR_Bus		
Access Taxis	AR_Taxis		
Access Carpools	AR_Carpool		
Access Pedestrians	AR_Pedest	Boolean	Y: Is allowed N: Not allowed
Access Trucks	AR_Trucks		
Access Through Traffic	AR_Traff		
Access Deliveries	AR_Deliv		
Access Emergency Vehicles	AR_EmerVeh		

Motorized graph. This graph is used by the mode of private car driving. It consists of vertices and edges. For each line feature in the street line layer whose AR_Auto equals to Y, i.e. allowed, it is modeled as an edge $e = (u, v)$. Note that the vertices may not be junctions in reality. It depends on the digitalization method of the data collector. Since there are one-way roads for vehicles, the directions of edges should be identified. To determine the travel direction of a street line, three attributes from the street line layer are needed (Table 3.4). The Ref_In_ID and Nref_In_ID will be modeled as the tail and head vertices of *e*; that is, *u* and *v*. Both (u, v) and (v, u), (u, v) or (v, u) solely will be added to the graph if Dir_Travel value is B, F or T respectively.

Table 3.4 Three attributes used for determining the travel direction of a street line feature.

Attribute	Field name	Type	Value
Reference Node ID	Ref_In_ID	Numeric(10)	
Non-Reference Node ID	Nref_In_ID	Numeric(10)	
Direction of Travel	Dir_Travel	Char(1)	B: both directions F: From reference node T: to reference node

Two cost values – distance and average travel time – are assigned to each edge. The distance value of a street line feature is not directly provided in the road layer database. But it can be calculated by summing up the Great Circle Distances (Equation (1)) between each pair of adjacent shape points along the street polyline.

$$d((x_1, y_1), (x_2, y_2)) = r \arccos\left[\sin\frac{\pi y_1}{180}\sin\frac{\pi y_2}{180} + \cos\frac{\pi y_1}{180}\cos\frac{\pi y_2}{180}\cos\frac{\pi(x_2 - x_1)}{180}\right] \tag{1}$$

where r=6378137.0 meters according to the spatial reference system of the raw dataset, (x_1, y_1) and (x_2, y_2) are the longitude/latitude pairs of the two points with decimal degrees as their units. As a result, the distance cost of an edge is in meters.

Although the travel times on street lines of vehicles are highly dynamic, which means the travel time on one street line is different from one vehicle to another and from hour to hour (and further influenced by some temporary events on roads such as randomly occurred congestions or accidents), the underlying graph is kept static by assigning statistically observed average travel time. And this average travel time can be acquired from the navigation database by retrieving the Speed_Cat attribute (Table 3.5). Since the values of Speed_Cat are given as a series of numeric ranges, the arithmetic means of them is taken as average speeds for the street lines whose Speed_Cat value is between 2 and 7 (see the Average speed column in Table 3.5), while take 140 and 6 for 1 and 8 respectively. After that, the values of average speed are transformed into those with meters per minute as their units. So the average travel time (in minutes) can be obtained by dividing the distance value by the average speed.

Table 3.5 Attribute used to determine the average driving speed on a street line feature.

Attribute	Field name	Type	Value km/hour	Average speed km/hour	Average speed m/min
Speed Category	Speed_Cat	Char(1)	1: > 130	1: 140	1: 2333.3
			2: 101-130	2: 115	2: 1916.7
			3: 91-100	3: 95	3: 1583.3
			4: 71-90	4: 80	4: 1333.3
			5: 51-70	5: 60	5: 1000.0
			6: 31-50	6: 40	6: 666.7
			7: 11-30	7: 20	7: 333.3
			8: < 11	8: 6	8: 100.0

Pedestrian graph is constructed in the same manner as motorized graph. However, in conventional navigation database intended for car navigation, pedestrian-only ways are normally excluded. The main reason is the street line data are collected by cars with GPS devices which are unable to reach the pedestrian-only area. For multimodal route planning, however, the pedestrian mode is no longer ignorable. Therefore, a sufficient pedestrian network with high coverage is necessary for the modeling process. Ideally, the pedestrian network should be as detailed as possible, e.g. it should contain information about individual platforms of public transit stations and individual parking lots. By integrating the conventional navigation database for cars and Digital Landscape Model (DLM) data owned by governmental surveying agency, United Maps provides a dataset where quite a few pedestrian-only ways are added. Table 3.6 shows the amount of enrichment after the integration process carried out by United Maps for an area out of the south of Bavaria State in Germany (ca. 10000 km^2). Apart from the link features provided by United Maps, the platform information of underground stations in Munich is manually collected by the author.

Table 3.6 Comparison between NAVTEQ and United Maps navigation datasets.

	NAVTEQ	United Maps	Percentage of enrichment
Total length of road (m)	16,863,892	39,799,075	136%
Number of spatial objects	110,575	246,236	123%

Basically, for each link in the street line layer whose AR_Pedest equals to Y, it is modeled as an edge $e = (u, v)$. The edge is double-directed. The distance cost of the edge is calculated in the same way as that in motorized graph. According to the research and statistics from transportation and human behavior fields (Transafety 1997, Carey 2005), the average human walking speed is about 4.8 km/h, that is 80 m/min. Thus, this value is taken for calculating the average travel time cost of an edge.

3.2.2 Public transit network

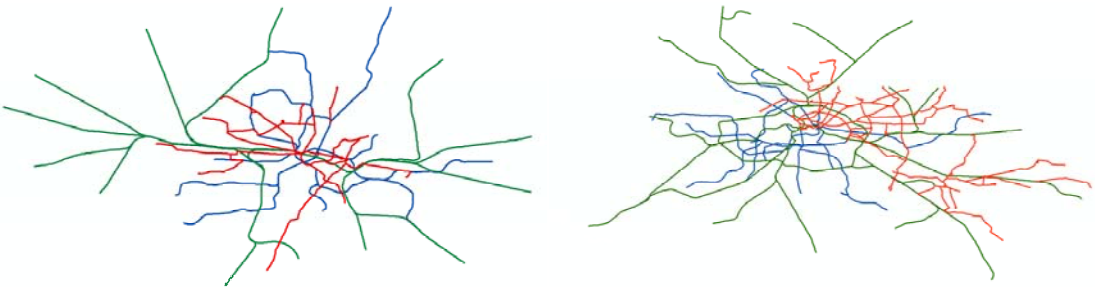
There are several approaches for modeling railway networks which can give good references to the urban public transit network modeling. The timetable is usually taken as the anchor point for public transit network models, and great efforts have been made to explore how the time-dependent schedule information can be best integrated with the time-independent physical networks.

Instead of embedding the timetable information in the public transit network as dynamic edge costs, these two parts are kept separated in the model. This section concentrates solely on the static public transit network. The timetable will be discussed in the modeling of mode-switching actions (Section 3.3) for the purpose of calculating waiting times at stations. The public transit modes relevant for the study in this work include underground, suburban and tram lines as shown in Figure 3.5.

(*a*) (*b*) (*c*)

Figure 3.5 The public transit vehicles on railway in Munich. (*a*). underground train (U-Bahn)[33]; (*b*). suburban train (S-Bahn)[34]; (*c*). tram (Tram or Straßen-Bahn)[35].

Figure 3.6 shows sample public transit networks composed of the three different modes in Munich and Berlin. Note that the bus lines can be modeled with the same method as introduced in this section although they are excluded because United Maps has not integrated them yet.



(*a*) (*b*)

Figure 3.6 Example of public transit networks where suburban, underground and tram lines are denoted in green, blue and red respectively. (*a*). Munich; (*b*). Berlin.

Keeping timetable information separated, the public transit network can be modeled in a similarly straightforward and concise way to the road network. However, instead of modeling each public transit link between two stations as an edge in the corresponding mode graph directly, preprocessing is necessary due to the special properties of the public transit stations and lines. Before introducing the preprocessing , it is necessary to clarify the terminology

[33] http://www.flickr.com/photos/woodpeckar/86130828

[34] http://commons.wikimedia.org

[35] http://www.tram-muenchen.de

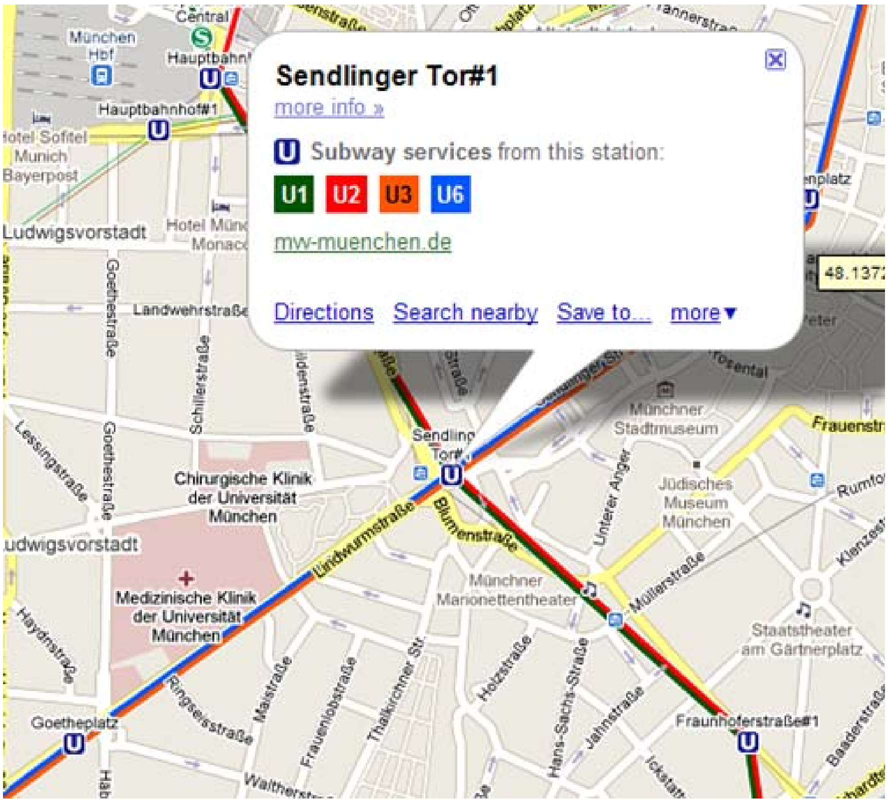
with regard to public transit networks.

A *public station*, or *station*, is a place where vehicles pick up or drop off passengers. A station must contain one or more *platforms* where vehicles stop and passengers can get on the vehicles. Incident stations are connected by *transit link*(*s*) which are fixed on roads or railways. If a sequence of stations is designated to be served by one specific route of a mode, this route is called a *service line* (e.g. U2, S8, Tram27).

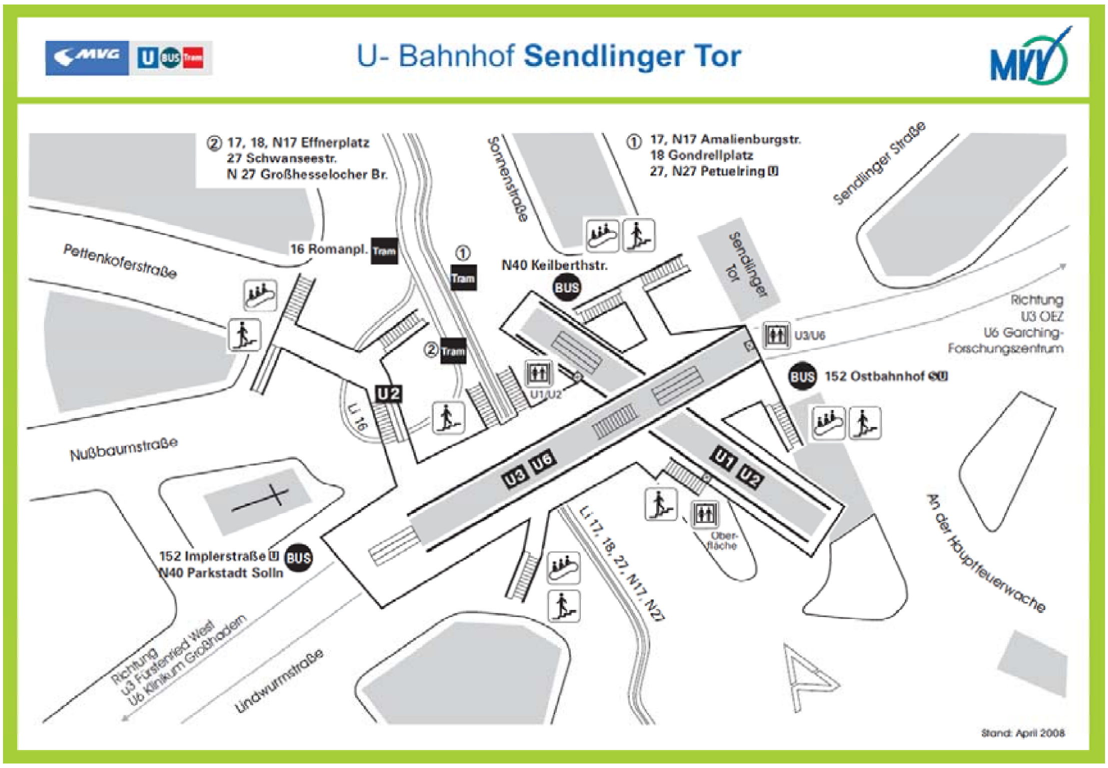
The preprocessing is essentially composed of the following three separations:

Separation of shared stations by different service lines. Two important objectives of designing an urban public transportation system are: 1). to bring more convenience to travelers who choose the public transit networks; 2). to promote the transport efficiency of the system as much as possible. As a result, public transit system planners tend to build some transport hubs where two or more service lines intersect with each other. In other words, the public stations designed to be hubs are shared by different service lines, which makes it easy and efficient to transfer between the lines.

Taking Sendlinger Tor underground station in Munich for example (Figure 3.7), four underground lines – U1, U2, U3 and U6 – intersect here.



(*a*)



(*b*)

Figure 3.7 Cartographic presentation of Sendlinger Tor underground station. (*a*). from Google Maps[36]; (*b*). from MVV[37].

In the raw dataset, Sendlinger Tor station is digitalized as one single point feature but attributed as shared by the four underground service lines. Such kind of shared stations are modeled by splitting them into different vertices belonging to each specific service line respectively. Figure 3.8 shows the example of Sendlinger Tor.

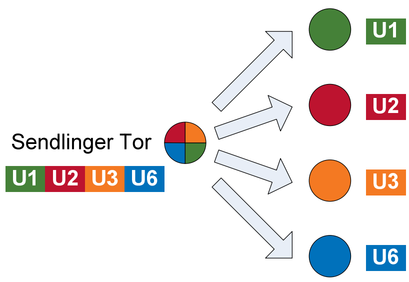


Figure 3.8 Separation of Sendlinger Tor station into four vertices belonging to U1, U2, U3 and U6.

Separation of shared transit links by different service lines. Between some of the station pairs, there are two or more transit links. For example, U1 and U2 share the underground railway line segment from Hauptbahnhof to Sendlinger Tor station with staggered scheduled

36 http://goo.gl/maps/M4WY

37 http://www.muenchnerubahn.de

trains. For each shared railway line segment, it is digitalized into one polyline feature (with attributes attached as that for the stations) in the raw database. Such sort of transit links are split into individual ones belonging to different service lines. Figure 3.9 shows this operation for the above mentioned example.

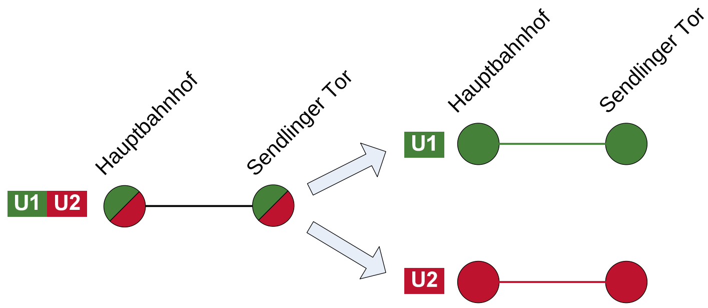


Figure 3.9 Separation of underground lines U1 and U2 between Hauptbahnhof and Sendlinger Tor stations.

Separation of the two heading directions of one service line. A normal public service line serves both directions. However, to model the two different directions of one service line between two stations into two edges with opposite directions connecting a pair of vertices is inappropriate because it may imply that there will be no impedance for changing the travel direction within one service line during a trip, which is usually not true. In reality, it takes people some time to change from one platform to another and wait for the next train going to the other direction. In order to take the costs of such directional change into account, each station vertex is split into two for both heading directions of the service line.

Figure 3.10 shows the example of separation operation on Sendlinger Tor underground station, which is the final result of the preprocessing. The separation of the underground lines U1 and U2 between Hauptbahnhof and Sendlinger Tor stations are illustrated in Figure 3.11 accordingly.

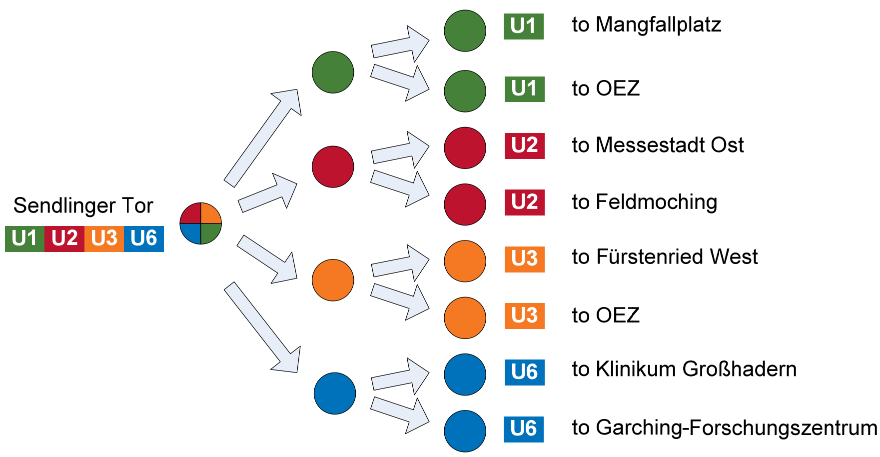


Figure 3.10 Further separation of Sendlinger Tor station according to the different heading directions.

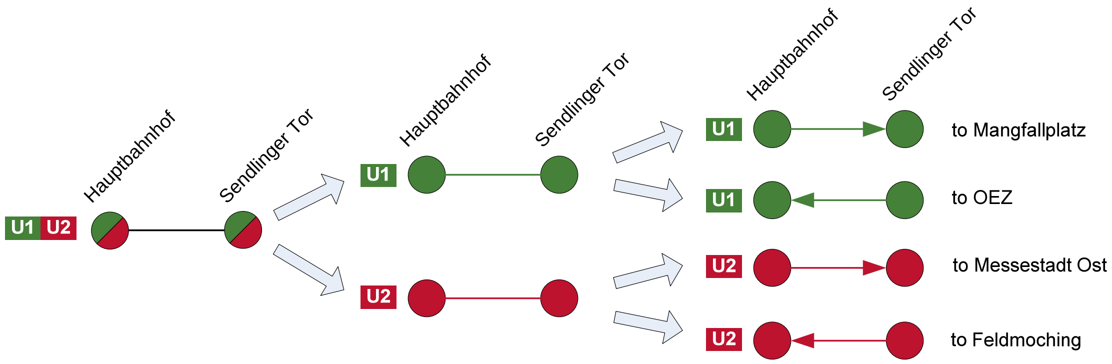


Figure 3.11 Separation of underground lines U1 and U2 between Hauptbahnhof and Sendlinger Tor according to their different heading directions.

To sum up, the graph for public transit network is constructed by inserting an edge $e = (u, v)$ for one transit link segment from one station to another. Any vertex *u* in the graph indicates the anchor position of a specific service line heading either direction. With regard to the edge weight in public transit network, the travel time cost value can be easily assigned since the transport time between any two stations is predefined in the service line schedules. For the other cost value – distance, it is recorded in consistence with the edge structure of road network although this value is hardly used for route planning in public transit networks.

It can be observed that this modeling method produces a graph per public transit mode, which is unconnected however. In fact, this connectivity "problem" occurs not only here, but also between the road and public transit network graphs. To allow the switching between such unconnected graphs, this thesis work proposes the concept of Switch Point and compares them to the "plugs and sockets" connecting different mode graphs.

3.3 Mode-switching action

In modern urban transportation systems, many intermodal facilities such as parking places, park and ride lots, transit hubs, trailheads, etc. are provided besides the basic transportation networks as it has been introduced in Section 2.1. These facilities make it easy to transfer between different transportation modes. With the collection and digitalization of intermodal facilities from the real world and the integration of their information with navigation databases, it becomes possible to conduct automatic multimodal route planning.

The concept of Switch Point was proposed by the author to abstract the intermodal facilities as points where a travel mode can switch from one to another. Based on this concept, multimodal path-finding algorithms can be designed to solve the general multimodal shortest path problem (Liu and Meng 2009). *Switch Point* refers to the spots where transferring from one mode to another takes place. A matrix, namely Switch Point Matrix, was adopted to express all the feasible switching relationships. Whether a vertex *v* is eligible to be a Switch Point from mode *i* to mode *j* or not can be determined by checking two conditions: 1). Is *v* accessible by both G_i and G_j? 2). Is the amenity attribute value on *v* equals to the (i, j) element in Switch Point Matrix? The Switch-Point-based data model makes it possible to develop multimodal shortest-path algorithms which can make good use of the traditional label-setting and label-correcting algorithms.

However, our original Switch-Point-based approach will meet some difficulties to plan an optimal route in a real-world transportation network where all the urban transportation means must be considered. First, it requires the modes be given sequentially as a part of the input. In real-world applications, it is not always possible to determine a mode sequence in advance. Users will experience it as a bothersome task to type in a mode sequence by themselves. They will more likely expect the system to provide an optimal mode combination. Second, it is difficult to describe Switch Points containing complex switching relations and various switching costs. For example, a parking lot for car may contain two entrances for cars and three exits for pedestrians as illustrated in Figure 3.12. There is no shared vertex accessible by both the motorized and pedestrian graphs. Instead, the vertices in the switch-from and switch-to graphs reveal a many-to-many relation ($2:3$ in the example). Every entrance/exit pair can be a possible switching from car driving to walking. Furthermore, this switching process needs time for finding a vacant parking lot which is usually not free of charge. Third, only intermodal switching actions were considered and allowed. The intro-modal switching actions, which are existent in reality, were excluded in the discussion and prohibited by assigning NIL values on the diagonal of Switch Point Matrix. It leads to the difficulty of modeling such switching action as transferring from car-driving to itself at some gas station.

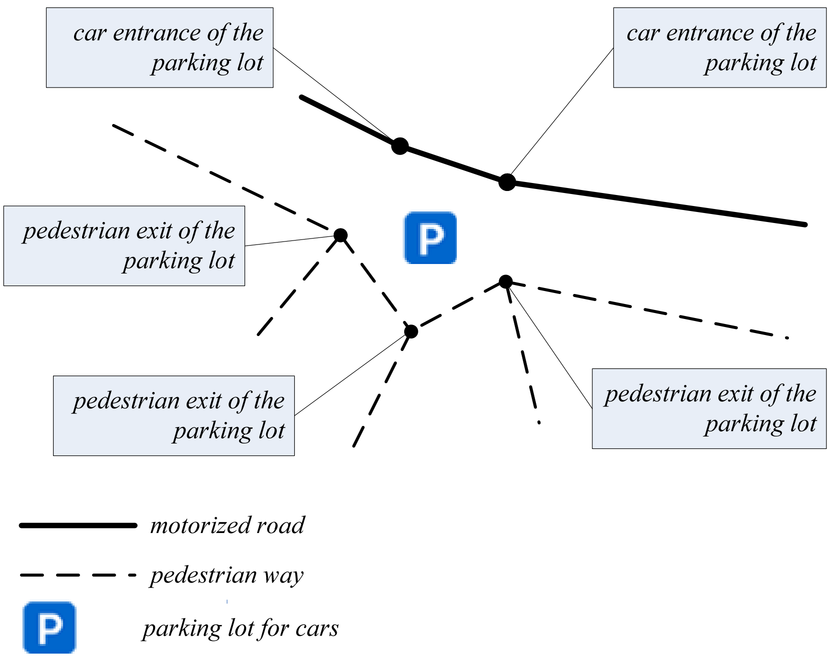


Figure 3.12 A parking lot containing two car entrances and three pedestrian exits.

To overcome these difficulties, the concept of Switch Point needs to be extended and formally described in detail.

3.3.1 Switch Point

Conceptually, the term Switch Point is abstracted from the intermodal facilities in transportation realm. Concrete examples of Switch Point in the real life can be parking places, park and ride lots, bike and ride lots, public transit stations, etc. The travel mode switching can occur at these places when people plan a multimodal route. In the author's previous work (Liu and Meng 2009), Switch Point Matrix (SPM) was defined to record all the special conditions needed to be fulfilled by a switching action between two different modes. The *special conditions* were illustrated by the example of SPM in Transportation (SPM-T). By looking up in the SPM, the answer to the question "*What kinds of facilities can I use when changing from mode A to mode B?*" can be given. It turns out that the elements in the matrix indicate the feasible types of switching actions. Thus, the special condition is defined as Switch Type.

Definition 2 Switch Type.
A Switch Type λ is a class corresponding to one type of possible mode-switching actions in a specific domain. $\Lambda = \{\lambda_i | i \in \mathbb{N}\}$ consists of all Switch Types in that domain.

For an arbitrary mode pair, a subset $\Lambda' \subseteq \Lambda$ can be determined based on domain-specific knowledge. The original SPM can be scrutinized to a more general form – Switch Type Matrix (STM, see Equation (2) where *N* is the number of modes). The STM-T (originally called SPM-T) is an instance in the domain of transportation.

$$\mathrm{STM} = [\Lambda'_{i,j}]_{N \times N} \tag{2}$$

Note that Λ' can be $\varnothing$, which is corresponding to the NIL-elements in the original expression of SPM. Λ' can also contain more than one element, which means there are different possible Switch Types between two modes. Taking the Λ' in STM-T between the modes of car driving and walking for example, there are at least two Switch Types. If the user is a driver, an available parking lot is necessary for this double-modal route and thus all the vertex pairs meeting this condition are eligible for switching. However, if the user is only a passenger which means somebody else drives the car (e.g. by taxi), any position allowed for temporary parking in the road network can be used for mode switching in this case.

Let $M = \{m_i | i \in \mathbb{N}\}$ be a set of modes, Λ a set of Switch Types, $G_M = \{G_i = \{V_i, E_i\}\}$ the Multimodal Graph Set (MMGS) consisting of the graphs with each corresponding to a mode in *M*, $\boldsymbol{V} = \bigcup V_i$, $\boldsymbol{E} = \bigcup E_i$, and C a set of switching-cost functions. The following concepts can be defined.

Definition 3 Switch Relation.
A Switch Relation Γ is a 6-ary relation on $M \times M \times \Lambda \times \boldsymbol{V} \times \boldsymbol{V} \times C$ defined as follows:

$(m_f, m_t, \lambda, v_f, v_t, c) \iff$ *a switching action from vertex v_f in the graph of mode m_f to vertex v_t in the graph of mode m_t, with the type of λ and cost c.*

Definition 4 Switch Point.
Given a Switch Relation Γ, a Switch Point γ is identified by a 6-tuple in Γ.

Definition 5 Switch Condition.
Given a Switch Relation Γ, a Switch Condition φ is a set of criteria expressed by a propositional formula applying on Γ.

From the point of view of a relational database system, Γ can be implemented as a table consisting of a set of 6-tuples expressed by γ. And φ can correspond to the predicates in a WHERE clause of SQL selection operation.

If the parking lot example at the beginning of this section is reviewed once again, an attributed expression can be given as shown in Figure 3.13. And based on the given definitions, six eligible Switch Point tuples in this case are listed in Table 3.7 expressing a subset of Γ.

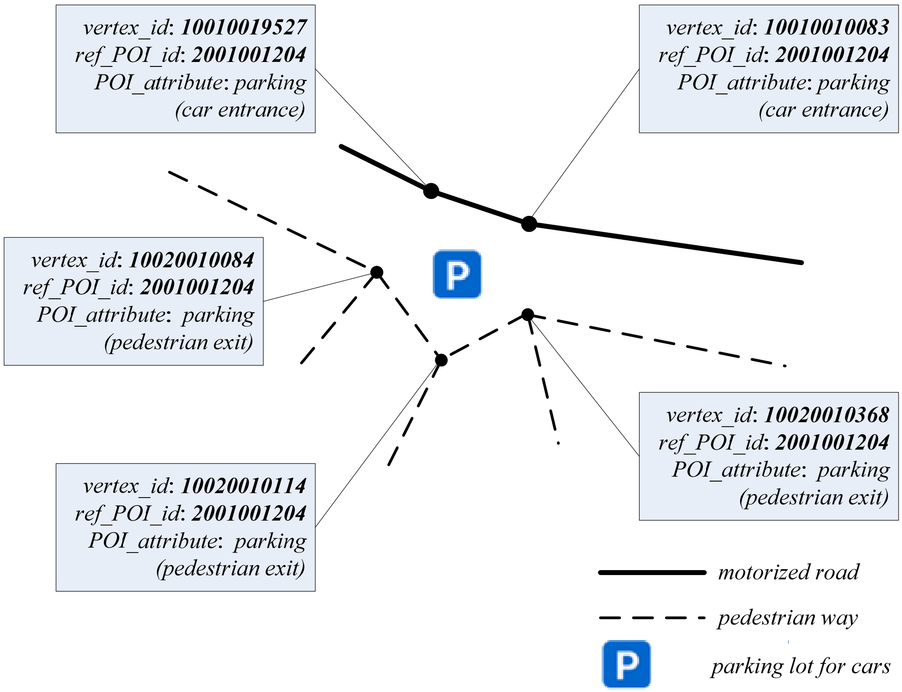


Figure 3.13 Attributed expression of the example in Figure 3.12.

Table 3.7 Eligible Switch Points in Figure 3.13.

from_mode (m_f)	to_mode (m_t)	type (λ)	from_vertex (v_f)	to_vertex (v_t)	cost* (c)
car_driving	walking	car_parking	10010019527	10020010084	2.5
car_driving	walking	car_parking	10010019527	10020010114	4
car_driving	walking	car_parking	10010019527	10020010368	3.5
car_driving	walking	car_parking	10010010083	10020010084	3.5
car_driving	walking	car_parking	10010010083	10020010114	4.5
car_driving	walking	car_parking	10010010083	10020010368	2

*the metric of the cost in this table is an estimated time (in minutes) of the corresponding switching

3.3.2 Time-dependency on Switch Point

Switch Point can be time-dependent. The time-dependency is mainly represented by time-variant attributes of Switch Points including numeric and non-numeric fields.

The cost of a switching, which is one typical numeric attribute of Switch Point, is sometimes time-dependent. Taking the switching action from walking to public transit system as an example, the switching cost *c* is denoted by t_S to indicate the mode-switching delay in this case. t_S depends on the time interval between the user's arrival time τ_a at the station and the departure time of the next train/bus τ_n, which is expressed by Equation (3).

$$t_S = \tau_n - \tau_a \tag{3}$$

To calculate the consumed time of a planned route more accurately, the time-dependent mode-switching delay must be considered. The separation of mode-switching actions and basic networks brings convenience to the modeling work of the time-dependent switching cost because it allows static weights on network edges on the one hand and dynamic costs assigned to Switch Points on the other hand.

τ_a can be calculated by adding the consumed time so far t_a to the departure time at the origin τ_o. And τ_n can be acquired by looking up in the timetable of the station. As a result, Equation (3) can be rewritten as Equation (4).

$$t_S = \tau_n - (\tau_o + t_a) \tag{4}$$

Consequently, the integration of timetables of the public transit system is essential to the switching cost calculation. It is a natural way to publish the timetable information by station name, line number and heading direction. Most of the public transit agencies adopt this format for their timetables. Figure 3.14 shows a sample timetable of the underground station Theresienstraße in Munich. People can easily know when the next train/bus will come by keeping the weekday and current time in mind and looking up in the table. This modeling approach is consistent with this natural way of human cognition.

Unlike previous timetable modeling methods (Pajor 2009) which record the travel time between adjacent stations, solely the departure times for each public stations are recorded. Formally, a public transit timetable is a 6-ary relation on $\mathcal{S} \times \mathcal{S} \times \mathcal{S} \times \mathcal{L} \times \mathcal{W} \times \mathrm{T}$ where $\mathcal{S}$ is a set of stations, $\mathcal{L}$ a set of service lines, $\mathcal{W}$ a set of weekdays (from Monday to Sunday) and T a set of departure times. A 6-tuple $(S, S_{dir}, S_{dest}, L, W, \tau_n)$ is interpreted as a train/bus of line $L \in \mathcal{L}$ going from station $S \in \mathcal{S}$ with the destination station $S_{dest} \in \mathcal{S}$, heading the direction of station $S_{dir} \in \mathcal{S}$, departing on weekday $W \in \mathcal{W}$ at time $\tau_n \in \mathrm{T}$.

It should be noted that the departure time τ_n is periodical, e.g. in 24 hours of a day from 0:00 to 23:59. The periodicity of τ_n is denoted by Π. Further, the calculation of waiting time t_S at a public station has to account for this periodicity. If the user's arrival time τ_a is less than the train/bus's departure time τ_n then the waiting time can be simply calculated by Equation (3). However, it is also possible for a user to arrive at the station at night and the coming train/bus will depart during the next day. In this case, $\tau_n < \tau_a$ holds and the waiting time is composed of two parts: 1) from τ_a until midnight; 2) from midnight until τ_n. With regard to the time period Π the waiting time improved from Equation (3) can therefore be obtained.

$$t_S = \begin{cases} \tau_n - \tau_a & \text{if } \tau_n \geq \tau_a, \\ \tau_n + \Pi - \tau_a & \text{otherwise.} \end{cases} \tag{5}$$

Figure 3.15 shows a typical underground piecewise linear function of waiting time acquired from the real timetable in Figure 3.14. The time period between 19:00 and 20:10 is selected.

Gültig ab 13. Dezember 2009

U2 Feldmoching Ⓢ - Hauptbahnhof Ⓢ - Kolumbusplatz - Giesing Ⓢ - Innsbrucker Ring - Trudering Ⓢ - Messestadt Ost ➡

MVV

Theresienstraße

Feldmoching Ⓢ Ⓡ – Theresienstraße – Königsplatz 01 – Hauptbahnhof Ⓢ Ⓡ 03 – Sendlinger Tor 05 – Fraunhoferstraße 06 – Kolumbusplatz 08 – Silberhornstraße 09 – Untersbergstraße 10 – Giesing Bahnhof Ⓢ 11 – Karl-Preis-Platz 13 – Innsbrucker Ring 15 – Josephsburg 17 – Kreillerstraße 18 – Trudering Ⓢ 20 – Moosfeld 21 – Messestadt West 23 – Messestadt Ost 25

Fahrzeit in Minuten

Zeitkartenringe: 1/2 – 1 – 1/2 – 2 – 2/3 – 2 – 2/3 – 3 – 3/4

Uhr	Montag-Donnerstag (Schule)	Freitag (Schule)	Montag-Donnerstag (Ferien)	Uhr
4	26	26	26	4
5	07 27 37 47 57	07 27 37 47 57	07 27 37 47 57	5
6	07 17 27 37 47 57	07 17 27 37 47 57	07 17 27 37 47 57	6
7	02 07 12 17 22 27 32 37 42 47 52 57	02 07 12 17 22 27 32 37 42 47 52 57	02 07 12 17 22 27 32 37 42 47 52 57	7
8	02 07 12 17 22 27 32 37 42 47 52 57	02 07 12 17 22 27 32 37 42 47 52 57	02 07 12 17 22 27 32 37 42 47 52 57	8
9	02 07 12 17 22 27 34 37 44 47 57	02 07 12 17 22 27 34 37 44 47 57	02 07 12 17 22 27 34 37 44 47 57	9
10	07 17 27 37 47 57	07 17 27 37 47 57	07 17 27 37 47 57	10
11	07 17 27 37 47 57	07 17 27 37 47 57	07 17 27 37 47 57	11
12	07 17 27 37 47 57	07 17 27 37 47 57	07 17 27 37 47 57	12
13	04■ 07 14■ 17 24■ 27 34■ 37 44■ 47 54■ 57	07 17 27 37 47 57	07 17 27 37 47 57	13
14	04■ 07 14■ 17 24■ 27 34■ 37 44■ 47 52 57	07 17 27 37 47 57	07 17 27 37 47 57	14
15	02 07 12 17 22 27 32 37 42 47 52 57	07 17 27 37 47 57	07 12 17 22 27 32 37 42 47 52 57	15
16	02 07 12 17 22 27 32 37 42 47 52 57	07 17 27 37 47 57	02 07 12 17 22 27 32 37 42 47 52 57	16
17	02 07 12 17 22 27 32 37 42 47 52 57	07 17 27 37 47 57	02 07 12 17 22 27 32 37 42 47 52 57	17
18	02 07 12 17 22 27 32 37 42 47 52 57	07 17 27 37 47 57	02 07 12 17 22 27 32 37 42 47 52 57	18
19	02▼ 07 12▼ 17 22▼ 27 37 47 57	07 17 27 37 47 57	02▼ 07 12▼ 17 22▼ 27 37 47 57	19
20	07 17 27 37 47 57	07 17 27 37 47 57	07 17 27 37 47 57	20
21	07 17 27 37 47 57	07 17 27 37 47 57	07 17 27 37 47 57	21
22	07 17 27 37 47 57	07 17 27 37 47 57	07 17 27 37 47 57	22
23	07 17 27 37 47 57✕	07 17 27 37 47 57▼	07 17 27 37 47 57✕	23
0	07 27 47	07 27 47	07 27 47	0
1	08 38[V97]	08 38	08 38[V97]	1
2	08[V97]	08	08[V97]	2

✕ = bis Sendlinger Tor ■ = bis Kolumbusplatz ▼ = bis Innsbrucker Ring V97 = Nächte vor Feiertagen, auch 14./15. und 15./16.02.2010

Ferienfahrplan: 24.12.09 - 05.01.10; 15. - 20.02.10; 29.03. - 10.04.10; 14.05.10; 25.05. - 05.06.10; 02.08. - 13.09.10; 02.11. - 05.11.10

Am 24. und 31. Dezember Betrieb wie Samstag

www.mvv-muenchen.de

Änderungen vorbehalten SMS-Abfahrten: 120 an 0175 4321409 SWM/MVG Tel.: 01803/442266 9Ct./M

Figure 3.14 A sample timetable (valid from 13.12.2009) of the underground station Theresienstraße (heading Munich Central Station) during school days in Munich provided by MVV.

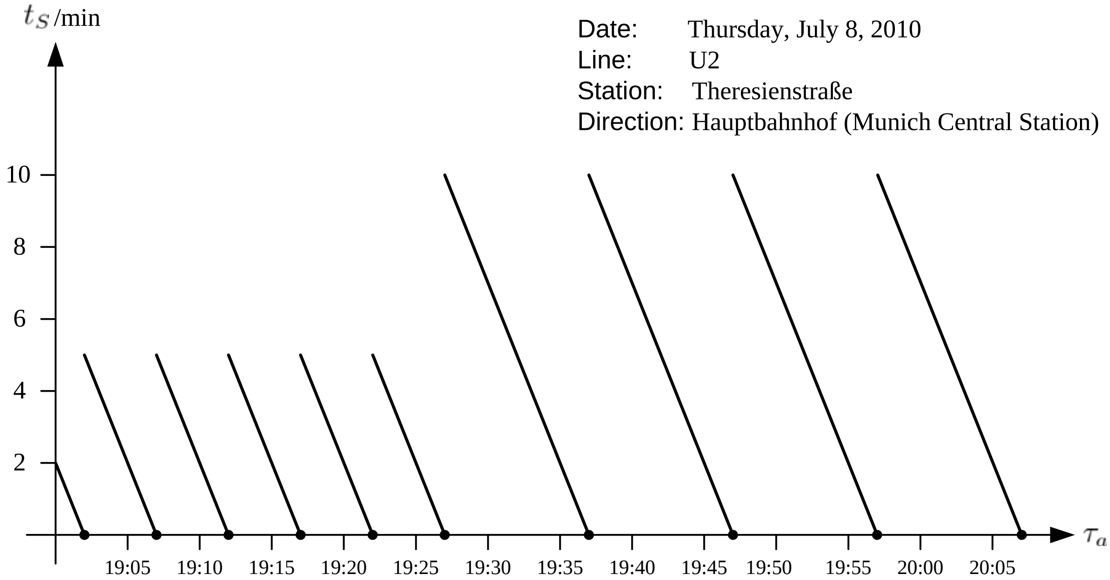


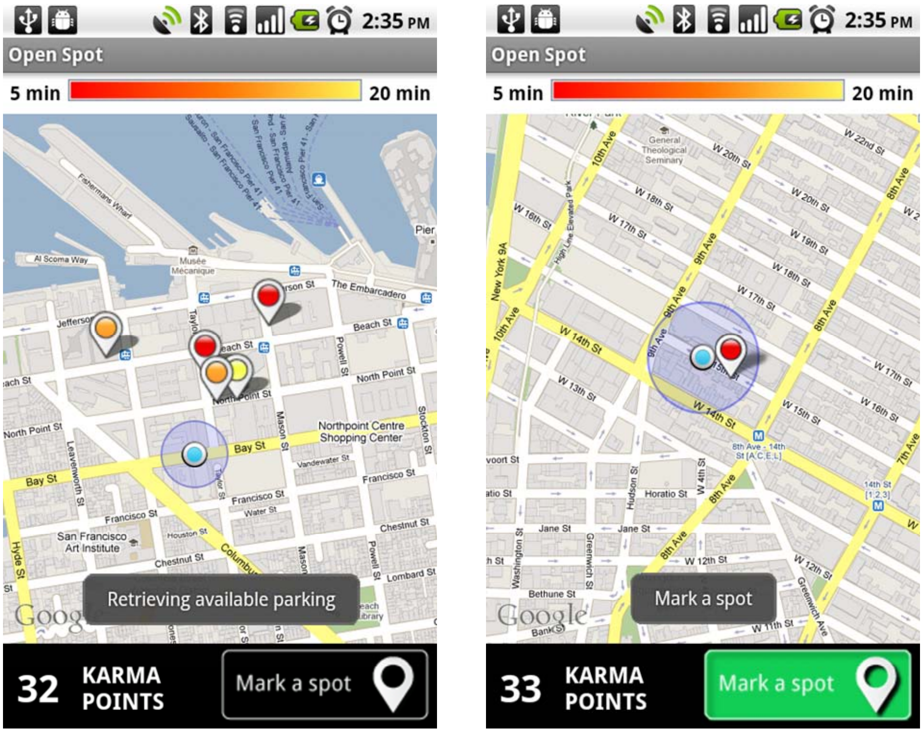
Figure 3.15 Piecewise linear function of waiting time at Theresienstraße station of underground line 2 on Thursday, July 08, 2010 according to the timetable in Figure 3.14. The underground train comes every five minutes from 19:02 to 19:27, and every 10 minutes from 19:37 to 20:07.

With regard to the dynamics of a Switch Point, not only the cost of it can be time-dependent, but the availability attribute may vary with time. A typical example is that when planning a driving-walking double-modal route with a parking place as the Switch Point, it is necessary to make sure that the planned parking place has vacant parking lots when the driver arrive there. The ratio of vacant parking lots in a parking place is apparently time-variant as vehicles keep arriving and leaving. In some cities (e.g. Munich), the number of available parking lots in some park houses is displayed on electronic panels outside Figure 3.16 shows the photo of such a sign taken at the crossing of Gabelsbergerstraße and Türkenstraße in Munich. It displayed that there were no parking lots vacant on Altstadtring Ost, while 1748 available on Altstadtring West at that time.

Figure 3.16 Electronic panels of available parking lots on Altstadtring in Munich at 2:07 p.m.,

November 20, 2010.

In the real-life, looking for an available parking lot is usually a frustrating job for drivers. Recently, Google Inc. released an application – Open Spot – to help drivers find, mark and share available parking spots on Android mobile platform (see Figure 3.17 for the screenshots of the application).



(*a*) (*b*)

Figure 3.17 Open Spot application[38] for finding and sharing parking information. (*a*). Find parking nearby; (*b*). Share an available spot with others.

Such dynamic information on Switch Point can be expressed by attributes varying with time, and should be taken into account when planning multimodal routes.

3.3.3 Constraints on Switch Point

Constraints on Switch Points or *Switch Constraints* can be applied to control the behavior of multimodal path finding. For example, the simplest constraint on a driving-walking double modal trip with the necessity of a parking lot might be: the parking lot should be free. The eligible Switch Points subjected to such kind of simple constraints can be obtained by executing query with the constraints added in the Switch Conditions. More complex constraints may be related to the trip source, or dependent on time.

Source-related constraints

During a trip, some measurements are accumulating as the trip carries on. They can be travel

[38] Open Spot Home in Google Labs: http://openspot.googlelabs.com/

distance, travel time, fare, inter-/intra-modal switch times, fuel consumption, carbon emission, etc. Their values begin to count from the beginning of a trip and increase along the trip. When planning a multimodal route, some constraints in terms of the accumulative values along the trip on Switch Points can be introduced. Since these constraints are associated with the trip source, they are called *source-related constraints*. A novel example of a source-related constraint came out with the emerging electric vehicles which are charged in a different way from that for fuel-driven vehicles (see Figure 3.18). From the technical specification of a typical type of electronic car – MINI E – from BMW (2010), two constraints can be derived, which may restrict the application of electronic vehicles for route planning to some extent:

- the driving distance is limited (ca. 250 km after a full charge);
- the recharging time is long (ca. 26.5 hours for normal charge, 3 hours for fast charge).

Figure 3.18 A MINI E car is in charging.[39]

Other examples of source-related constraints are: walking distance/time limit; valid travel time/number of stations/rings restriction constrained by the user's ticket for public transit. The source-related constraints can be expressed by some restrictions on the temporary value labels on vertices, which can be provided outside the route calculation library by clients. It is possible to express not only single-attribute constraints, but also the composite ones. From path-finding point of view, the partial multimodal paths that do not meet the constraints are invalid and unnecessary to be proceeded any more.

Time-dependent constraints

There are also time-dependent constraints applicable on Switch Points. Some of the switching actions are not available or not allowed during some time span; that is, they should be eliminated if the route planning algorithm attempts to do modal switching action at such places at specific time. For example, bicycles are not allowed to take in underground and suburban trains from Monday to Friday during the time spans of 6:00 – 9:00 a.m. and 4:00 – 6:00 p.m. in Munich according to the rules by MVV. Similar to the source-related constraints,

[39] http://www.worldcarfans.com

time-dependent constraints can be expressed by some restrictions on the temporary values at the arrival time τ_a on vertices. And the multimodal path via a Switch Point will be discarded if this switching action is invalid at time τ_a.

3.3.4 Switch Point modeling

After the mode graphs are constructed, necessary information can be extracted from the raw navigation dataset to model Switch Point tuples that should fill the Switch Relation table. The modeling processes may vary with different types of switching. The parking lot for cars and public station (underground station in this example) are taken as examples to show the process. Besides the modeling workflow used in the real engineering practice, the ideal procedure of modeling Switch Point is introduced as well. The distance existing between the reality and ideality is mainly caused by the lack of data. Although the model of Switch Point in this work can support many-to-many complex switch relationship, such information of the intermodal facilities is hardly provided in the conventional navigation database. Hitherto, none of the transportation datasets has such detailed information. The common way of organizing the intermodal facilities is to store them in a POI layer. As a consequence, this process can also be regarded as the integration of POIs with underlying networks.

Parking lot for cars.

As a standalone POI layer, Car_Parkings contains some basic information of the parking lots with the geometric type of Point. For every parking lot point feature, a Switch Point tuple is created. Then, the nearest-neighbor search operation is applied in the motorized and pedestrian network one after another, and the two nearest nodes belonging to car driving and walking modes respectively can be obtained. After that, the vertices in the two mode graphs corresponding to the searched nodes are recorded as *from_vertex* and *to_vertex* of this Switch Point tuple. An estimated time cost of parking is assigned (five minutes in our practice). In addition, the availability of this parking lot and its ID are recorded as well. Figure 3.19 shows an example of constructing a Switch Point tuple for the parking lot whose ID is 100129964207. The result record is shown in Table 3.8.

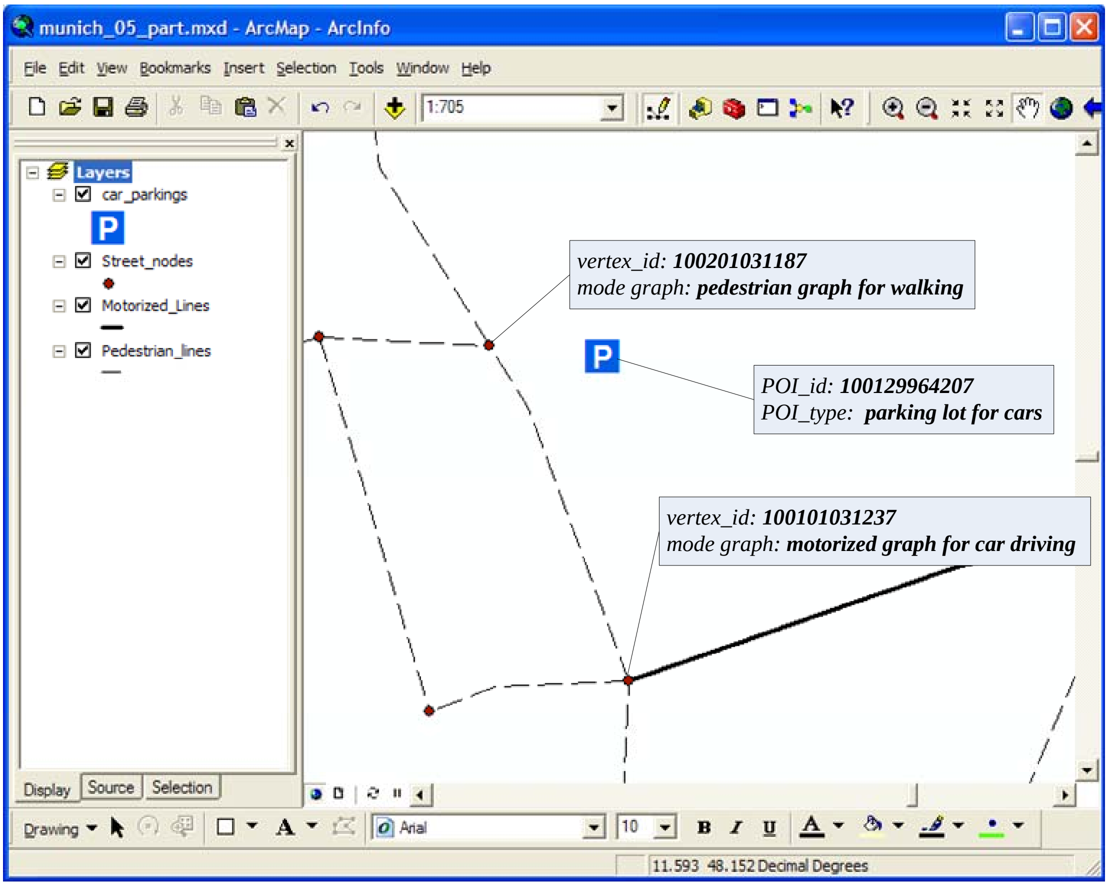


Figure 3.19 A parking lot feature and the nearby road network in the sample dataset of Munich.

Table 3.8 Switch Point tuple constructed from the parking lot in Figure 3.19.

from_mode	to_mode	type	from_vertex	to_vertex	is_available	ref_POI	cost (mins)
car_driving	walking	car_parking	10010019527	10020010084	TRUE	100129964207	5

Apparently, this switching model is a rough approximation of the real parking action. The point feature in Figure 3.19 is an abstraction of all the nearby parking positions. In fact, the parking positions within this area are the exact points of switching from driving a car to walking, that is, the Switch Points. And these positions are highly dynamic as cars are arriving and leaving. Ideally, these parking positions as well as their static and dynamic attributes can be digitalized. If so, both the motorized and pedestrian networks would extend to these positions, and the Switch Points would be truly points. That would lead to exact multimodal route planning results.

Public station.

The public transit stations are stored in separated POI layers according to their modes in the raw dataset, which are Underground_stations, Suburban_stations and Tram_stations. Since the separation operation has been conducted to ensure that for one public transit service line heading one direction at a station, there is one specific vertex in the corresponding mode graph. For each station in one layer, if it has been separated into *n* distinctive vertices in the mode graph, 2*n* Switch Point tuples are created in which *n* tuples are from walking mode to this public transit mode and the rest n tuples are the other way around. In the raw dataset, every station node has a corresponding pedestrian node with exactly the same geographic position. This makes it possible to generate the platform vertices for each station. For every pedestrian vertex corresponding to the node with identical position to the station, m new pedestrian vertices are generated if there are m platforms at this station. Therefore, the relations between the n different public transit service lines and m platforms can be set up according to their correspondence in reality. For the Switch Point tuples switching from pedestrian to underground, the costs are time-dependent values t_S, while the costs of those n tuples switching reversely are 0. The availability and POI ID are recorded in a similar way to the parking lot example. Figure 3.20 and Figure 3.21 show the Switch Point tuples at the underground station Sendlinger Tor in Munich. The 16 created tuples are listed in Table 3.9.

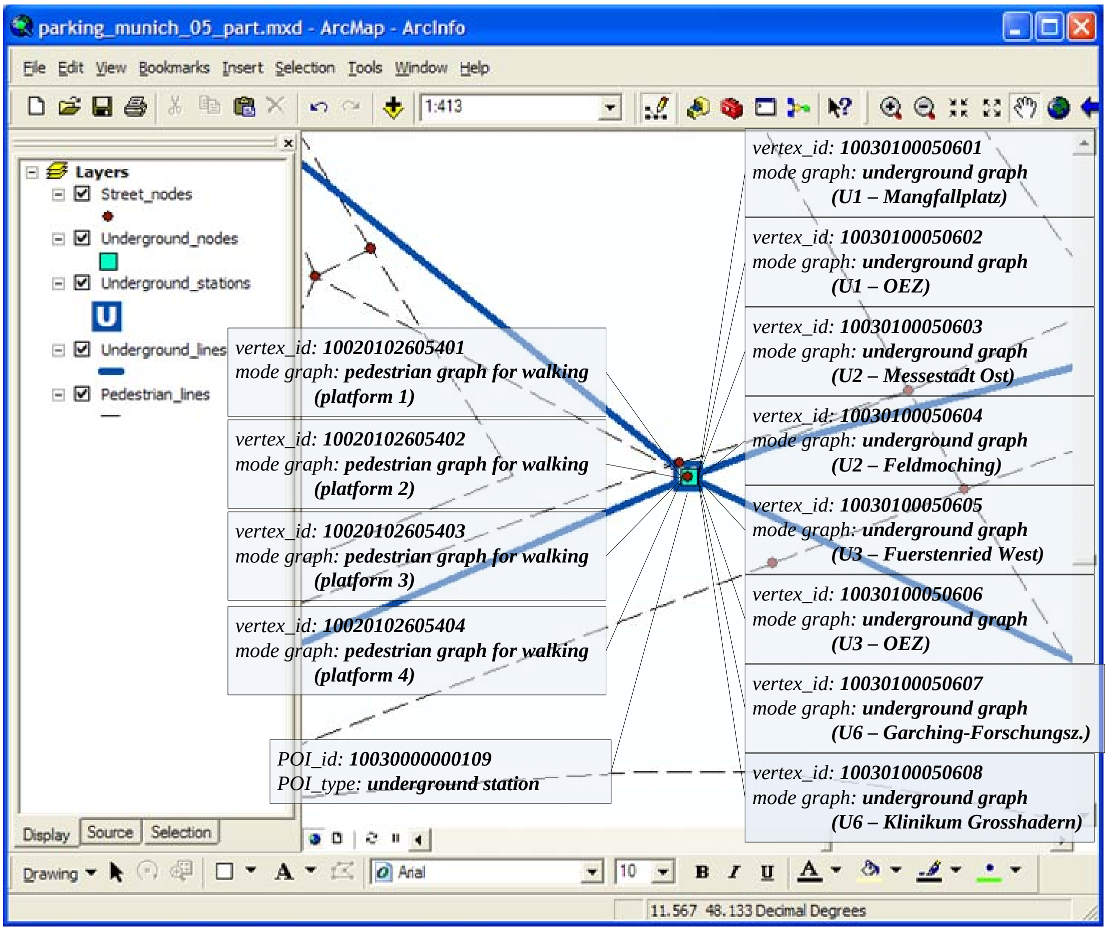


Figure 3.20 The underground network and pedestrian network near Sendlinger Tor underground station in Munich.

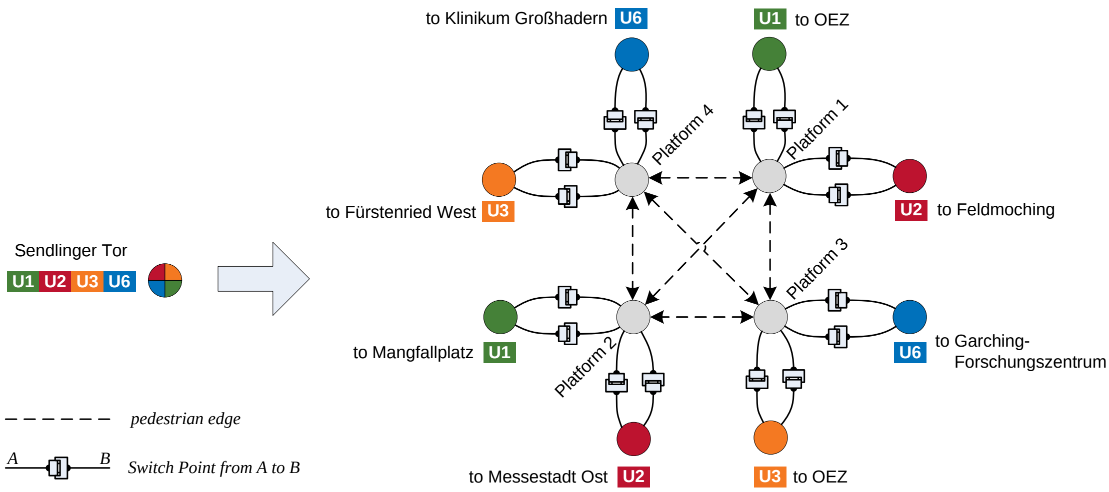


Figure 3.21 Switch Points at underground station Sendlinger Tor.

Table 3.9 Switch Point tuples constructed from the Sendlinger Tor underground station in Figure 3.20.

from_mode	to_mode	from_vertex	to_vertex	cost (mins)
walking	underground	10020102605401	10030100050602	t_S
walking	underground	10020102605401	10030100050604	t_S
walking	underground	10020102605402	10030100050601	t_S
walking	underground	10020102605402	10030100050603	t_S
walking	underground	10020102605403	10030100050606	t_S
walking	underground	10020102605403	10030100050607	t_S
walking	underground	10020102605404	10030100050605	t_S
walking	underground	10020102605404	10030100050608	t_S
underground	walking	10030100050602	10020102605401	0
underground	walking	10030100050604	10020102605401	0
underground	walking	10030100050601	10020102605402	0
underground	walking	10030100050603	10020102605402	0
underground	walking	10030100050606	10020102605403	0
underground	walking	10030100050607	10020102605403	0
underground	walking	10030100050605	10020102605404	0
underground	walking	10030100050608	10020102605404	0

The attributes type, is_available and ref_POI are omitted in the table because their values are “underground_station”, TRUE and 10030000000109 for all the 16 rows.

It should be noted that the platform information of underground stations in Munich is collected by the author manually because it is not included in the raw navigation database nor can be found from any public data source. If the platform nodes as well as the pedestrian paths connecting them exist in the raw dataset ideally, they do not have to be created virtually during the Switch Point construction process. Furthermore, the multimodal routing result would be more accurate if the entrances, exits, stairs, elevators and escalators of stations could be included in the raw database. The time of walking from one platform to another is non-negligible, and can be estimated as long as the distances between platforms are available.

3.4 Plug-and-Play multimodal graph set

With the support of the urban transportation network model and the concept of Switch Point, the overall data model of the multimodal transportation network can be completed. By giving necessary Switch Conditions, some mode graphs in the MMGS may form a subset containing the elements conditionally *pluggable* to each other. It should be noted that these input conditions are case-dependent, which means they may differ from one concrete multimodal routing task to another. According to the definitions of Switch Condition, Switch Relation and Switch Point, it is clear that a set of Switch Points can be retrieved with the given conditions. And every vertex pair within one Switch Point is just like a pair of plug and socket which allows *plug-and-play* of the related mode graphs.

3.4.1 Definition

For a mode graph $G_1 \in G_M$, V_1' is the set of *plug vertices* in G_1 if $V_1' = \{v | v \in \sigma_{m_f = m_1}(\Gamma).v_f\}$, and the set of *socket vertices* if $V_1' = \{v | v \in \sigma_{m_t = m_1}(\Gamma).v_f\}$ where σ is the selection operation in relational algebra. A graph with plug/socket vertices is illustrated in Figure 3.22. It should be noted that the analogy between plug/socket and the vertex pair within a Switch Point implies that the connection from a plug to a socket vertex is directional.

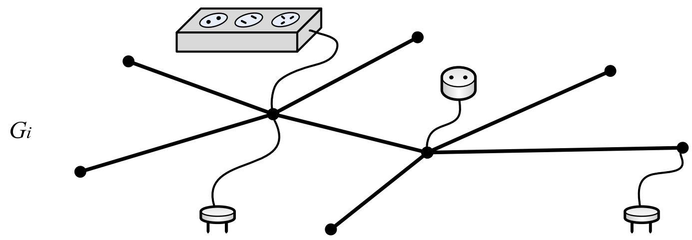


Figure 3.22 A mode graph with plugs and sockets.

For any ordered mode graph pair $(G_1, G_2) \in G_M \times G_M$, given a Switch Condition $\varphi_{1,2}$ whose subscript implies that it at least satisfies the criteria $(m_f = m_1) \wedge (m_t = m_2)$, the ordered mode graph pair (G_1, G_2) is said to be *one-way pluggable* iff $\Gamma'_{1,2} = \sigma_{\varphi_{1,2}}(\Gamma)$ is not $\varnothing$. Furthermore, (G_1, G_2) is *double-way pluggable* iff none of $\Gamma'_{1,2}$ and $\Gamma'_{2,1}$ is $\varnothing$. Figure 3.23 illustrated three pluggable graphs where (G_1, G_2) and (G_3, G_1) are one-way pluggable, while (G_2, G_3) is double-way pluggable.

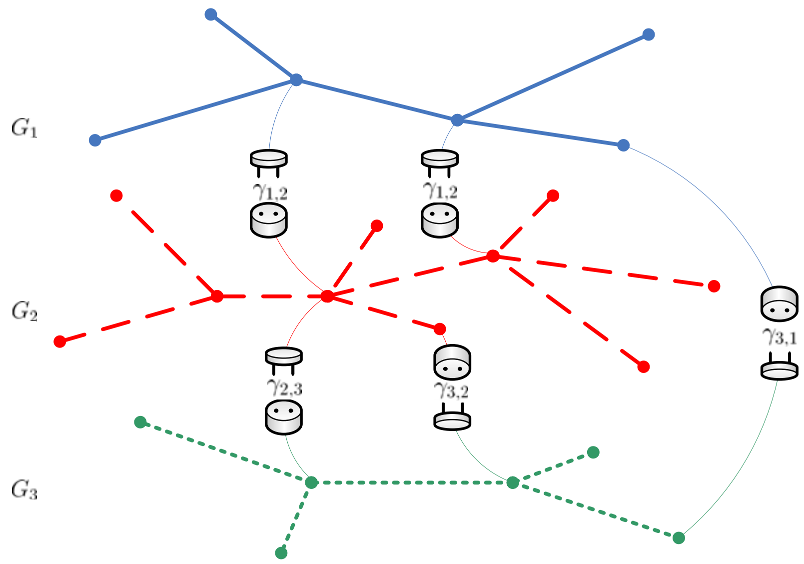


Figure 3.23 Example of one-way and double-way pluggable graphs.

If we take a second look at the examples in Section 3.3.4, it is obvious that the motorized and pedestrian graphs pair is one-way pluggable under the condition of Switch Type being parking lot for cars while the pedestrian and underground graphs pair are double-way pluggable under the condition of Switch Type being underground stations. With the support of the above definitions and discussions, the Plug-and-Play Multimodal Graph Set can be defined.

Definition 6 Plug-and-Play Multimodal Graph Set (PnPMMGS).
Given a Switch Relation Γ, *a MMGS* G_M *is a Plug-and-Play Multimodal Graph Set (PnPMMGS) if and only if* $\exists(G_1, G_2) \in G_M \times G_M$ *is one-way or double-way pluggable.*

3.4.2 Switch Condition Matrix and SCM-Plug operation

For a concrete multimodal path-finding job, let $M' \subseteq M$ be the set containing *n* modes which will be involved in the route calculation, and $G_{M'} \subseteq G_M$ correspondingly. If an ordered graph pair $(G_1, G_2) \in G_{M'} \times G_{M'}$ is one-way pluggable for a given $\varphi_{1,2}$, $\varphi_{1,2}$ is recorded. Otherwise, NIL is recorded if $\Gamma'_{1,2} = \varnothing$. All the $\varphi_{i,j}$ constitute Switch Condition Matrix (SCM, see Equation (6)). It should be noted that SCM depends on the concrete use case, while STM can be predefined by inferring from the domain-specific knowledge.

$$\mathrm{SCM} = [\varphi_{i,j}]_{n \times n} \tag{6}$$

The action of plugging two or more graphs together under a series of conditions expressed by SCM is called a SCM-Plug operation applying on $G_{M'}$, which is described by the Routine SCM-Plug. During a SCM-Plug operation, all the eligible Switch Points subjected to $\varphi_{i,j}$ are treated as new edges being added into the result graph. Note that the result of an SCM-Plug operation is a new graph not belonging to G_M.

SCM-Plug

Input: *M´*, SCM

Step 1: Initialize a new graph *G* consisting of an empty vertex set *V* and an empty edge set *E*, and its edge cost function is initialized to NIL;

Step 2: Assign to *V* and *E* with vertex/edge sets union: $V \leftarrow \bigcup_{i \in M'} V_i$; $E \leftarrow \bigcup_{i \in M'} E_i$; the cost function on *E* is inherited from those on its constituent edge subsets;

Step 3: Let $(i, j) \in M' \times M'$ be an ordered mode pair;

Step 4: Execute the selection operation on Switch Relation table Γ by giving Switch Condition $\varphi_{i,j}$ obtained from SCM, and get a result set of Switch Points Γ';

Step 5: For each Switch Point γ in Γ', construct an edge *e* by assigning the switch-from, switch-to vertices and switching cost values to *e* as its starting, ending vertices and edge cost, and append *e* to the edge set *E*;

Step 6: Repeat steps 4 and 5 for every mode pair $(i, j) \in M' \times M'$ if $\varphi_{i,j}$ in SCM is not NIL;

Step 7: Return *G* consisting of *V* and *E*, and its edge cost function.

The computational complexity of this routine is $O(n^2|\Gamma|)$ which means that the efficiency of SCM-PLUG operation is dependent on the number of Switch Points.

By applying SCM-PLUG operation when necessary, the connectivity “problem” raised in Section 3.2.2 can be smoothly solved. The method for modeling public transit network makes each public transit network a disconnected mode graph with every service line of one direction being a connected component. Taking the underground system in Munich as an example, there will be 12 connected components corresponding to the lines from U1 to U6 in both directions. However, if the underground and pedestrian graphs are plugged together under the condition of Switch Type being underground stations, the result graph will be connected totally. During the procedure, eligible Switch Points, which have time-dependent costs in this case, are appended as new edges with time-dependent cost functions.

To sum up, the multimodal network data model consists of two components: 1). the basic networks modeled as MMGS with static, time-independent edge cost functions; 2). Switch Points that can carry time-dependent information (e.g. cost, availability, etc.). For one multimodal route planning task, some of the static mode graphs can be conditionally plugged together by eligible Switch Points. The separation of static and dynamic information in transportation networks facilitates the optimal multimodal path-finding algorithms running on this model, which will be discussed in detail in the next chapter.

Chapter 4

Multimodal Routing Algorithms

This chapter deals with the algorithms for the planning of multimodal optimal paths. The design of algorithms is inseparable from the underlying data model. It begins with a formal description of the problem. Depending on whether the mode sequence is given in the input or not, there are two different types of routing tasks. Their corresponding solutions are introduced in Section 4.2. However, these solutions are not isolated from each other. On the contrary, they are related to each other and equivalent under some conditions. Furthermore, they can be applied in a hybrid fashion. The feasible mode combinations are determined by an inferring process based on case-dependent input and domain-specific knowledge. In the context of urban transportation, some rules are set for the decision process based on which the necessary input for multimodal shortest path algorithms can be obtained.

4.1 Formal description

In a nutshell, the multimodal urban transportation network is modeled as a set of weighted, directed time-independent graphs, i.e. MMGS, and a set of time-independent or time-dependent Switch Points. When there exist pluggable graph pair in the graph set, MMGS becomes PnPMMGS. The notational conventions used in this work are based on the third edition of *Introduction to Algorithms* (Cormen et al. 2009). In a multimodal shortest path (MMSP) problem, a mode set *M*, a non-negative weighted, digraph set $G_M = \{G_i = \{V_i, E_i\}\}$ denoting MMGS, a set of cost functions $C_M = \{c_i : E_i \to R^+\}$ and a Switch Relation Γ are the basic data structures ready to use. For a concrete multimodal route-planning task, the modes may either be manually input by the user or determined by the algorithm. Accordingly, the problems are formalized into two types of multimodal shortest path problem. Using multiple optimization criteria for route planning, which is called multi-criteria search, is not covered in this work. This thesis concentrates on single-criteria search algorithms.

4.1.1 Type I: Multimodal shortest path problem with deterministic mode sequence

In the first type of MMSP problem, besides *M*, G_M, C_M and Γ, for each concrete MMSP calculation task, two extra components are given: 1). a sequential list $\boldsymbol{M'} = (m_1, m_2, \ldots, m_n)$, $n \geq 1$, $m_k \in M$, indicating the modes to be involved in the path-finding process; 2). a Switch Condition list $\Phi = (\varphi_1, \varphi_2, \ldots, \varphi_{n-1})$ where φ_k corresponds to the Switch Condition from m_k to m_{k+1}. $\sigma_\varphi(\Gamma)$ is a function working on Γ and can select the Switch Point tuples subjected to a condition φ. In other words, it is assumed that the mode combination is given as a deterministic sequential list beforehand.

The cost of a path $p_k = (v_0, v_1, \ldots, v_j)$ within mode m_k is the sum of the cost values of its constituent edges in the mode graph of m_k (Equation (7)).

$$c(p_k) = \sum_{i=1}^{j} c_k(v_{i-1}, v_i) \tag{7}$$

The multimodal shortest path cost from $u \in V_1$ to $v \in V_n$ with $\boldsymbol{M'}$ is defined by:

$$\delta(u, v, \boldsymbol{M'}, \Phi) = min\left\{\sum_{k=1}^{n} c(p_k) : u \overset{p_1}{\rightsquigarrow} \gamma_{1,2} \overset{p_2}{\rightsquigarrow} \cdots \overset{p_{n-1}}{\rightsquigarrow} \gamma_{n-1,n} \overset{p_n}{\rightsquigarrow} v \mid \gamma_{i,i+1} \in \Gamma'_{i,i+1}\right\} \tag{8}$$

where $\Gamma'_{i,i+1} = \sigma_{\varphi_{i,i+1}}(\Gamma)$ if there exists an *n*-modal path from *u* to *v*. Thus, an *n*-modal shortest path from vertex $u \in V_1$ to vertex $v \in V_n$ is defined as any path *p* with the cost $c(p) = \delta(u, v, \boldsymbol{M'}, \Phi)$. The MMSP problem includes different variants, depending on whether a target vertex *t* is specified in the input. Two of the variants, which are *single-source MMSP problem with deterministic mode sequence* or *Type I single-source MMSP problem* (SSMMSP-I) and *point-to-point MMSP problem with deterministic mode sequence* or *Type I point-to-point MMSP problem* (P2PMMSP-I), are discussed in this work. Formally, they are defined as follows.

Definition 7 SSMMSP-I.
Given M, G_M, C_M, Γ, $\boldsymbol{M'}$, Φ and $s \in V_1$, we ask for a path p with $c(p) = \delta(s, v, \boldsymbol{M'}, \Phi)$ for each $v \in V_n$.

Definition 8 P2PMMSP-I.
Given M, G_M, C_M, Γ, $\boldsymbol{M'}$, Φ, $s \in V_1$ and $t \in V_n$, we ask for a path p with $c(p) = \delta(s, t, \boldsymbol{M'}, \Phi)$.

Besides, both problems can have a time-dependent version if a departure time $\tau_o < \Pi$ is given in the input.

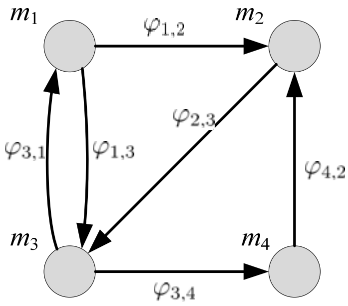
4.1.2 Type II: Multimodal shortest path problem with nondeterministic mode sequence

Instead of having the mode sequence as a part of the input, the optimal mode combination is to be automatically generated in this second type of the problem. Compared with Type I problem, the basic data structures of *M*, G_M, C_M and Γ are the same. Besides, the case-dependent part of input contains a subset $M' \subseteq M$ including *n* modes and a Switch Condition matrix (SCM).

This second type of MMSP problem is defined by introducing an auxiliary graph $\mathcal{G}$ whose adjacency matrix is expressed by the SCM. $\mathcal{G}$ is a non-weighted digraph. The vertex set of $\mathcal{G}$ consists of the *n* modes; the non-NIL element at (i, j) position denotes there is a directed edge from m_i to m_j. Figure 4.1 illustrate an example of SCM and its corresponding $\mathcal{G}$.

$$\begin{array}{c} \\ m_1 \\ m_2 \\ m_3 \\ m_4 \end{array} \begin{array}{c} \begin{array}{cccc} m_1 & m_2 & m_3 & m_4 \end{array} \\ \begin{bmatrix} \text{NIL} & \varphi_{1,2} & \varphi_{1,3} & \text{NIL} \\ \text{NIL} & \text{NIL} & \varphi_{2,3} & \text{NIL} \\ \varphi_{3,1} & \text{NIL} & \text{NIL} & \varphi_{3,4} \\ \text{NIL} & \varphi_{4,2} & \text{NIL} & \text{NIL} \end{bmatrix}_{4\times 4} \end{array}$$

(*a*)



(*b*)

Figure 4.1 An example of Switch Condition Matrix (SCM) and the expressed graph.

$\mathfrak{p}(i, j)$ denotes all paths from m_i to m_j in $\mathcal{G}$. It is a path set where each element is a mode sequence $\boldsymbol{M}'$ if expressed by vertices, and a Switch Condition list Φ if expressed by edges corresponding to a theoretically feasible mode sequence $\boldsymbol{M}'$. For a deterministic $\boldsymbol{M}'$ and a pair of vertices belonging to the first and last mode graphs respectively, the multimodal shortest path cost has already been defined by Equations (7) and (8). Consequently, with the support of these auxiliary concepts, the multimodal shortest path cost from $u \in V_1$ to $v \in V_n$ with SCM can be defined by:

$$\delta(u, v, M', \mathrm{SCM}) = min\,\{\delta(u, v, \boldsymbol{M}', \Phi) \mid \boldsymbol{M}', \Phi \in \mathfrak{p}(1, n)\} \tag{9}$$

The MMSP in this case from vertex $u \in V_1$ to vertex $v \in V_n$ is defined as any path *p* with cost $c(p) = \delta(u, v, M', \mathrm{SCM})$. Similar to the first type of MMSP problem, Type II MMSP problem has also two variants Definition 9 and Definition 10 – *single-source MMSP problem with non-deterministic mode sequence* or *Type II single-source MMSP problem* (SSMMSP-II) and *point-to-point MMSP problem with non-deterministic mode sequence* or *Type II point-to-point MMSP problem* (P2PMMSP-II) – depending on whether the target vertex is provided in the input.

Definition 9 SSMMSP-II.
Given M, G_M, C_M, Γ, M', SCM, $s \in V_1$, *we ask for a path p with* $c(p) = \delta(s, v, M', \mathrm{SCM})$ *for each* $v \in V_n$.

Definition 10 P2PMMSP-II.
Given M, G_M, C_M, Γ, M', SCM, $s \in V_1$ *and* $t \in V_n$, *we ask for a path p with* $c(p) = \delta(s, t, M', \mathrm{SCM})$.

Similar to the definitions of Type I problem, the time-dependent versions of the above two problems can be defined by providing a departure time $\tau_o < \Pi$ in the input. The auxiliary graph $\mathcal{G}$ is introduced here to facilitate the comprehension of Type II problem.

4.2 Multimodal shortest path algorithms

In our previous work (Liu and Meng 2009, 2010), the time-independent case of the Type I problem was studied and the solutions based on both label-setting and label-correcting methods were proposed. The time-dependency issue, however, is not involved. In this work, the concept of Switch Point is extended to include the dynamic aspect during the modelling process and both Type I and II of MMSP problem have the departure time τ_o in their input. Consequently, the MMSP algorithmic framework based on the original Switch Point is extended.

4.2.1 Algorithm for Type I problem

For the SSMMSP-I with *M*, G_M, C_M, Γ as basic data components and $\boldsymbol{M'}$, Φ, *s* as the case-dependent input, multimodal shortest path algorithms (MMSPA) can be generally described by the algorithmic framework (Algorithm I).

Algorithm I

Input: $\boldsymbol{M'}$, Φ, *s*

Step 1: Set $k \leftarrow 1$;

Step 2: Set *Q*← ∅; Execute MULTIMODALINITIALIZE(*k*, NIL, *s*, *Q*) when *k* = 1, or MULTIMODALINITIALIZE(*k*, φ_{k-1}, *s*, *Q*) when *k* > 1; ITERATIVERELAX(*k*, *Q*); $k \leftarrow k+1$;

Step 3: Repeat step 2 until $k > n$.

where the routine MULTIMODALINITIALIZE works as following:

MULTIMODALINITIALIZE

Input: *k*, φ, *s*, *Q*

Step 1: Begin by setting $distance[k][u] \leftarrow \infty$ and $predecessor[k][u] \leftarrow \text{NIL}$ for each vertex $u \in V_k$;

Step 2: If φ is NIL, set $distance[k][s] \leftarrow 0$, $predecessor[k][s] \leftarrow s$ and add *s* to *Q*; otherwise go to Step 3;

Step 3: If $distance[k][\gamma.v_t] > distance[k-1][\gamma.v_f] + \gamma.c$, set $distance[k][\gamma.v_t] \leftarrow distance[k-1][\gamma.v_f] + \gamma.c$, $predecessor[k][\gamma.v_t] \leftarrow \gamma.v_f$ and add $\gamma.v_t$ to *Q* for each tuple $\gamma \in \sigma_\varphi(\Gamma)$.

The ITERATIVERELAX routine iteratively relaxes the edges in G_k, and can be substituted by any existing SSSP algorithm based on labeling method with the initialization phase removed. A generic ITERATIVERELAX routine can be described as a three-step process as following:

ITERATIVERELAX

Input: *k*, *Q*

Step 1: Remove a vertex *u* from the candidate list *Q*;

Step 2: If $distance[k][v] > distance[k][u] + c_k(u,v)$, set $distance[k][v] \leftarrow distance[k][u] + c_k(u,v)$ and $predecessor[k][v] \leftarrow u$, add *v* to *Q* if $v \notin Q$ for each outgoing edge (*u*, *v*);

Step 3: If *Q*≠ ∅, then go to step 1.

Both label-setting and label-correcting methods can be applied into the algorithmic framework illustrated in Algorithm I. Their fundamental difference lies in the order according to which the distance labels are updated. Label-setting algorithms set one label as permanent in each step, while label-correcting algorithms consider all labels as temporary until the final step, when they all become permanent. Algorithm I reveals a computational complexity of $O(\sum_{k=1}^{n} g(|V_k|,|E_k|))$ where $O(g(|V_k|,|E_k|))$ is the computational complexity of ITERATIVERELAX.

Basically, Algorithm I executes the shortest path finding on the corresponding mode graph sequentially. The key action is the relay of the distance values on Switch Points during the initialization step in each round of path searching from the mode m_2 to m_n through which the MMSP can be finally found.

If the SSMMSP-I is time-dependent, i.e. $\tau_o < \Pi$ is provided as a part of the input, the algorithms need some minor modifications to consider the departure time of the trip. Specifically, the Step 2 in Algorithm I should be substituted by:

Step 2: Set $Q \leftarrow \varnothing$; Execute MULTIMODALINITIALIZE(k, NIL, s, τ_o, Q) when k = 1, or MULTIMODALINITIALIZE(k, φ_{k-1}, s, τ_o, Q) when k > 1; ITERATIVERELAX(k, τ_o, Q); $k \leftarrow k+1$;

Step 3 in MULTIMODALINITIALIZE becomes:

Step 3: If $distance[k][\gamma.v_t] > distance[k-1][\gamma.v_f] + \gamma.c + \tau_o$, set $distance[k][\gamma.v_t] \leftarrow distance[k-1][\gamma.v_f] + \gamma.c + \tau_o$, $predecessor[k][\gamma.v_t] \leftarrow \gamma.v_f$ and add $\gamma.v_t$ to Q for each tuple $\gamma \in \sigma_\varphi(\Gamma)$.

And Step 2 in ITERATIVERELAX is modified as:

Step 2: If $distance[k][v] > distance[k][u] + c_k(u,v) + \tau_o$, set $distance[k][v] \leftarrow distance[k][u] + c_k(u,v) + \tau_o$ and $predecessor[k][v] \leftarrow u$, add v to Q if $v \notin Q$ for each outgoing edge (u, v);

It should be noted that the $distance[k][u]$ label in the modified steps indicates the arrival time at vertex u in graph G_k.

For the case that the target vertex t is provided, i.e. P2PMMSP-I, the solution is almost identical to that illustrated in Algorithm I with the only change, which is different for applying label-correcting and label-setting method, as follows:

- **Algorithm I based on label-setting methods** (e.g. Dijkstra's algorithm). An additional stop condition is added in step 1 of ITERATIVERELAX for the last mode graph, i.e. when k = n: if u = t, the algorithm terminates.
- **Algorithm I based on label-correcting methods** (e.g. Bellman-Ford algorithm). The modification is also applied in the last mode graph, but a little bit more complex because of the characteristics of label-correcting methods. When a vertex is first removed from Q in Step 1 of ITERATIVERELAX, its attached labels may not represent the multimodal shortest path from s to the current vertex. The labels may be improved and the vertex may reenter Q. As a consequence, the algorithm cannot be terminated when t has been removed from Q at the first time as in the label-setting case. Instead, a sharper stop condition for ITERATIVERELAX can be set when k = n: if u = t, $distance[n][u]$ is an upper bound for the multimodal shortest path to t, denoted by $\bar{\delta}$. After that, only the vertices in Q meet $distance[n][\cdot] \leq \bar{\delta}$ will be considered, and the process proceeds until Q is empty.

Inject Switch Constraints into the routing process. As introduced in Section 3.3.3, there may be constraints on Switch Points. Unlike the input Switch Conditions that act as a set of pre-filters for selecting feasible Switch Points to be involved in the path-finding process, the constraints can be regarded as some on-the-fly filters acting on the varying label values of vertices to determine whether a switching action is valid or not.

Switch Constraints are expressed in a sequential list $\mathfrak{F} = (\mathfrak{f}_1, \mathfrak{f}_2, \ldots, \mathfrak{f}_{n-1})$, similar to the Switch Condition list Φ. And every constraint element in the list is expressed by a callback function $\mathfrak{f}$ with a vertex type parameter and a Boolean type return value. The callback function allows the constraints to be set at the client side and injected into the path-finding process. From the object-oriented point of view, this is consistent with the Inversion of Control (Fowler 2004) principle of designing a framework. This elegant design minimizes the impact on the algorithm and routines. In particular, the changes solely happen in Step 3 of MULTIMODALINITIALIZE with the additional input of $\mathfrak{F}$: when traverse the feasible Switch Points $\gamma \in \sigma_{\varphi}(\Gamma)$, check the value of $\mathfrak{f}(\gamma.v_f)$ and ignore this Switch Point if $\mathfrak{f}(\gamma.v_f) = \text{FALSE}$.

4.2.2 Algorithm for Type II problem

The SCM-PLUG operation is adopted to support the solution. Since the result of the operation is a mono-modal graph, any traditional algorithm for SSSP can work well on this graph. Compared with Type I problem, the basic data structures of *M*, G_M, C_M and Γ are the same. Besides, the case-dependent part of input contains a subset $M' \subseteq M$ including *n* modes, a matrix SCM and a source *s*. With M', SCM, *s* as the input, the MMSPA working on a PnPMMGS is described in.

Algorithm II

Input: M', SCM, *s*

Step 1: $G, c \leftarrow \text{SCM-PLUG}(M', \text{SCM})$;

Step 2: Execute SHORTESTPATHSEARCH on *G* with *s*.

where SHORTESTPATHSEARCH routine denotes any SSSP algorithm.

When the target vertex *t* is provided, the problem becomes P2PMMSP-II. Similar to the solution to P2PMMSP-I, an additional stop condition can be applied to terminate the SHORTESTPATHSEARCH earlier. Specifically, the strategies of setting the stop conditions for label-setting and label-correcting methods are the same as that in the last section. Furthermore, the heuristic SPA following a goal-directed strategy can be applied on the merged mono-modal graph if a target vertex is specified in the input. This is one of the classical speed-up techniques for SPA. The heuristic algorithm consists of two steps as well:

Step 1: $G, c \leftarrow \text{SCM-PLUG}(M', \text{SCM})$;

Step 2: Execute A* shortest path search on *G* with *s*, *t* and the heuristic function $\hat{h}(v)$.

where A* (Hart et al. 1968) is the classical heuristic SPA. The selection of $\hat{h}(v)$ depends on the metric of the cost function in use when executing A*. As to the shortest geographical distance, the direct geographic distance (beelining) is a well-known and feasible selection for $\hat{h}(v)$. However, it is not applicable when calculating the fastest route (Pajor 2009). A travelling time function dividing the direct geographic distance by the highest speed of all the traffic tools may be a reasonable substitute.

The time-dependent version of Type II problem with a known departure time $\tau_o < \Pi$ can be reduced to an *earliest arrival problem* (Geraets et al. 2007) in the time-dependent model where the edges from Switch Points have time-dependent cost functions. An earliest arrival problem requires its underlying time-dependent model with the FIFO *(First in, First out)* property to guarantee that along an edge it is impossible to depart later but arrive earlier. Without the FIFO property, the earliest arrival problem becomes NP-hard to solve (Ariel and Raphael 1990). In the data model proposed in this work, only the edges generated from Switch Points during the SCM-PLUG operation will have time-dependent cost functions. And the FIFO property can be fulfilled in this model because for any two people arriving at a station one after another to get on the same service line with the same direction, the departure time of the later arriving person will never be earlier than that of the other.

4.2.3 Algorithmic comparison

The Switch Condition list Φ for Algorithm II is a special case of SCM, which can be expressed SCM_Φ in Equation (10). In other words, the Plug-and-Play multimodal shortest-path problem can be solved by applying SCM-PLUG operation first and then a traditional SPA on the mono-modal graph. The routing results by the two approaches are exactly the same as declared by Theorem 1.

$$\mathrm{SCM}_\Phi = \begin{array}{c c} & \begin{array}{cccccc} m_1 & m_2 & m_3 & \cdots & m_{n-1} & m_n \end{array} \\ \begin{array}{c} m_1 \\ m_2 \\ \vdots \\ m_{n-1} \\ m_n \end{array} & \begin{bmatrix} \mathrm{NIL} & \varphi_1 & \mathrm{NIL} & \cdots & \mathrm{NIL} & \mathrm{NIL} \\ \mathrm{NIL} & \mathrm{NIL} & \varphi_2 & \cdots & \mathrm{NIL} & \mathrm{NIL} \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \mathrm{NIL} & \mathrm{NIL} & \mathrm{NIL} & \cdots & \mathrm{NIL} & \varphi_{n-1} \\ \mathrm{NIL} & \mathrm{NIL} & \mathrm{NIL} & \cdots & \mathrm{NIL} & \mathrm{NIL} \end{bmatrix}_{n \times n} \end{array} \tag{10}$$

Before introducing the theorem of the equivalence relation between the two methods, the effectiveness of Algorithm I can be proved first with the help of the two theorems in (Liu and Meng 2009).

Lemma 1 Effectiveness of the Algorithm I.
Given M, G_M, C_M, Γ, $\boldsymbol{M'}$, Φ and $s \in V_1$ as the input of Algorithm I, we can have the shortest distance values from s on all the vertices $v \in V_{\boldsymbol{M'}} = \bigcup V_k, k \in \boldsymbol{M'}$ after the algorithm terminates.

***Proof*:**
The general algorithmic framework can be categorized into two classes according to the taxonomy of SPA based on the labeling method: label-setting and label-correcting algorithms. For both categories, the proposition holds according to the Theorem 3-1 and 3-2 in (Liu and Meng 2009) respectively. ■

Theorem 1 Equivalence of Algorithm I and Algorithm II.
Given M, G_M, C_M, Γ, $\boldsymbol{M'}$ and its corresponding set M', Φ and its matrix expression SCM_Φ, $s \in V_1$, we have the following equivalence relation:

$$\text{Algorithm I}(\boldsymbol{M'}, \Phi, s) \iff \text{Algorithm II}(M', \mathrm{SCM}_\Phi, s).$$

And after both algorithms terminate, we have the same shortest distance values from *s* on all the vertices $v \in V_{\boldsymbol{M'}} = \bigcup V_k, k \in \boldsymbol{M'}$.

Proof:
Let *G* and *c* be obtained by $\mathrm{SCM\text{-}P{\small LUG}}(M', \mathrm{SCM})$. In fact, Algorithm I can be rewritten into a non-concise, equivalent form as following:

Step 1: Begin by setting $k \leftarrow 1$;

Step 2: Set $distance[k][u] \leftarrow \infty$ and $predecessor[k][u] \leftarrow \mathrm{NIL}$ for each vertex $u \in V_k$; $k \leftarrow k+1$;

Step 3: Repeat step 2 until $k > n$;

Step 4: Set $distance[1][s] \leftarrow 0$ and $predecessor[1][s] \leftarrow s$;

Step 5: Execute ITERATIVEEDGERELAX(1);

Step 6: Set $k \leftarrow 1$;

Step 7: If $distance[k][\gamma.v_t] > distance[k-1][\gamma.v_f] + \gamma.c$, set $distance[k][\gamma.v_t] \leftarrow distance[k-1][\gamma.v_f]$ and $predecessor[k][\gamma.v_t] \leftarrow \gamma.v_f$ for each tuple $\gamma \in \sigma_{\varphi_k}(\Gamma)$;

Step 8: Execute ITERATIVEEDGERELAX(*k*+1); $k \leftarrow k+1$;

Step 9: Repeat step 7 and 8 until $k > n-1$.

where the work done in Step 1 to 4 is equivalent to the initialization phase on *G*. And the edge relaxation phase on *G* is completely done in Step 5 to 9 where Step 7 relaxes the one-way edges created by the eligible Switch Points. Therefore, it does exactly the same work as an SSSP algorithm based on labeling method, which is Algorithm II.

It is already known by Lemma 1 that the shortest distance values can be obtained by executing Algorithm I. And Algorithm II can yield the same results due to the equivalence relation between them. ■

The advantage of Algorithm II lies in that the final optimal mode sequence can be given by the algorithm rather than input manually. This characteristic makes it suitable in the situation where a definite mode sequence is difficult to determine in advance, e.g. the route planning task in a public transportation network consisting of underground, suburban, tram and bus lines altogether or some of them. In practice, Algorithm I and Algorithm II can be used in a hybrid fashion by treating the graph obtained by SCM-PLUG operation as one of the mono-modal graphs in the sequential list of input for Algorithm I. In the prototype system as shown in the next chapter, the problem of route planning for the combination of car-driving and public transportation is solved by applying SCM-PLUG operation on all the selected public transit networks and pedestrian network first, then constructing a (*car-driving*, *public transportation*) mode list as well as feasible Switch Conditions and delivering them to Algorithm I.

4.3 Multimodal routing plan inferring

The algorithmic framework proposed in the last section is generic in nature, which means that the application of the approach is not restricted to transportation field. As it can be seen in the input of the algorithms, the modes which will be involved in the path calculation have to be given in a sequential list. However, users would rather provide the system with their preferences or some options than the final optimal mode sequence. Therefore, a gap may occur between what the routing engine needs and what users can give as illustrated in Figure 4.2,

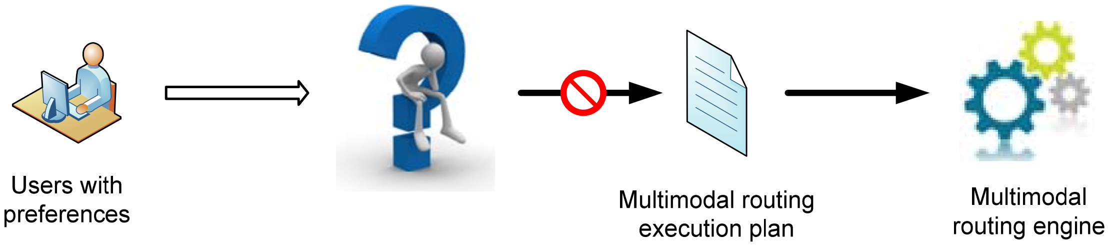


Figure 4.2 A mismatch between users' input and what the multimodal routing engine needs.

To acquire the execution plan of multimodal route planning, it is necessary to have an inferring process based on users' preferences and the knowledge from specific domain (Figure 4.3). The main task of the inferring component is to generate a group of case-dependent inputs including mode sequence $\boldsymbol{M}'$, Switch Condition list Φ and Switch Constraint list $\mathfrak{F}$ if they exist for the multimodal route planning algorithms. The generated inputs must be consistent with the commonsense and rules from the specific application domain.

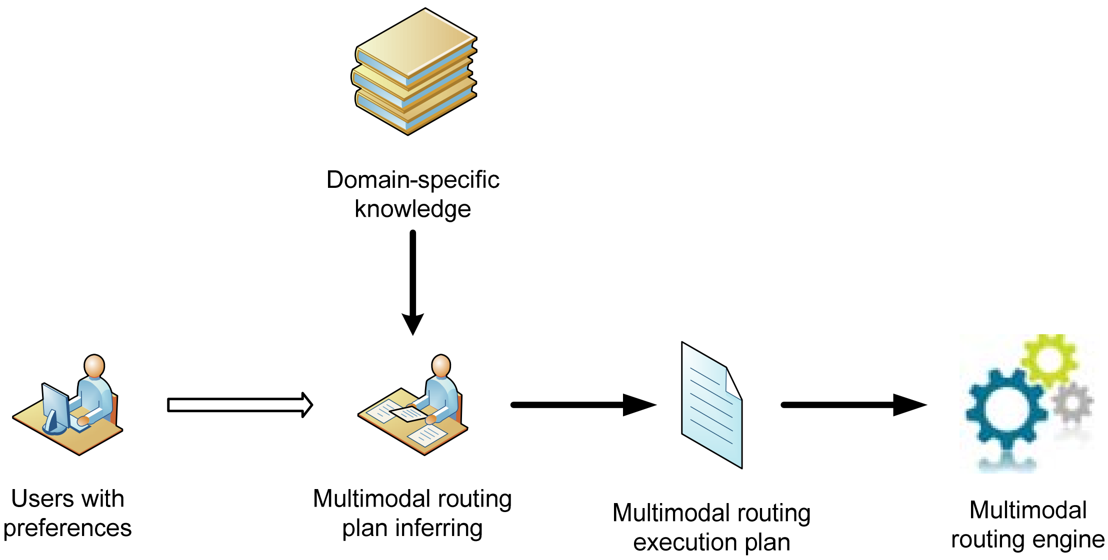


Figure 4.3 Elimination of the mismatch problem by inferring the routing plan based on domain-specific knowledge.

If the discussion is restricted in urban transportation field, the shortest path with multiple travel modes reveals some properties which can help infer the feasible mode combinations.

- **The number of switch points in the MMSP is limited.** There is a set of alternative transportation modes available for users to make their travel plan. Although people may travel using more than one mode, they are not disposed to undertake too many mode-switching actions. This property can be seen as a constraint on the number of modal transfers on a MMSP.

- **The sequence of switch points in the MMSP has some regularity.** The regularity of the switch points sequence can be set equal to the transportation mode sequence. A multimodal path may become less logical if the involved transportation mode combination seldom or never appears in a travel plan. This property can be seen as a constraint on the mode list.

By taking the constraints into account the path will become “viable” as defined in (Lozano and Storchi 2001). However, finding the MMSP and determining the relative logic of mode combination are treated as two separate problems in this work. The attempt to couple the algorithms with the concrete mode constraints may limit the acceptance level of routing algorithms in applications other than transportation route planning, e.g. computer network routing, circuit designing, social network routing, etc.

In transportation field specifically addressed in this work, the inferring process is implemented by applying some rules of the combination of transportation modes and providing some pre-routing options for users. The transportation modes and Switch Types involved in the inferring process are listed in Table 4.1 and Table 4.2 respectively.

Table 4.1 Transportation modes.

Transportation mode	Abbreviation
private car driving	*D*
walking	*W*
by underground train	*U*
by suburban train	*S*
by tram train	*T*

Table 4.2 Switch Types in urban transportation system.

<table>
<tr><th>Type</th><th>Abbreviation</th><th>from_mode</th><th>to_mode</th></tr>
<tr><td>parking lot for cars</td><td>P</td><td>D</td><td>W</td></tr>
<tr><td>physical connection</td><td>phy-conn</td><td>D</td><td>W</td></tr>
<tr><td>charging station for cars</td><td>charging</td><td>D</td><td>W</td></tr>
<tr><td rowspan="2">park and ride</td><td rowspan="2">P+R</td><td>D</td><td>U</td></tr>
<tr><td>D</td><td>S</td></tr>
<tr><td rowspan="3">kiss and ride</td><td rowspan="3">Kiss+R</td><td>D</td><td>U</td></tr>
<tr><td>D</td><td>S</td></tr>
<tr><td>D</td><td>T</td></tr>
<tr><td rowspan="2">underground station</td><td rowspan="2">U-station</td><td>W</td><td>U</td></tr>
<tr><td>U</td><td>W</td></tr>
<tr><td rowspan="2">suburban station</td><td rowspan="2">S-station</td><td>W</td><td>S</td></tr>
<tr><td>S</td><td>W</td></tr>
<tr><td rowspan="2">tram station</td><td rowspan="2">T-station</td><td>W</td><td>T</td></tr>
<tr><td>T</td><td>W</td></tr>
</table>

A group of options (Table 4.3) is provided for determining feasible execution plans of optimal path finding in urban multimodal transportation networks. In Table 4.4, some rules guiding the inferring work are listed.

Table 4.3 Routing options.

Routing option	Value	Abbreviation
private car available on departure	T/F	q_1
need parking for the car	T/F	q_2
can use underground lines	T/F	q_3
can use suburban lines	T/F	q_4
can use tram lines	T/F	q_5
mode graph(s) that source vertex belongs to	a subset of M	`M(s)`
mode graph(s) that target vertex belongs to	a subset of M	`M(t)`
driving distance limit	R^+	`L(D)`

Table 4.4 Rules on transportation modes combination and modes switching.

No.	Rule	Implication
1	If q_1=F, then q_2=F	A parking lot is not needed if the user does not have a car available on departure.
2	If q_1=F, then mode *D* will not be involved	The private car driving will not be considered if the user does not have a car available on departure.
3	If either M(s) or M(t) contains *W*, then mode *W* will be involved	Walking mode will be considered if the route starts or ends in a walkable area.
4	If M(s) does not contain *D*, then mode *D* will not be involved	The private car driving will not be considered if the user's origin spot is in a non-motor area.
5	If q_1=T and M(s) contains *D*, then mode *D* will not be involved	The private car driving will be considered if the user has a car available on departure and the origin is in a motor area.
6	If mode *D* is involved, then it will be the first mode and appear only once in the mode list	The private car driving, if available, will only appear as the first mode and not be used any more during the trip.
7	If q_3=T, then mode *U* and *W* will be involved, Switch Type $\lambda_{U,W}$=U-station and $\lambda_{W,U}$=U-station will be considered	If the user can use underground transport system, both underground train taking and walking will be considered. Underground train stations will be feasible Switch Points between these two modes.
8	If q_4=T, then mode *S* and *W* will be involved, Switch Type $\lambda_{S,W}$=S-station and $\lambda_{W,S}$=S-station will be considered	If the user can use suburban transport system, both suburban train taking and walking will be considered. Suburban train stations will be feasible Switch Points between these two modes.
9	If q_5=T, then mode *T* and *W* will be involved, Switch Type $\lambda_{T,W}$=T-station and $\lambda_{W,T}$=T-station will be considered	If the user can use tram transport system, both tram train taking and walking will be considered. Tram train stations will be feasible Switch Points between these two modes.
10	If mode *D* and any of the modes *U*, *S* and *T* are involved simultaneously and q_2=T, then Switch Type $\lambda_{D,U}$=P+R and $\lambda_{D,S}$=P+R will be considered	If the private car driving is available on departure, and the user can use public transit system (any of the three), and a parking lot is necessary, then Park and Ride lots will be feasible Switch Points from the mode of car driving mode to public transit taking.

11	If mode *D* and any of the modes *U*, *S* and *T* are involved simultaneously and q_2=F, then Switch Type $\lambda_{D,U}$=Kiss+R, $\lambda_{D,S}$=Kiss+R and $\lambda_{D,T}$=Kiss+R will be considered	If the private car driving is available on departure, and the user can use public transit system (any of the three), and a parking lot is unnecessary, then Kiss and Ride lots will be feasible Switch Points from the mode of car driving mode to public transit taking.
12	Among modes *U*, *S* and *T*, if more than one of them are involved, the involved mode graphs as well as the mode graph of *W* should be plugged together	The involved public transit graphs and walking graph should be combined as a whole by applying SCM-PLUG operation.
13	If q_1=T and q_2=T, then Switch Type $\lambda_{D,W}$=P will be considered	A parking lot is necessary when switching from car driving to walking.
14	If q_1=T and q_2=F, then Switch Type $\lambda_{D,W}$=phy-conn will be considered	The user is a passenger and someone else drives the car. A temporary parking place will be the Switch Point from car driving to walking.
15	If q_1=T, q_2=T and L(D)>0, then Switch Type $\lambda_{D,W}$=P and $\lambda_{D,W}$=charging, Switch Constraint L(D) will be considered	A parking lot or a charging station will be feasible Switch Points from car driving to walking.

By asking for a group of routing options, a set of multimodal route planning execution plans can be provided based on some specific rules. In the context of multimodal urban transportation, the possible combinations of different routing options can be so diverse that an exhausted list is hardly possible. Some typical input routing options and the inferred routing plans are demonstrated in Table 4.5.

Table 4.5 Typical routing options and their corresponding routing plans.

Routing option								**Routing plan**		
q_1	q_2	q_3	q_4	q_5	M(s)	M(t)	L(D)	**modes**	**Switch Conditions**	**Switch Constraints**
F	F	F	F	F	$\{D,W\}$	W	N/A	(W)	N/A	N/A
F	F	T	F	T	$\{D,W\}$	W		(W)	N/A	
								$\{W,U,T\}$	SCM=($\varphi_{W,U}=$ ($m_f=W$ AND $m_t=U$ AND $\lambda=$U-station); $\varphi_{W,T}=$($m_f=W$ AND $m_t=T$ AND $\lambda=$T-station); $\varphi_{U,W}=$($m_f=U$ AND $m_t=W$ AND $\lambda=$U-station); $\varphi_{T,W}=$($m_f=T$ AND $m_t=W$ AND $\lambda=$T-station))	
T	T	F	F	F	$\{D,W\}$	$\{D,W\}$		(D)	N/A	
								(W)	N/A	
								(D,W)	Φ=($m_f=D$ AND $m_t=W$ AND $\lambda=$P)	
T	F	T	T	T	$\{D,W\}$	W		(W)	N/A	
								$\{W,U,S,T\}$	SCM=($\varphi_{W,U}=$ ($m_f=W$ AND $m_t=U$ AND $\lambda=$U-station); $\varphi_{W,S}=$($m_f=W$ AND $m_t=S$ AND $\lambda=$S-station); $\varphi_{W,T}=$($m_f=W$ AND $m_t=T$ AND $\lambda=$T-station); $\varphi_{U,W}=$($m_f=U$ AND $m_t=W$ AND $\lambda=$U-station); $\varphi_{S,W}=$($m_f=S$ AND $m_t=W$ AND $\lambda=$S-station); $\varphi_{T,W}=$($m_f=T$ AND $m_t=W$ AND $\lambda=$T-station))	
								$(D,\{W,U,S,T\})$	Φ=($m_f=D$ AND $m_t=W$ AND $\lambda=$Kiss+R); SCM: the same as that in the above $\{W,U,S,T\}$ case.	
T	T	T	T	T	$\{D,W\}$	W	8 km	(W)	N/A	N/A

	(*D*,*W*)	Φ=($m_f = D$ AND $m_t = W$ AND λ =P)	$\mathfrak{F}$ =($\mathfrak{f}_1(v)${
			`If v.driving_distance < 4 km`
			`  return TRUE;`
			`else return FALSE;`})

	(*D*,*W*)	Φ=($m_f = D$ AND $m_t = W$ AND λ =charging)	$\mathfrak{F}$ =($\mathfrak{f}_1(v)${
			`If v.driving_distance < 8 km`
			`  return TRUE;`
			`else return FALSE;`})

	{*W*,*U*,*S*,*T*}	SCM=($\varphi_{W,U}$ = ($m_f = W$ AND $m_t = U$ AND λ =U-station); $\varphi_{W,S}$ =($m_f = W$ AND $m_t = S$ AND λ =S-station); $\varphi_{W,T}$ =($m_f = W$ AND $m_t = T$ AND λ =T-station); $\varphi_{U,W}$ =($m_f = U$ AND $m_t = W$ AND λ =U-station); $\varphi_{S,W}$ =($m_f = S$ AND $m_t = W$ AND λ =S-station); $\varphi_{T,W}$ =($m_f = T$ AND $m_t = W$ AND λ =T-station))	N/A

	(*D*,{*W*,*U*,*S*,*T*})	Φ=($m_f = D$ AND $m_t = W$ AND λ =P);
		SCM: the same as that in the above {*W*,*U*,*S*,*T*} case.
			$\mathfrak{F}$ =($\mathfrak{f}_1(v)${
			`If v.driving_distance < 4 km`
			`  return TRUE;`
			`else return FALSE;`})

	(*D*,{*W*,*U*,*S*,*T*})	Φ =($m_f = D$ AND ($m_t = U$ OR $m_t = S$) AND λ =P+R);
		SCM: the same as that in the above {*W*,*U*,*S*,*T*} case.
			$\mathfrak{F}$ =($\mathfrak{f}_1(v)${
			`If v.driving_distance < 4 km`
			`  return TRUE;`
			`else return FALSE;`})

	(D,{W,U,S,T})	Φ=($m_f = D$ AND $m_t = W$ AND $\lambda =$charging);
		SCM: the same as that in the above {W,U,S,T} case.
			$\mathfrak{F} =(\mathfrak{f}_1(v)\{$
			`If v.driving_distance < 8 km`
			`  return TRUE;`
			`else return FALSE;})`

Chapter 5

Experiments and Implementations

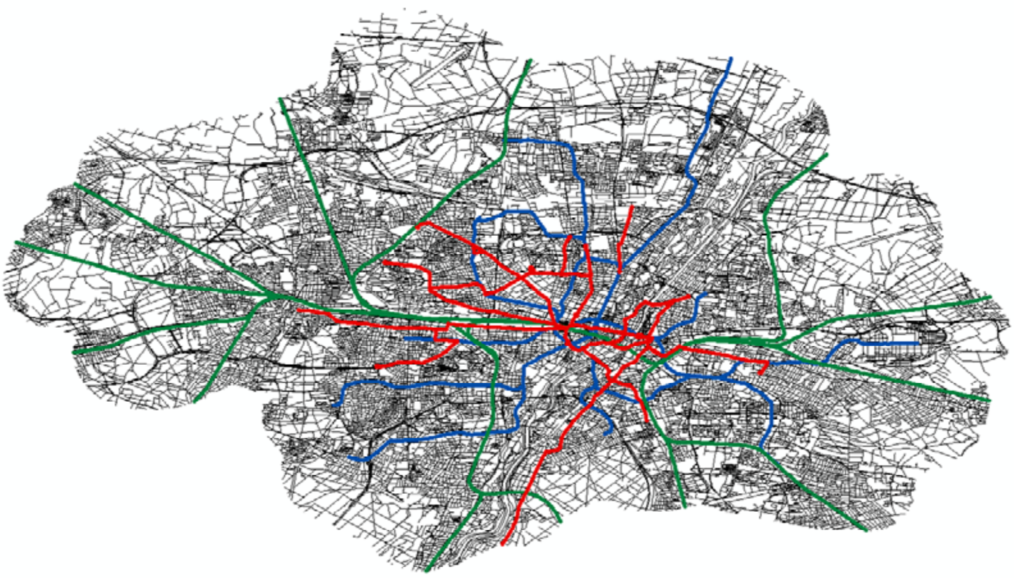
In the experimental phase, two regions in the raw navigation dataset are selected as test beds first. And then, the routing-relevant information is extracted to construct MMGS and Switch Point data model. The data is populated into a database management system (DBMS). After that, the proposed multimodal routing approach is implemented in a C library, and the computing performances of the proposed algorithms are evaluated. The verification is followed by implementation of a web-based multimodal route planning prototype system in which the routing library is integrated with the MMGS and Switch Point data in DBMS as test data.

In this chapter, the above workflow is introduced in detail. Some real multimodal route planning application scenarios in two German cities are described with the help of the online prototype system, and some analysis is conducted.

5.1 Test bed and preprocessing

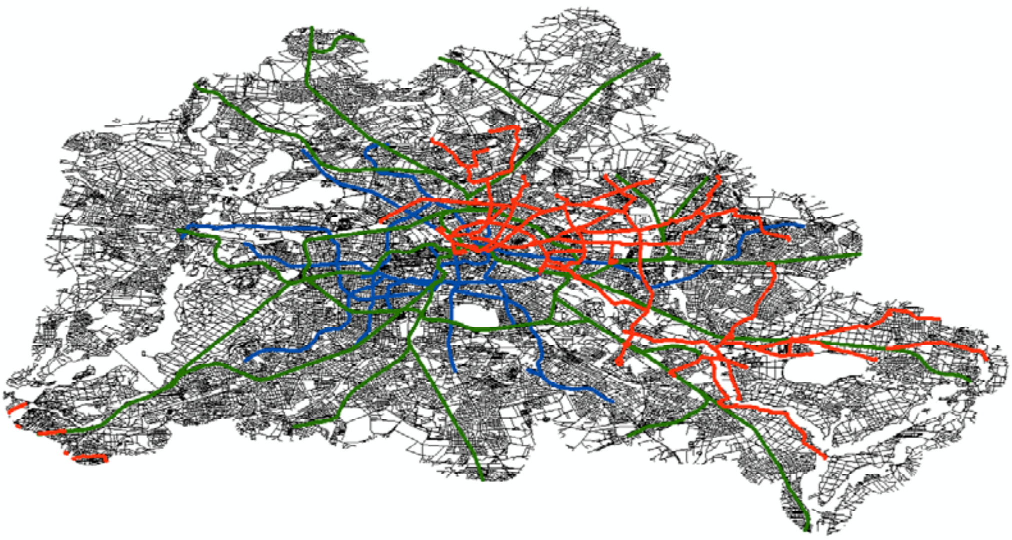
All the networks used throughout the experiments are based on real world data, which means they are not generated artificially or by simulation. However, not all the information needed for modeling is available in the raw data. For example, the timetables of the public transit system are missing. Therefore, the synthetic data is used for this part.

The spatial datasets of two German cities – Munich and Berlin – provided by United Maps GmbH that contain integrated information from NAVTEQ and ATKIS are taken as the test data. Figure 5.1 illustrates the corresponding transportation networks. The lines in green, red, blue and black indicate suburban, tram, underground and road networks respectively. The raw data is provided in shapefile format, and contains 13 different layers listed in Table 5.1.



1 : 126,890

(*a*)



1 : 226,158

(*b*)

Figure 5.1 Transportation networks for the experimental study. (*a*). Munich; (*b*). Berlin.

Table 5.1 Layers in the raw navigational dataset.

Layer	**Geometry type**	**Remarks**
Street_lines	Line	Line features in road network
Street_nodes	Point	Node features in road network
Underground_lines	Line	Line features in underground network
Underground_nodes	Point	Node features in underground network

Suburban_lines	Line	Line features in suburban network
Suburban_nodes	Point	Node features in suburban network
Tram_lines	Line	Line features in tram network
Tram_nodes	Point	Node features in tram network
Car_parkings	Point	Parking lots POI
Underground_stations	Point	Underground stations POI
Suburban_stations	Point	Suburban stations POI
Tram_stations	Point	Tram stations POI
Park_and_Rides	Point	Park and Ride lots POI

The mode graphs and Switch Points are generated according to the modeling approach described in Chapter 3. The data preparation consists of three parts:

- **Road network.** Motorized and pedestrian graphs are derived from road network. The information stored in the raw dataset is sufficient to construct these two mode graphs. The additional work has to be done when constructing pedestrian graph is that to create the public transit station platform vertices and the corresponding edges according to the data collected manually by the author. Layers Street_lines and Street_nodes in raw data are used for this purpose.

- **Public transit network.** With respect to the public transit networks, the raw data is insufficient for the construction of the corresponding three static mode graphs with the separation operations. Underground, suburban and tram graphs are constructed from layers Underground_lines, Underground_nodes, Suburban_lines, Suburban_nodes, Tram_lines and Tram_nodes.

- **Switch Point.** The remaining five layers of POI are used to create the data relevant to Switch Point, which includes seven Switch Types listed in Table 5.2. All of the different types of Switch Point have meanings and can be associated with some real-world facilities except the *physical connection*. This latter type of Switch Point works in a situation that allows an optimal path planning for a car passenger who only needs a temporary parking to get out of the car since the driver can drive the car away. It means that any geographical connection between a motorized road segment and a pedestrian path segment in the network is eligible to be a Switch Point from car driving to walking if a temporary parking is allowed at that connection place.

Table 5.2 Basic information of the Switch Points.

Type	**from_mode**	**to_mode**	**Number of tuples**	
			MUN	**BER**
P	*D*	*W*	123	624
phy-conn	*D*	*W*	29944	58130
P+R	*D*	*U*	34	18
	D	*S*	52	32

Kiss+R	*D*	*U*	92	183
	D	*S*	56	146
	D	*T*	158	422
U-station	*W*	*U*	92	183
	U	*W*	92	183
S-station	*W*	*S*	56	146
	S	*W*	56	146
T-station	*W*	*T*	158	422
	T	*W*	158	422

The basic information of the networks for test is listed in Table 5.3. Five mode graphs are constructed for experimental studies: motorized, pedestrian, underground, suburban and tram graphs denoted by *D*, *W*, *U*, *S* and *T* respectively. The useful attributes for multimodal route calculation are extracted from the raw dataset and exported together with the Switch Relation table into two databases – multimodal_munich and multimodal_berlin – hosted in a DBMS server. The databases containing the network topology and Switch Point information are retrieved on demand and constructed into the data structures representing the in-memory data model.

Table 5.3 Basic Information of the mode graphs.

Data source	**Area of coverage (width, length)***	**Mode**	**|*V*|**	**|*E*|**	**|*E*|/|*V*|**
Munich (MUN)	(29.50, 25.00)	*D*	31636	69757	2.20
		W	46333	128210	2.77
		U	97	200	2.06
		S	73	170	2.33
		T	170	386	2.27
Berlin (BER)	(48.22, 41.71)	*D*	59660	138428	2.32
		W	74423	205928	2.77
		U	202	402	1.99
		S	169	392	2.32
		T	493	1064	2.16

*****The unit of width and length is km

The Entity-Relationship diagram of the two databases stored in DBMS is shown in Figure 5.2.

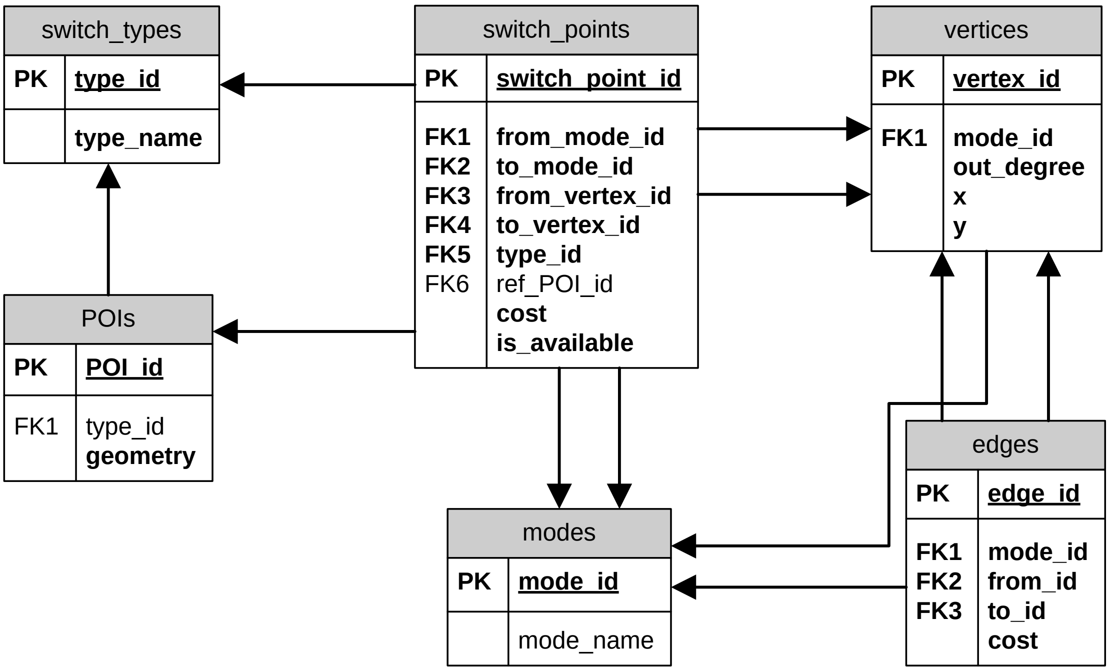


Figure 5.2 Entity-Relationship diagram of multimodal routing database.

5.2 Evaluation of algorithmic performance

Algorithm I and Algorithm II are instantiated by applying the labeling method of TWO-Q search which has been empirically verified to be one of the fastest SPA working on real transportation networks (Zhan 1997). As a result, the abstract Algorithm I and Algorithm II are specified into Multimodal TWO-Q Algorithm (MMTQ) and Plug-and-Play Multimodal TWO-Q Algorithm (PNP-MMTQ). The two concrete algorithms are implemented in a MMSP calculation library – libmmspa – written in C. They work in the hybrid fashion to cope with different types of multimodal routing plans. The evaluations are performed on a PC with a 3.16GHz Intel Core2 Duo CPU and 1GB physical memory, running under Ubuntu Linux with the kernel version of 2.6.28. The code is compiled using GCC version 4.3.3 with O3 optimization flag.

To evaluate the performances of the algorithms, various input modes and the Switch Conditions are adopted in Table 5.4. Eight routing plans are involved in the performance evaluation, where P1 and P2 are mono-modal; P3 and P4 stand for double-modal routing tasks with different Switch Types in road network; P5 for the combined public transportation network; P6, P7 and P8 for the tasks of taking a car first, and then public transportation system, but with different Switch Types.

Table 5.4 Multimodal routing plans for performance evaluation.

Key	Input modes	Input Switch Conditions
P1	*D*	N/A
P2	*W*	N/A
P3	(*D*,*W*)	Φ=($m_f = D$ AND $m_t = W$ AND λ =P)
P4		Φ=($m_f = D$ AND $m_t = W$ AND λ =phy-conn)

P5	{W,U,S,T}	SCM=($\varphi_{W,U}$ =($m_f = W$ AND $m_t = U$ AND $\lambda =$U-station); $\varphi_{W,S}$ =($m_f = W$ AND $m_t = S$ AND $\lambda =$S-station); $\varphi_{W,T}$ =($m_f = W$ AND $m_t = T$ AND $\lambda =$T-station); $\varphi_{U,W}$ =($m_f = U$ AND $m_t = W$ AND $\lambda =$U-station); $\varphi_{S,W}$ =($m_f = S$ AND $m_t = W$ AND $\lambda =$S-station); $\varphi_{T,W}$ =($m_f = T$ AND $m_t = W$ AND $\lambda =$T-station))
P6		Φ=($m_f = D$ AND ($m_t = U$ OR $m_t = S$) AND $\lambda =$P+R); SCM: the same as that in the above {W,U,S,T} case.
P7	(D,{W,U,S,T})	Φ=($m_f = D$ AND $m_t = W$ AND $\lambda =$P); SCM: the same as above.
P8		Φ=($m_f = D$ AND $m_t = W$ AND $\lambda =$Kiss+R); SCM: the same as above.

The reported running times do not include data retrieval or log output. For the graphs corresponding to mode D and W, a sample of 500 vertices on each dataset were randomly selected at the outset and designated as the two groups of sample source vertices. For a given input routing plan and the sample source vertices, an average running time of executing the path finding algorithm on all the sample source vertices was recorded.

Table 5.5 Computing speeds for given multimodal routing plans.

Key	Routing approach	Mean running time (ms)		Standard Deviation (ms)	
		MUN	BER	MUN	BER
P1	MMTQ	24.5	55.7	5.7	7.7
P2	MMTQ	44.7	80.7	5.4	7.1
P3	MMTQ	69.0	152.0	13.4	22.2
P4	MMTQ	74.0	150.2	7.1	14.4
P5	PNP-MMTQ	45.0	87.7	5.8	12.8
P6	MMTQ+PNP-MMTQ	67.8	140.9	13.1	18.4
P7	MMTQ+PNP-MMTQ	67.1	148.2	13.2	20.2
P8	MMTQ+PNP-MMTQ	66.2	143.7	14.1	19.7

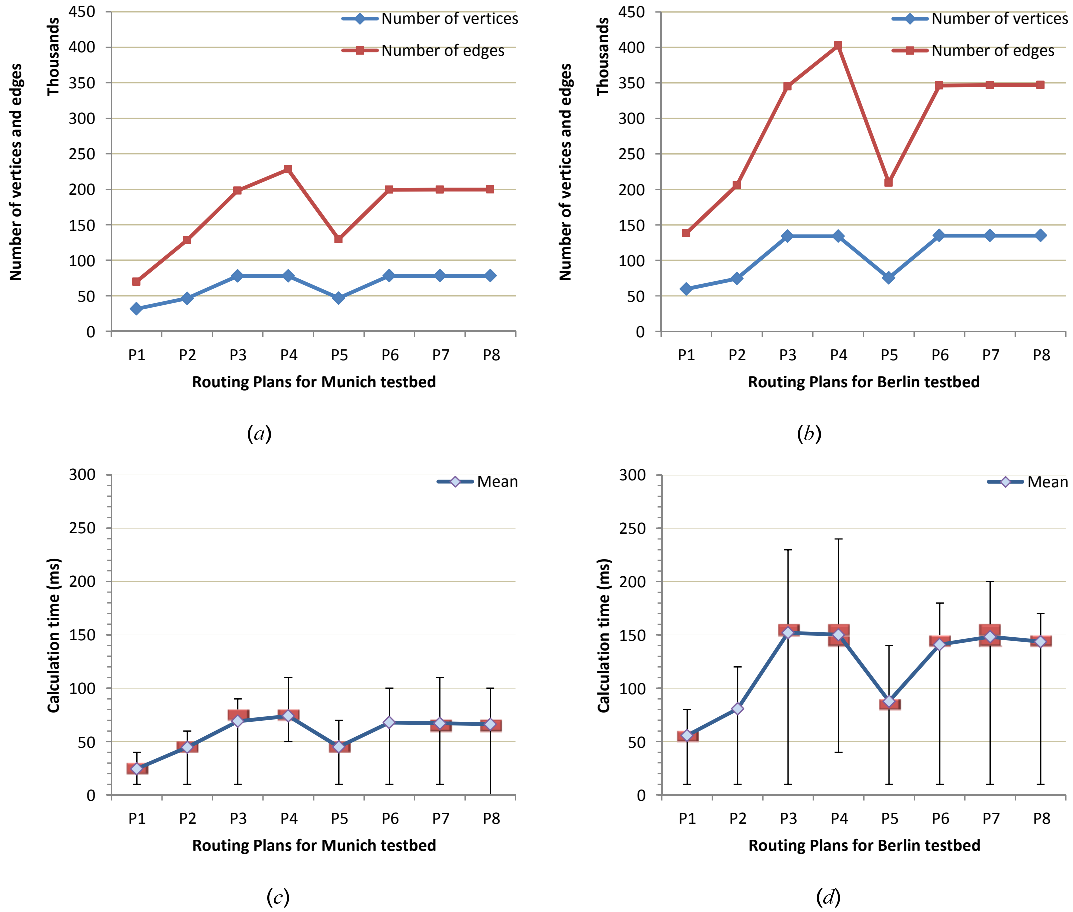


Figure 5.3 Input data volume and computing speeds. (*a*). Input data for Munich test bed; (*b*). Input data for Berlin test bed; (*c*). Path-finding times on Munich test bed; (*d*). Path-finding times on Berlin test bed.

The evaluation results are presented in Table 5.5 and a box-and-whisker plot in Figure 5.3. They show that the MMSP finding tasks can be finished at the millisecond-level on an inner-city scale transportation network. And the running time keeps pace with the volumes of the input data.

5.3 Prototype systems

The convincing evaluation results has led to the development of an online prototype system *Multimodal Route Planner* that provides multimodal optimal (fastest or shortest) path-finding services in urban transportation networks for both Munich[40] and Berlin[41]. The system is publicly accessible from the Internet. The architecture of the system is shown in Figure 5.4, which reveals a typical Browser/Server schema.

40 http://129.187.175.46:8888/multimodal_munich/route_planner/

41 http://129.187.175.46:8888/multimodal_berlin/route_planner/

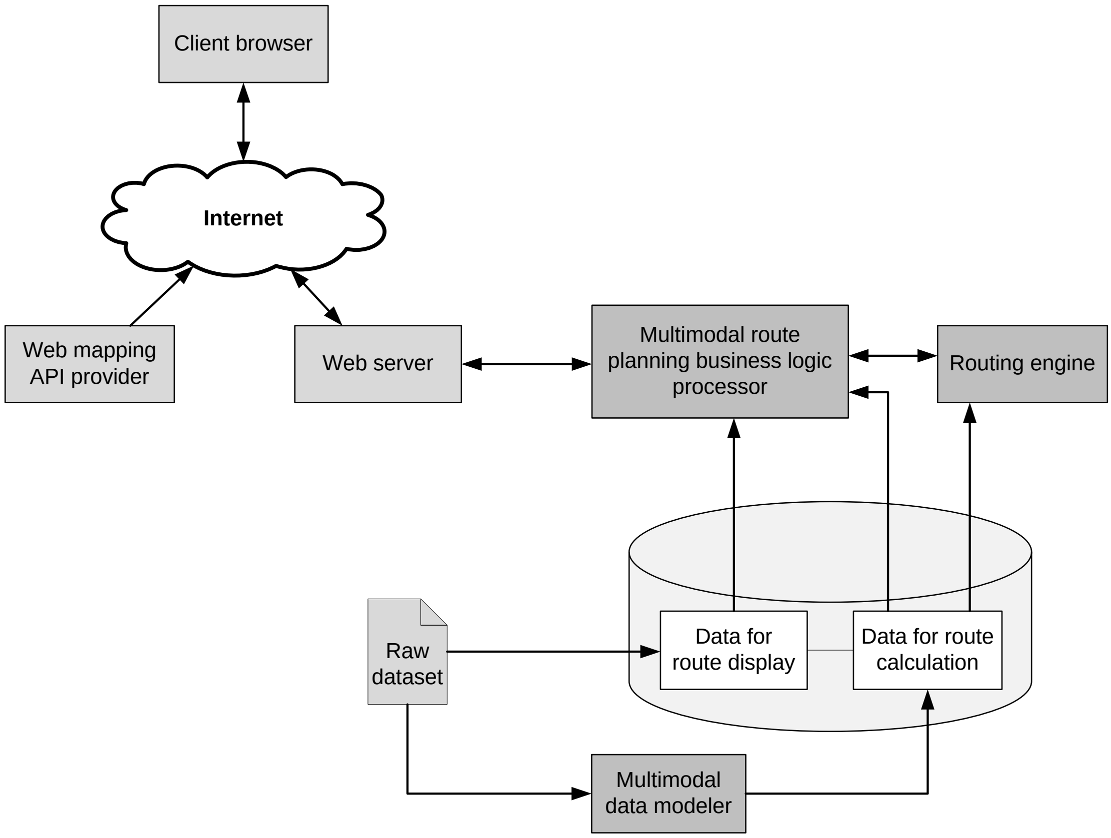


Figure 5.4 Architecture of the prototype system.

In the same development environment as the routing library and the two multimodal routing databases, the Multimodal route planning business logic processor is implemented with Ruby on Rails web framework. The versions of the development tools are Ruby 1.8.7 and Rails 2.3.3 respectively. Two plug-ins for Ruby on Rails – Georuby version 1.3.4 and Spatial_adapter version 0.3.1 – are installed to make it possible to handle spatial data types in PostgreSQL 8.3.11 (with PostGIS 1.3.3 extension) DBMS server. Additionally, a JavaScript library Mapstraction is used, which integrates different web mapping APIs (e.g. Google Maps API) to display base maps and show routing results on the maps in a web page. For the Web server, Apache version 2.2.11 is set up with an additional module Passenger version 2.2.10 installed to help deploy the Ruby on Rails application. The implemented C library, libmmspa, is used as the Routing engine. Besides the data for route calculation created via the Multimodal data modeler, the raw dataset is imported into the DBMS for the purpose of visualizing the routing results. The tool used for importing shapefile into PostgreSQL database is the utility ruby program shp2sql.

The system infers all the feasible route calculation plans by acquiring some customized routing options as well as the source and target points from a user with a friendly interface. Then it calculates all the feasible routes and sorts them by their travel times. Figure 5.5 shows a screenshot of the system for Munich on initialization.

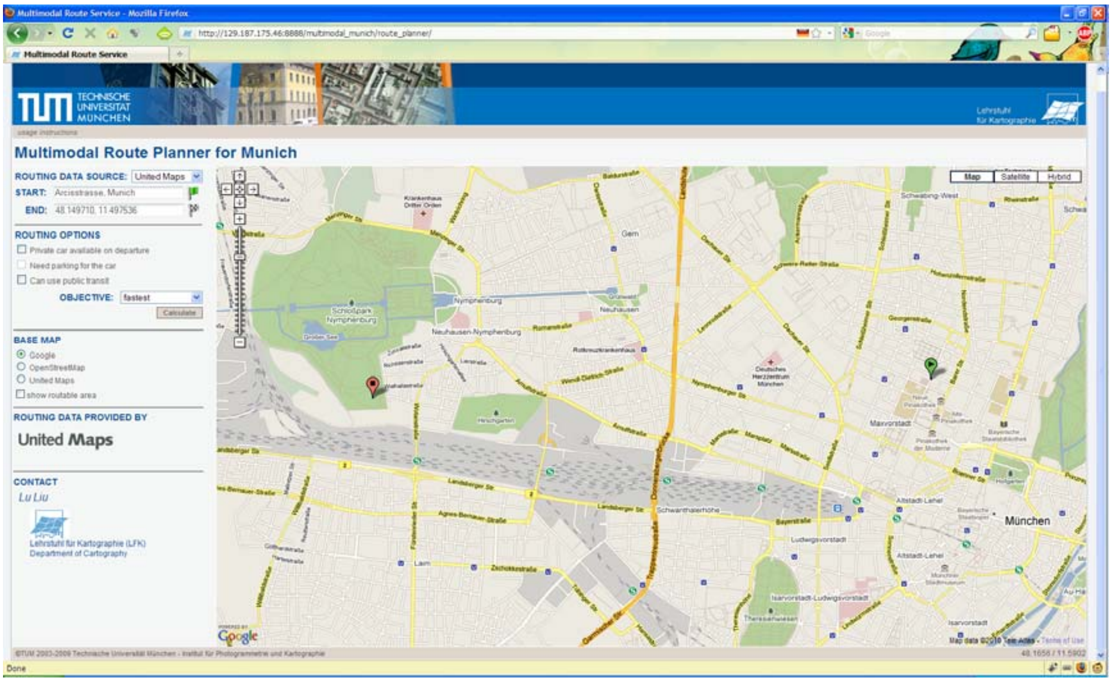


Figure 5.5 Screenshot of Multimodal Route Planner for Munich in a browser.

5.4 Case studies

In the following sections, two real-world multimodal route-planning scenarios with various routing options are introduced and their routing results generated by the prototype system are analyzed.

5.4.1 Scenario 1: Travelers in Munich

In this scenario, the traveler decides to travel from the crossing of Arcisstraße and Heßstraße near Technische Universität München to a pedestrian path junction with coordinates of longitude 48.150 and latitude 11.497 in Schloßpark Nymphenburg as shown in Figure 5.6.

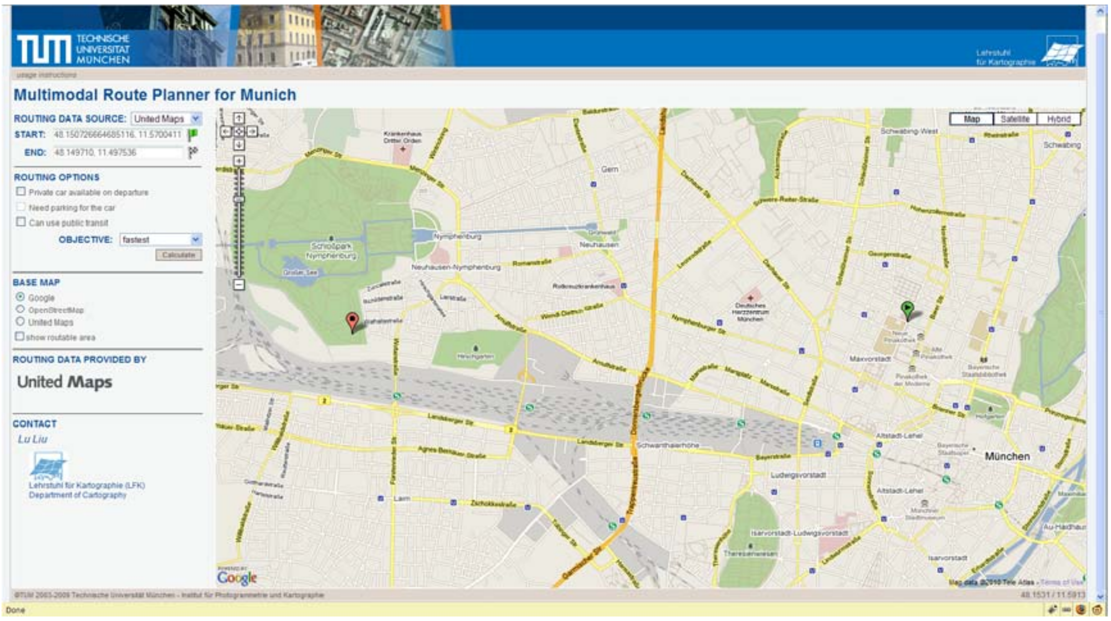


Figure 5.6 Route source and target point setting for the case study in Munich.

After determining the source/target pair for this case, nine different combinations of routing options (listed in Table 5.6) are chosen.

Table 5.6 Routing options for the case study in Munich.

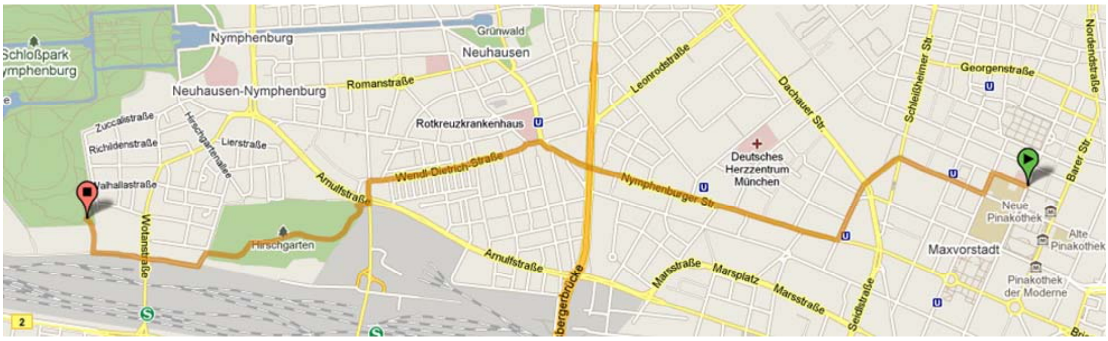
Options	**Private car available on departure**	**Need parking for car**	**Can use public transit**			**Objective**
			Suburban	**Underground**	**Tram**	
1	NO	NO	NO	NO	NO	fastest
2	NO	NO	NO	YES	NO	fastest
3	NO	NO	YES	YES	YES	fastest
4	YES	NO	NO	NO	NO	fastest
5	YES	YES	NO	NO	NO	fastest
6	YES	YES	NO	NO	NO	shortest
7	YES	YES	NO	NO	YES	fastest
8	YES	NO	YES	YES	YES	fastest
9	YES	YES	YES	YES	YES	fastest

For each routing option combination, a group of routing results can be obtained from the system (see Table 5.7).

Table 5.7 Multimodal route planning results according to different routing options in Table 5.6.

Options	Multimodal route planning results			
	Mode sequence	Switch Point(s)	Cost remarks	Route sketch
1	*W*	N/A	6.35 km, 76 mins	Figure 5.7(*a*)
2	(*W*,*U*,*W*)	Underground stations: Theresienstraße (U2), Hauptbahnhof (U2, U5), Laimer Platz (U5)	8.44 km, 43 mins where 2.59 km, 31 mins for *W*	Figure 5.7(*i*)
	W	N/A	6.35 km, 76 mins	Figure 5.7(*a*)
3	(*W*,*T*,*W*,*S*,*W*)	Tram stations: Pinakotheken (27), Ottostraße (27) Suburban stations: Karlsplatz (S1-S4, S6-S8), Laim (S1-S4, S6-S8)	7.53 km, 31 mins where 1.71 km, 20 mins for *W*	Figure 5.7(*f*)
	W	N/A	6.35 km, 76 mins	Figure 5.7(*a*)
4	(*D*,*W*)	A temporary parking position near Winfriedstraße 19	6.96 km, 13 mins where 0.27 km, 3 mins for *W*	Figure 5.7(*c*)
	W	N/A	6.35 km, 76 mins	Figure 5.7(*a*)
5	(*D*,*W*)	A parking lot in Schlosspark Nymphenburg	7.26 km, 25 mins where 1.29 km, 15 mins for *W*	Figure 5.7(*b*)
	W	N/A	6.35 km, 76 mins	Figure 5.7(*a*)
6	*W*	N/A	6.35 km, 76 mins	Figure 5.7(*a*)
7	(*D*,*W*)	A parking lot in Schlosspark Nymphenburg	7.26 km, 25 mins where 1.29 km, 15 mins for *W*	Figure 5.7(*b*)
	(*W*,*T*,*W*,*T*,*W*)	Tram stations: Pinakotheken (27), Ottostraße (27), Lenbachplatz (19), Fürstenrieder Straße (19)	7.93 km, 40 mins where 1.71 km, 20 mins for *W*	Figure 5.7(*e*)
	W	N/A	6.35 km, 76 mins	Figure 5.7(*a*)
8	(*D*,*W*)	A temporary parking position near Winfriedstraße 19	6.96 km, 13 mins where 0.27 km, 3 mins for *W*	Figure 5.7(*c*)

	(D,S,W)	Kiss and Ride at Suburban station Hirschgarten (S1-S4, S6, S8) Suburban station: Laim (S1-S4, S6, S8)	7.43 km, 20 mins where 0.99 km, 11 mins for W	Figure 5.7(g)
	(W,T,W,S,W)	Tram stations: Pinakotheken (27), Ottostraße (27) Suburban stations: Karlsplatz (S1-S4, S6, S8), Laim (S1-S4, S6, S8)	7.53 km, 31 mins where 1.71 km, 20 mins for W	Figure 5.7(f)
	W	N/A	6.35 km, 76 mins	Figure 5.7(a)
9	(D,W,S,W)	Parking lot: Landshuter Allee. Suburban stations: Donnersbergerbrücke (S1-S4, S6, S8), Laim (S1-S4, S6, S8)	7.33 km, 22 mins where 1.23 km, 14 mins for W	Figure 5.7(d)
	(D,S,W)	Park and Ride: P+R Heimeranplatz. Suburban stations: Heimeranplatz (S7, S20, S27), Donnersbergerbrücke(S1-S4, S6-S8, S20, S27), Laim (S1-S4, S6, S8)	10.07 km, 23 mins where 0.99 km, 11 mins for W	Figure 5.7(h)
	(D,W)	A parking lot in Schlosspark Nymphenburg	7.26 km, 25 mins where 1.29 km, 15 mins for W	Figure 5.7(b)
	(W,T,W,S,W)	Tram stations: Pinakotheken (27), Ottostraße (27) Suburban stations: Karlsplatz (S1-S4, S6-S8), Laim (S1-S4, S6-S8)	7.53 km, 31 mins where 1.71 km, 20 mins for W	Figure 5.7(f)
	W	N/A	6.35 km, 76 mins	Figure 5.7(a)


Nymphenburg
Grünwald
Neuhausen
Neuhausen-Nymphenburg
Romanstraße
Rotkreuzkrankenhaus
Leonrodstraße
Dachauer Str.
Georgenstraße
Nordendstraße
Zuccalistraße
Richildenstraße
Lierstraße
Hirschgartenallee
Wendl-Dietrich-Straße
Deutsches Herzzentrum München
Arnulfstraße
Nymphenburger Str.
Barer Str.
Neue Pinakothek
Alte Pinakothek
Maxvorstadt
Valhallastraße
Wotanstraße
Hirschgarten
Marsstraße
Marsplatz
Seidlstraße
Pinakothek der Moderne

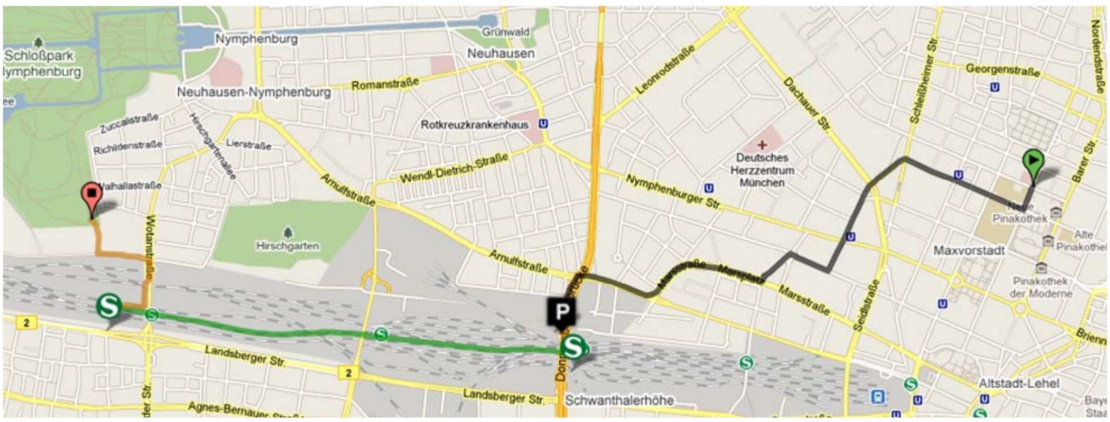
(*a*)


Nymphenburg
Grünwald
Neuhausen
Neuhausen-Nymphenburg
Romanstraße
Rotkreuzkrankenhaus
Leonrodstraße
Dachauer Str.
Georgenstraße
Nordendstraße
Zuccalistraße
Richildenstraße
Lierstraße
Hirschgartenallee
Wendl-Dietrich-Straße
Deutsches Herzzentrum München
Arnulfstraße
Nymphenburger Str.
Barer Str.
Maxvorstadt
Alte Pinakothek
Valhallastraße
Wotanstraße
Hirschgarten
Marsstraße
Marsplatz
Seidlstraße
Pinakothek der Moderne

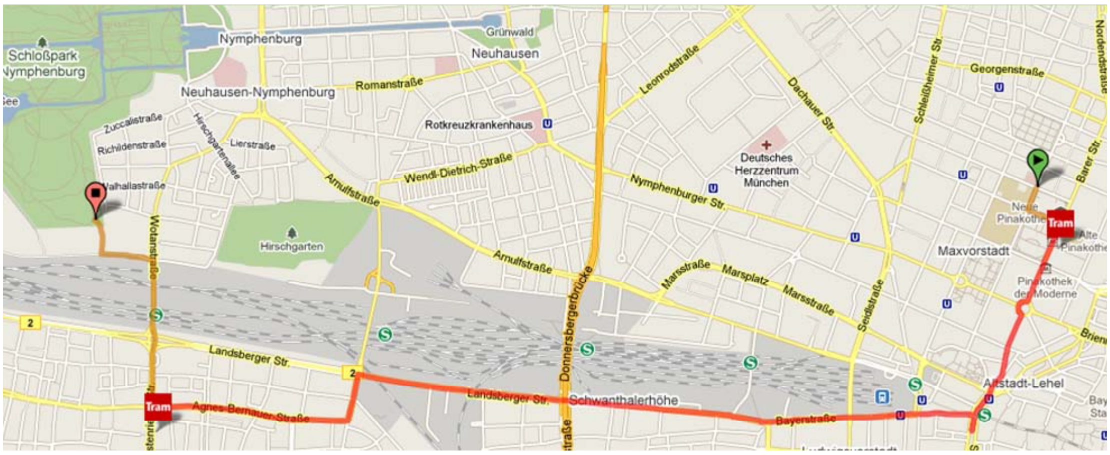
(*b*)

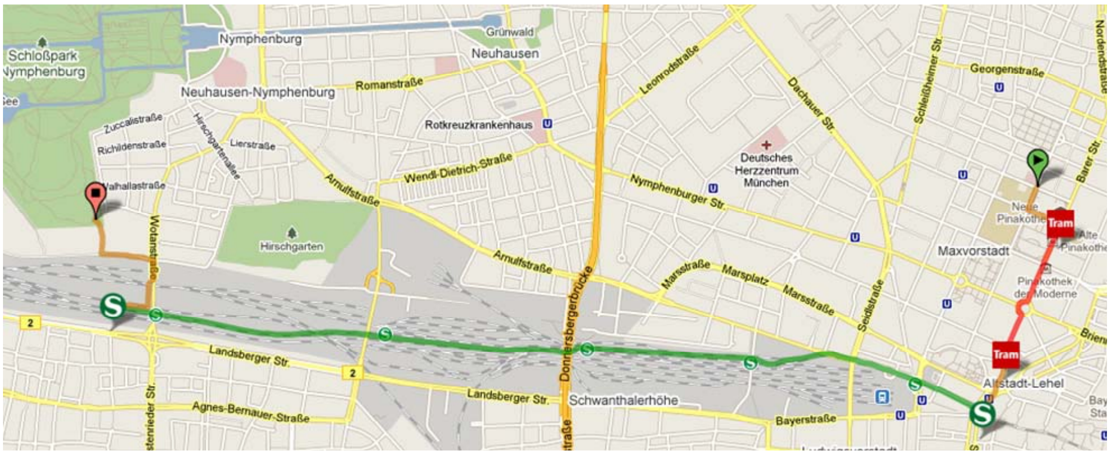

Nymphenburg
Grünwald
Neuhausen
Neuhausen-Nymphenburg
Romanstraße
Rotkreuzkrankenhaus
Leonrodstraße
Dachauer Str.
Georgenstraße
Nordendstraße
Zuccalistraße
Richildenstraße
Lierstraße
Hirschgartenallee
Wendl-Dietrich-Straße
Deutsches Herzzentrum München
Arnulfstraße
Nymphenburger Str.
Barer Str.
Maxvorstadt
Alte Pinakothek
Wotanstraße
Hirschgarten
Marsstraße
Marsplatz
Seidlstraße
Pinakothek der Moderne

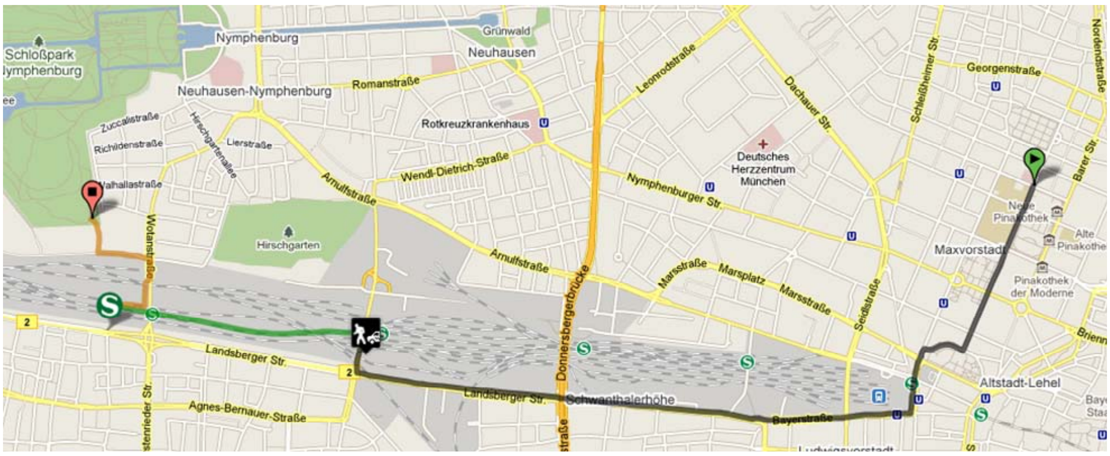
(*c*)

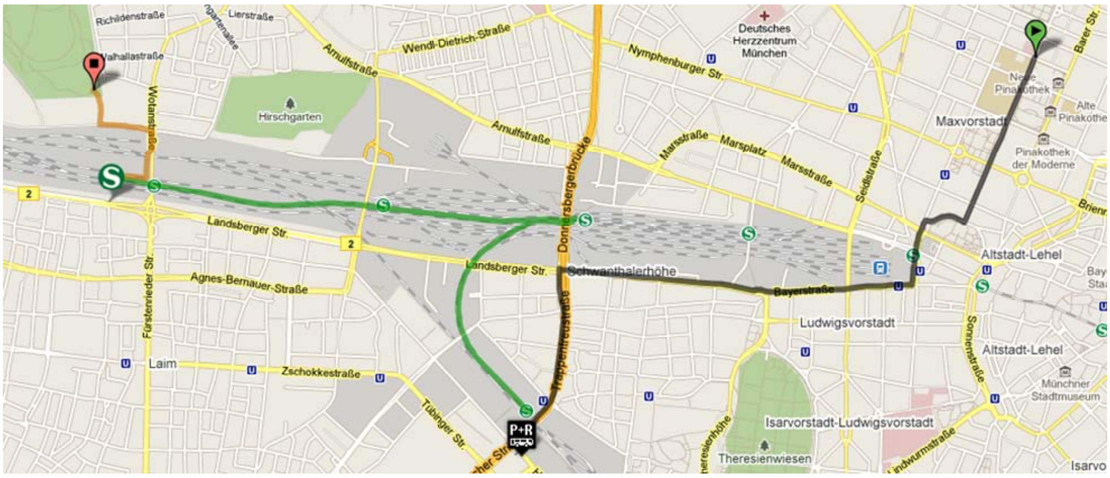

Schloßpark Nymphenburg
Nymphenburg
Grünwald
Neuhausen
Neuhausen-Nymphenburg
Romanstraße
Rotkreuzkrankenhaus
Leonrodstraße
Dachauer Str.
Georgenstraße
Nordendstraße
Zuccalistraße
Richildenstraße
Lierstraße
Hirschgartenallee
Wendl-Dietrich-Straße
Deutsches Herzzentrum München
Arnulfstraße
Nymphenburger Str.
Barer Str.
Neue Pinakothek
Alte Pinakothek
Maxvorstadt
Valhallastraße
Wotanstraße
Hirschgarten
Marsstraße
Marsplatz
Seidlstraße
Pinakothek der Moderne
Landsberger Str.
Agnes-Bernauer-Straße
Schwanthalerhöhe
Altstadt-Lehel

(*d*)

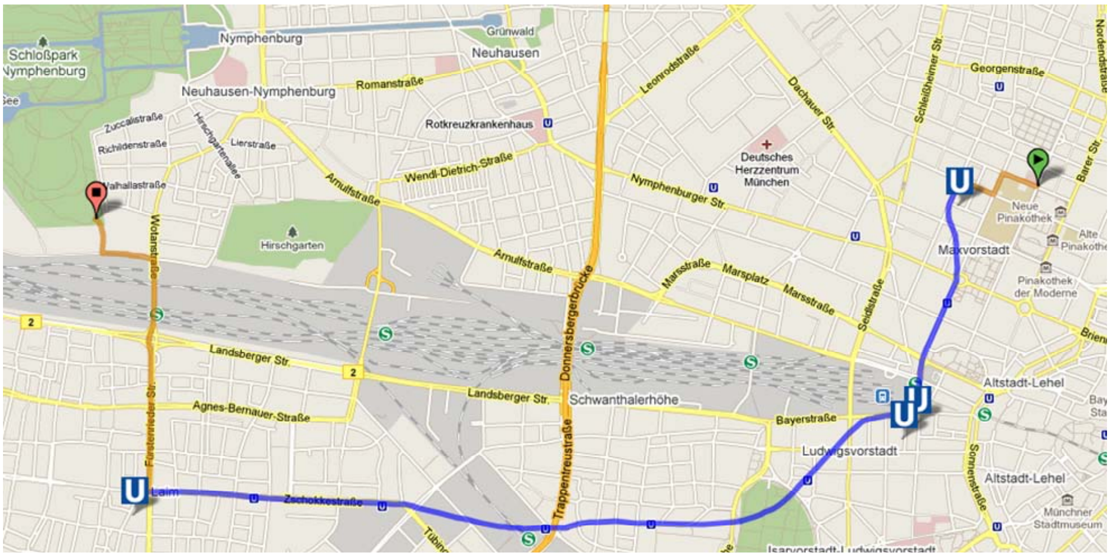

Schloßpark
Nymphenburg
Nymphenburg
Grünwald
Neuhausen
Neuhausen-Nymphenburg
Romanstraße
Rotkreuzkrankenhaus
Leonrodstraße
Dachauer Str.
Schleißheimer Str.
Georgenstraße
Nordendstraße
Barer Str.
Zuccalistraße
Richildenstraße
Hirschgartenallee
Lierstraße
Wendl-Dietrich-Straße
Deutsches Herzzentrum München
Nymphenburger Str.
Valhallastraße
Arnulfstraße
Hirschgarten
Wotanstraße
Maxvorstadt
Neue Pinakothek
Alte Pinakothek
Pinakothek der Moderne
Marsstraße
Marsplatz
Seidlstraße
Donnersbergerbrücke
Landsberger Str.
Agnes-Bernauer-Straße
Schwanthalerhöhe
Bayerstraße
Altstadt-Lehel
Tram
(e)


Schloßpark
Nymphenburg
Nymphenburg
Grünwald
Neuhausen
Neuhausen-Nymphenburg
Romanstraße
Rotkreuzkrankenhaus
Leonrodstraße
Dachauer Str.
Schleißheimer Str.
Georgenstraße
Nordendstraße
Barer Str.
Zuccalistraße
Richildenstraße
Hirschgartenallee
Lierstraße
Wendl-Dietrich-Straße
Deutsches Herzzentrum München
Nymphenburger Str.
Valhallastraße
Arnulfstraße
Hirschgarten
Wotanstraße
Maxvorstadt
Neue Pinakothek
Alte Pinakothek
Pinakothek der Moderne
Marsstraße
Marsplatz
Seidlstraße
Donnersbergerbrücke
Landsberger Str.
Agnes-Bernauer-Straße
Schwanthalerhöhe
Bayerstraße
Altstadt-Lehel
Tram
(f)


Schloßpark
Nymphenburg
Nymphenburg
Grünwald
Neuhausen
Neuhausen-Nymphenburg
Romanstraße
Rotkreuzkrankenhaus
Leonrodstraße
Dachauer Str.
Schleißheimer Str.
Georgenstraße
Nordendstraße
Barer Str.
Zuccalistraße
Richildenstraße
Hirschgartenallee
Lierstraße
Wendl-Dietrich-Straße
Deutsches Herzzentrum München
Nymphenburger Str.
Valhallastraße
Arnulfstraße
Hirschgarten
Wotanstraße
Maxvorstadt
Neue Pinakothek
Alte Pinakothek
Pinakothek der Moderne
Marsstraße
Marsplatz
Seidlstraße
Donnersbergerbrücke
Landsberger Str.
Agnes-Bernauer-Straße
Schwanthalerhöhe
Bayerstraße
Altstadt-Lehel
(g)



(*h*)



(*i*)

Figure 5.7 Multimodal route planning results of the Munich scenario. The routes in black, brown, blue, green and red indicate motorized, pedestrian, underground, suburban and tram lines respectively.

As it can be seen from Table 5.7 and Figure 5.7, if there is a private car available on departure, the fastest way is to take the car as a passenger, drive to Winfriedstraße 19 first, and then let the driver drive away and the passenger walk to the destination. This double-modal route is shown in Figure 5.7(*c*), and will take ca. 13 minutes and 6.96 km where ca. 3 minutes and 270m for walking. If all the public transit modes are additionally taken into account and the user has to be the driver (as depicted by routing option 9 in Table 5.6), the fastest way is composed of: drive to the parking lot Landshuter Allee, park the car there, take the suburban train from Donnersbergerbrüke station to Laim station, and then walk to the destination. Such a route (Figure 5.7(*d*)) will take ca. 22 minutes and 7.33 km where ca. 14 minutes and 1.23 km for walking.

5.4.2 Scenario 2: Electronic car drivers in Berlin

Unlike the basic multimodal routing functions as described in the demo for Munich, Scenario 2 is assumed for electronic car drivers in Berlin. Electronic car drivers have additional constraints with regard to the driving distance limited by capability of batteries.

For the case study in this scenario, the electronic car driver decides to start from Lichtenberg (Siegfriedstraße 203) to a pedestrian junction with coordinates of longitude 52.513, latitude 13.357 in Tiergarten as shown in Figure 5.8.

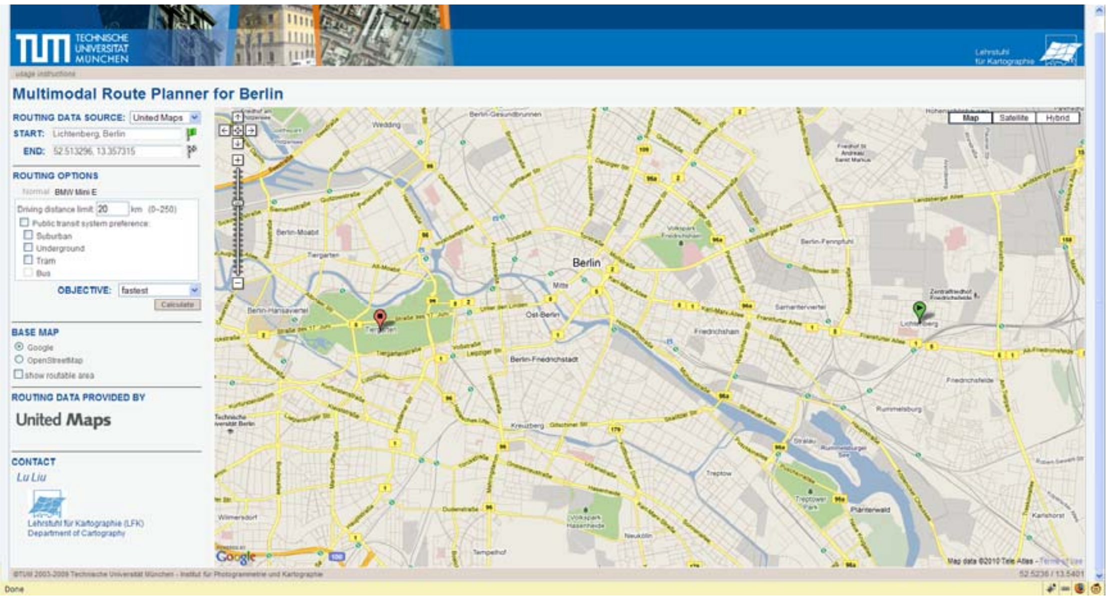


Figure 5.8 Route source and target point setting for the case study in Berlin.

The eight routing option combinations are listed in Table 5.8.

Table 5.8 Routing options for the case study in Berlin.

Options	Driving distance limit	Can use public transit			Objective
		Suburban	Underground	Tram	
1	20	NO	NO	NO	fastest
2	20	NO	NO	NO	shortest
3	10	NO	NO	NO	fastest
4	5	NO	YES	NO	fastest
5	7	NO	NO	YES	fastest
6	9	YES	YES	NO	fastest
7	8	YES	YES	YES	fastest
8	200	YES	YES	YES	fastest

With respect to the Switch Type in this case, the more choices become feasible. Charging stations are taken into account other than those used in the Munich demo, which leads to more diverse routing results. The system provides routing results (see Table 5.9 and Figure 5.9) corresponding to each of the routing option combinations.

Table 5.9 Multimodal route planning results according to different routing options in Table 5.8.

options	Multimodal route planning results			
	Mode sequence	**Switch Point(s)**	**Cost remarks**	**Route sketch**
1	(D,W)	Parking lot: Potsdamer Platz/Tor/Arkaden at Ludwig-Beck-Straße	11.36 km, 31 mins where 1.36 km, 16 mins for W	Figure 5.9(a)
	(D,W)	Charging station: Leipziger Platz	11.16 km, 34 mins where 1.66 km, 19 mins for W	Figure 5.9(b)
	W	N/A	10.45 km, 125 mins	Figure 5.9(c)
2	W	N/A	10.45 km, 125 mins	Figure 5.9(c)
3	(D,W)	Charging station: Leipziger Platz	11.16 km, 34 mins where 1.66 km, 19 mins for W	Figure 5.9(b)
	(D,W)	Parking lot: Hildegard-Jadamowitz-Straße	11.03 km, 83 mins where 6.4 km, 76 mins for W	Figure 5.9(d)
	W	N/A	10.45 km, 125 mins	Figure 5.9(c)
4	(W,U,W)	Underground stations: Lichtenberg (U5), Alexanderplatz (U5, U2), Potsdamer Platz (U2)	11.56 km, 48 mins where 2.0 km, 24 mins for W	Figure 5.9(e)
	(D,W,U,W)	Parking lot: Rathausstraße Underground stations: Frankfurter Allee (U5), Alexanderplatz (U5, U2), Potsdamer Platz (U2)	11.83 km, 48 mins where 2.03 km, 24 mins for W	Figure 5.9(f)
	(D,W)	Parking lot: Rathausstraße	10.78 km, 111 mins where 9.05 km, 108 mins for W	Figure 5.9(g)
	W	N/A	10.45 km, 125 mins	Figure 5.9(c)
5	(D,W,T,W,T,W)	Charging station: TOTAL at Margarete-Sommer.Straße Tram stations: Kniprodestr./Danziger Str. (M10), Torstr./ U Oranienburger Tor , U Oranienburger Tor , S+U Friedrichstr.	14.25 km, 62 mins where 2.77 km, 33 mins for W	Figure 5.9(h)

	(D,W,T,W,T,W)	Parking lot: Bernhard-Bästlein-Straße Tram stations: Bernhard-Bästlein-Str. , Torstr./ U Oranienburger Tor , U Oranienburger Tor , S+U Friedrichstr.	13.35 km, 63 mins where 2.77 km, 33 mins for W	Figure 5.9(i)
	(T,W,T,W)	Tram stations: Fanningerstr. (), Torstr./ U Oranienburger Tor ,U Oranienburger Tor , S+U Friedrichstr.	13.17 km, 66 mins where 2.64 km, 31 mins for W	Figure 5.9(j)
	(D,W)	Charging station: TOTAL at Holzmarktstraße	11.28 km, 75 mins where 5.48 km, 65 mins for W	Figure 5.9(k)
	(D,W)	Parking lot: Rathauspassagen at Gabelsbergerstraße	10.65 km, 97 mins where 7.75 km, 93 mins for W	Figure 5.9(l)
	W	N/A	10.45 km, 125 mins	Figure 5.9(c)
6	(D,W,U,W,S,W)	Charging station: Friedrichstraße Underground stations: Stadtmitte (U6), Friedrichstraße (U6) Suburban stations: Friedrichstraße (S5, S7, S9, S75), Bellevue (S5, S7, S9, S75)	14.01 km, 35 mins where 1.31 km, 15 mins for W	Figure 5.9(m)
	(D,W,U,W,S,W)	Parking lot: Rathauspassagen at Gabelsbergerstraße Underground stations: Samariterstraße (U5), Alexanderplatz (U5) Suburban stations: Alexanderplatz (S5, S7, S9, S75), Bellevue (S5, S7, S9, S75)	12.87 km, 39 mins where 1.54 km, 18 mins for W	Figure 5.9(n)
	(D,W)	Charging station: ContiPark at Glinkastraße /Französische Straße	11.14 km, 41 mins where 2.34 km, 28 mins for W	Figure 5.9(o)
	(W,U,W,S,W)	Underground stations: Lichtenberg (U5), Frankfurter Allee (U5) Suburban stations: Frankfurter Allee (S8, S41, S42, S85), Ostkreuz (S5, S7, S8, S9, S41, S42, S75, S85), Bellevue (S5, S7, S9, S75)	14.09 km, 41 mins where 1.77 km, 21 mins for W	Figure 5.9(p)
	(D,W)	Parking lot: Kosmos at Ludwig-Pick-Straße	10.69 km, 88 mins where 6.88 km, 82 mins for W	Figure 5.9(q)
	W	N/A	10.45 km, 125 mins	Figure 5.9(c)

7	(T,W,U,W,S,W)	Tram stations: Fanningerstr. , S+U Lichtenberg/Siegfriedstr. Underground stations: Lichtenberg (U5), Frankfurter Allee (U5) Suburban stations: Frankfurter Allee (S8, S41, S42, S85), Ostkreuz (S5, S7, S8, S9, S41, S42, S75, S85), Bellevue (S5, S7, S9, S75)	14.09 km, 39 mins where 1.54 km, 18 mins for W	Figure 5.9(r)
	(D,W,U,W,S,W)	Parking lot: Rathauspassagen at Gabelsbergerstraße Underground stations: Samariterstraße (U5), Alexanderplatz (U5) Suburban stations: Alexanderplatz (S5, S7, S9, S75), Bellevue (S5, S7, S9, S75)	12.87 km, 39 mins where 1.54 km, 18 mins for W	Figure 5.9(n)
	(D,W,S,W)	Charging station: TOTAL at Holzmarktstraße Suburban stations: Berlin Ostbahnhof (S5, S7, S9, S75), Bellevue (S5, S7, S9, S75)	14.55 km, 44 mins where 1.73 km, 20 mins for W	Figure 5.9(s)
	(D,W)	Charging station: TOTAL at Holzmarktstraße	11.28 km, 75 mins where 5.48 km, 65 mins for W	Figure 5.9(k)
	(D,W)	Parking lot: Kosmos at Ludwig-Pick-Straße	10.69 km, 88 mins where 6.88 km, 82 mins for W	Figure 5.9(q)
	W	N/A	10.45 km, 125 mins	Figure 5.9(c)
8	(D,W)	Parking lot: Sony Center at Ben-Gurion-Straße	11.24 km, 27 mins where 1.06 km, 12 mins for W	Figure 5.9(t)
	(D,W)	Charging station: Leipziger Platz	11.16 km, 34 mins where 1.66 km, 19 mins for W	Figure 5.9(b)
	(T,W,U,W,S,W)	Tram stations: Fanningerstr. , S+U Lichtenberg/Siegfriedstr. Underground stations: Lichtenberg (U5), Frankfurter Allee (U5) Suburban stations: Frankfurter Allee (S8, S41, S42, S85), Ostkreuz (S5, S7, S8, S9, S41, S42, S75, S85), Bellevue (S5, S7, S9, S75)	14.09 km, 39 mins where 1.54 km, 18 mins for W	Figure 5.9(r)
	(D,U,W)	Park and Ride: P+R Bundesplatz Underground stations: Bundesplatz (U9), Hansaplatz (U9)	21.1 km, 45 mins where 1.24 km, 14 mins for W	Figure 5.9(u)
	W	N/A	10.45 km, 125 mins	Figure 5.9(c)


Berlin
Mitte
Ost-Berlin
Berlin-Friedrichstadt
Friedrichshain
Samariterviertel
Berlin-Fennpfuhl
Volkspark Friedrichshain
Lichtenberg
Tiergarten
Alt-Moabit
Invalidenstraße
Torstraße
Friedrichstraße
Unter den Linden
Straße des 17. Juni
Tiergartenstraße
Voßstraße
Leipziger Str
Holzmarktstraße
Mühlenstraße
Warschauer Str
Petersburger Str
Landsberger Allee
Storkower Str
Möllendorffstraße
Karl-Marx-Allee
Frankfurter Allee
Boxhagener Str
Kurfürstenstraße
Lützowufer
Mollstraße

(*a*)


Berlin
Mitte
Ost-Berlin
Berlin-Friedrichstadt
Friedrichshain
Samariterviertel
Berlin-Fennpfuhl
Volkspark Friedrichshain
Lichtenberg
Tiergarten
Alt-Moabit
Invalidenstraße
Torstraße
Friedrichstraße
Unter den Linden
Straße des 17. Juni
Tiergartenstraße
Holzmarktstraße
Mühlenstraße
Warschauer Str
Petersburger Str
Landsberger Allee
Storkower Str
Möllendorffstraße
Karl-Marx-Allee
Frankfurter Allee
Boxhagener Str
Kurfürstenstraße
Lützowufer
Mollstraße

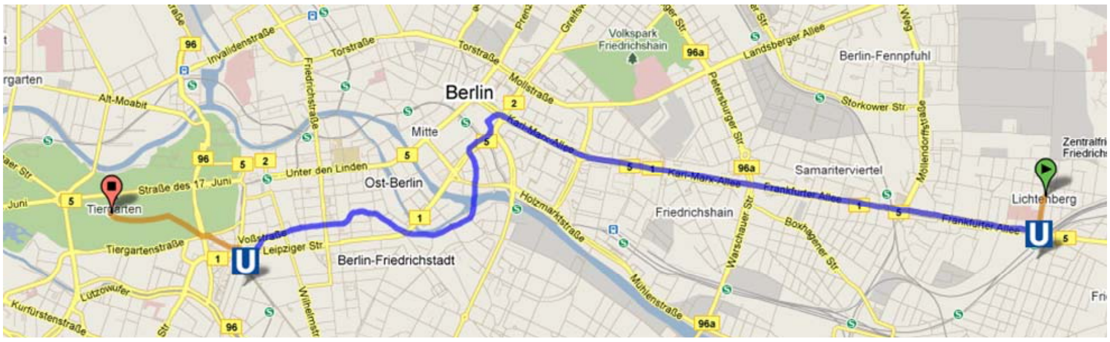
(*b*)


Berlin
Mitte
Ost-Berlin
Berlin-Friedrichstadt
Friedrichshain
Samariterviertel
Berlin-Fennpfuhl
Volkspark Friedrichshain
Lichtenberg
Tiergarten
Alt-Moabit
Invalidenstraße
Torstraße
Friedrichstraße
Unter den Linden
Straße des 17. Juni
Tiergartenstraße
Voßstraße
Leipziger Str
Holzmarktstraße
Mühlenstraße
Warschauer Str
Petersburger Str
Landsberger Allee
Storkower Str
Möllendorffstraße
Karl-Marx-Allee
Frankfurter Allee
Boxhagener Str
Kurfürstenstraße
Lützowufer
Mollstraße

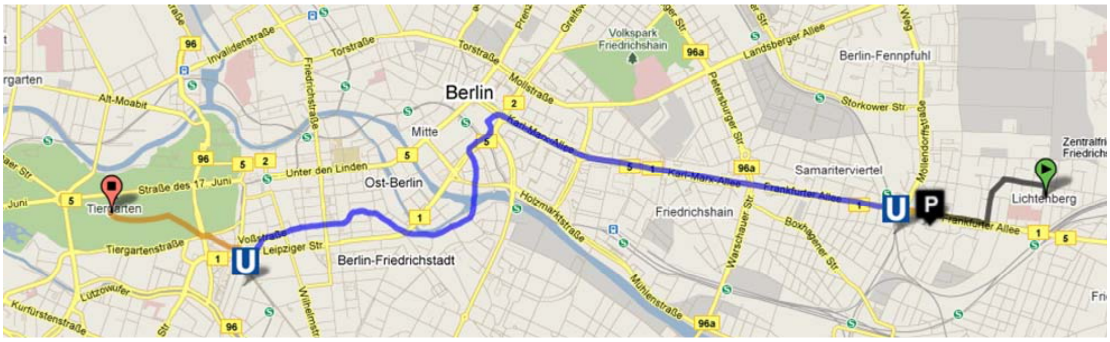
(*c*)


Berlin
Mitte
Ost-Berlin
Berlin-Friedrichstadt
Friedrichshain
Samariterviertel
Berlin-Fennpfuhl
Volkspark Friedrichshain
Lichtenberg
Tiergarten
Alt-Moabit
Invalidenstraße
Torstraße
Friedrichstraße
Unter den Linden
Straße des 17. Juni
Tiergartenstraße
Voßstraße
Leipziger Str
Holzmarktstraße
Mühlenstraße
Warschauer Str
Petersburger Str
Landsberger Allee
Storkower Str
Möllendorffstraße
Karl-Marx-Allee
Frankfurter Allee
Boxhagener Str
Kurfürstenstraße
Lützowufer
Mollstraße

(*d*)


Berlin
Mitte
Ost-Berlin
Berlin-Friedrichstadt
Friedrichshain
Samariterviertel
Berlin-Fennpfuhl
Volkspark Friedrichshain
Lichtenberg
Unter den Linden
Straße des 17. Juni
Leipziger Str.
Karl-Marx-Allee
Frankfurter Allee
Holzmarktstraße
Mühlenstraße
Torstraße
Invalidenstraße
Alt-Moabit
Tiergartenstraße

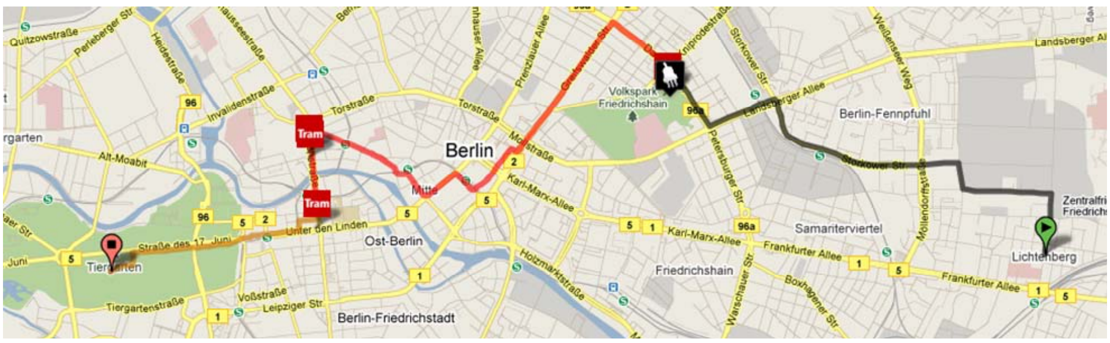
(*e*)


Berlin
Mitte
Ost-Berlin
Berlin-Friedrichstadt
Friedrichshain
Samariterviertel
Berlin-Fennpfuhl
Volkspark Friedrichshain
Lichtenberg
Unter den Linden
Straße des 17. Juni
Leipziger Str.
Karl-Marx-Allee
Frankfurter Allee
Holzmarktstraße
Mühlenstraße
Torstraße
Invalidenstraße
Alt-Moabit
Tiergartenstraße

(*f*)

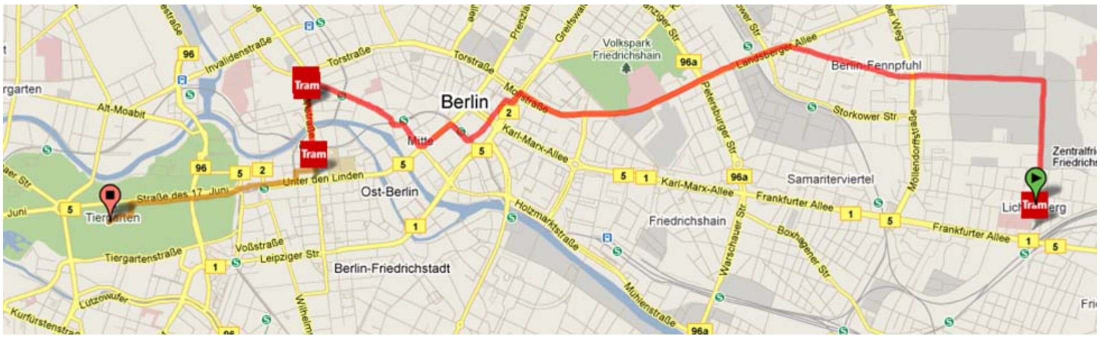

Berlin
Mitte
Ost-Berlin
Berlin-Friedrichstadt
Friedrichshain
Samariterviertel
Berlin-Fennpfuhl
Volkspark Friedrichshain
Lichtenberg
Unter den Linden
Straße des 17. Juni
Leipziger Str.
Karl-Marx-Allee
Frankfurter Allee
Holzmarktstraße
Mühlenstraße
Torstraße
Invalidenstraße
Alt-Moabit
Tiergartenstraße

(*g*)

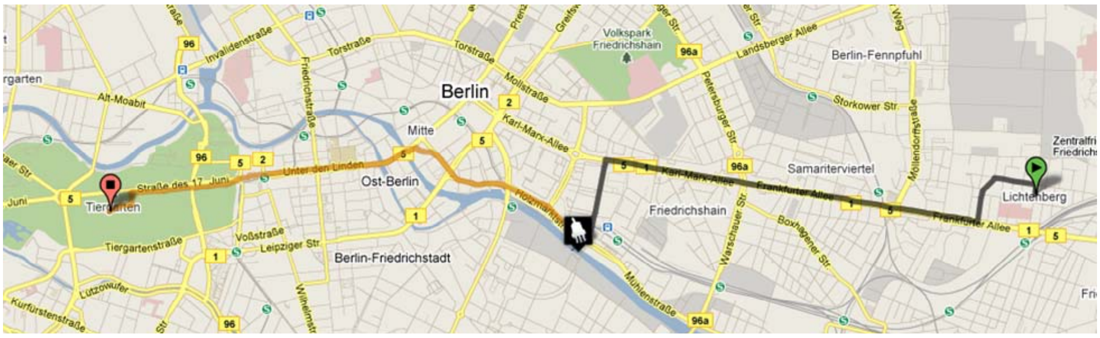

Tram
Tram
Berlin
Mitte
Ost-Berlin
Berlin-Friedrichstadt
Friedrichshain
Samariterviertel
Berlin-Fennpfuhl
Volkspark Friedrichshain
Lichtenberg
Unter den Linden
Straße des 17. Juni
Leipziger Str.
Karl-Marx-Allee
Frankfurter Allee
Holzmarktstraße
Torstraße
Invalidenstraße
Alt-Moabit
Tiergartenstraße
Prenzlauer Allee
Landsberger Allee

(*h*)

(*i*)



(*j*)



(*k*)

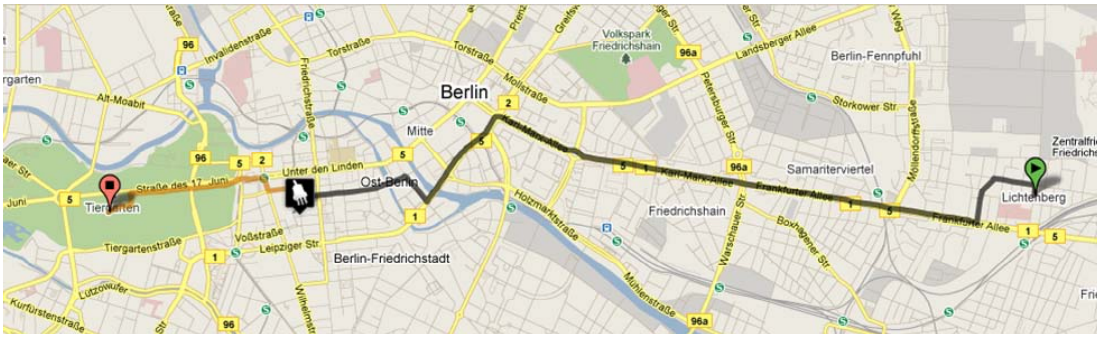
(*l*)


Invalidenstraße
Torstraße
Alt-Moabit
Berlin
Mitte
Unter den Linden
Ost-Berlin
Straße des 17. Juni
Tiergarten
Voßstraße
Leipziger Str.
Berlin-Friedrichstadt
Tiergartenstraße
Lützowufer
Kurfürstenstraße
Volkspark Friedrichshain
Landsberger Allee
Berlin-Fennpfuhl
Storkower Str.
Samariterviertel
Friedrichshain
Karl-Marx-Allee
Frankfurter Allee
Lichtenberg
Holzmarktstraße
Mühlenstraße

(*m*)


Invalidenstraße
Torstraße
Alt-Moabit
Berlin
Mitte
Unter den Linden
Ost-Berlin
Straße des 17. Juni
Tiergarten
Voßstraße
Leipziger Str.
Berlin-Friedrichstadt
Tiergartenstraße
Lützowufer
Kurfürstenstraße
Volkspark Friedrichshain
Landsberger Allee
Berlin-Fennpfuhl
Storkower Str.
Samariterviertel
Friedrichshain
Karl-Marx-Allee
Frankfurter Allee
Lichtenberg
Holzmarktstraße
Mühlenstraße

(*n*)


Invalidenstraße
Torstraße
Alt-Moabit
Berlin
Mitte
Unter den Linden
Ost-Berlin
Straße des 17. Juni
Tiergarten
Voßstraße
Leipziger Str.
Berlin-Friedrichstadt
Tiergartenstraße
Lützowufer
Kurfürstenstraße
Volkspark Friedrichshain
Landsberger Allee
Berlin-Fennpfuhl
Storkower Str.
Samariterviertel
Friedrichshain
Karl-Marx-Allee
Frankfurter Allee
Lichtenberg
Holzmarktstraße
Mühlenstraße

(*o*)


Invalidenstraße
Torstraße
Alt-Moabit
Berlin
Mitte
Unter den Linden
Ost-Berlin
Straße des 17. Juni
Tiergarten
Voßstraße
Leipziger Str.
Berlin-Friedrichstadt
Tiergartenstraße
Lützowufer
Kurfürstenstraße
Volkspark Friedrichshain
Landsberger Allee
Berlin-Fennpfuhl
Storkower Str.
Samariterviertel
Friedrichshain
Karl-Marx-Allee
Frankfurter Allee
Lichtenberg
Holzmarktstraße
Mühlenstraße

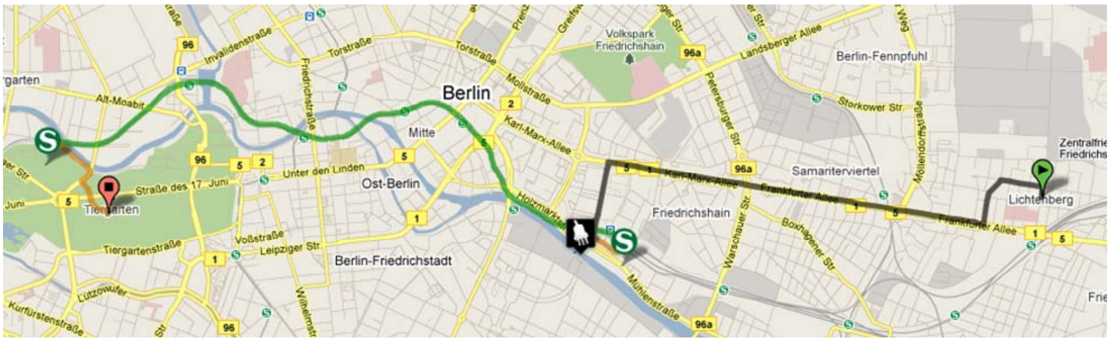
(*p*)


Berlin
Mitte
Ost-Berlin
Berlin-Friedrichstadt
Friedrichshain
Samariterviertel
Berlin-Fennpfuhl
Volkspark Friedrichshain
Unter den Linden
Straße des 17. Juni
Karl-Marx-Allee
Frankfurter Allee
Lichtenberg

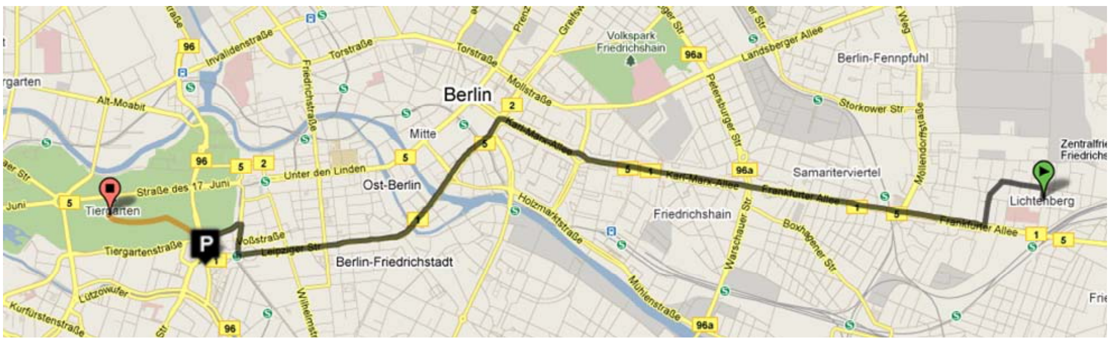
(*q*)


Berlin
Mitte
Ost-Berlin
Berlin-Friedrichstadt
Friedrichshain
Samariterviertel
Berlin-Fennpfuhl
Volkspark Friedrichshain
Unter den Linden
Straße des 17. Juni
Karl-Marx-Allee
Frankfurter Allee

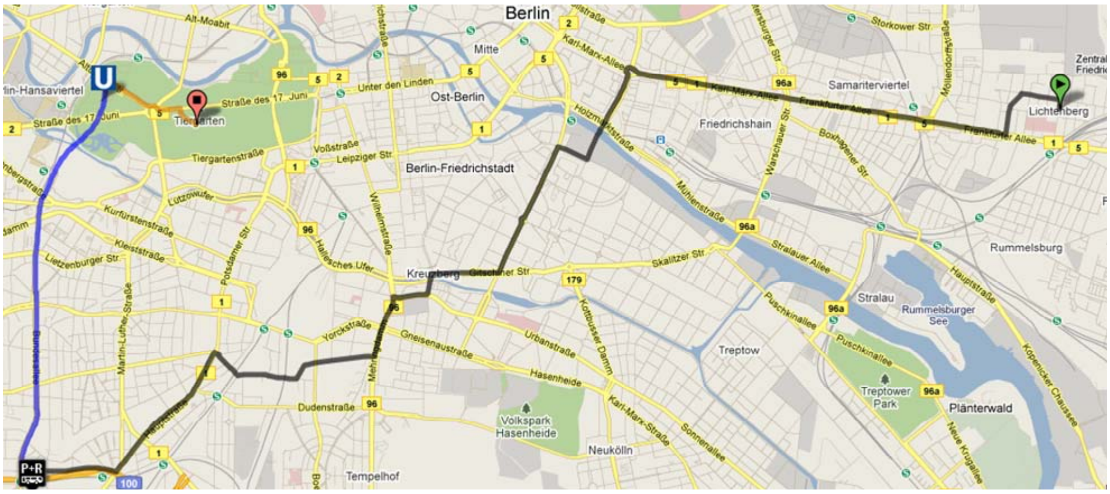
(*r*)


Berlin
Mitte
Ost-Berlin
Berlin-Friedrichstadt
Friedrichshain
Samariterviertel
Berlin-Fennpfuhl
Volkspark Friedrichshain
Unter den Linden
Straße des 17. Juni
Karl-Marx-Allee
Frankfurter Allee
Lichtenberg

(*s*)


Berlin
Mitte
Ost-Berlin
Berlin-Friedrichstadt
Friedrichshain
Samariterviertel
Berlin-Fennpfuhl
Volkspark Friedrichshain
Unter den Linden
Straße des 17. Juni
Frankfurter Allee
Lichtenberg

(*t*)



(*u*)

Figure 5.9 Multimodal route planning results of the Berlin scenario.

As it shows in Table 5.9 and Figure 5.9, when the electronic car has enough battery to be driven to a parking lot or a charging station near the destination (like the cases of option 1, 2 and 8), the faster routes will be produced, e.g. the route shown in Figure 5.9(*t*) which will take ca. 27 minutes and 11.24 km where 12 minutes and 1.06 km for walking. Furthermore, if the public transit system is taken into account when the driving distance is very limit (like the case of option 6), the fastest way is: drive to the charging station Friedrichstraße, leave the car there in charge, take the underground train from Stadtmitte station to Friedrichstraße station, change to suburban train and go to Bellevue station, and then walk to the destination (Figure 5.9(*m*)). This multimodal route will take ca. 35 minutes and 14.01 km where 15 minutes and 1.31 km for walking.

Chapter 6

Conclusions and Outlook

6.1 Conclusions

Facing the increasing demand on intelligent multimodal navigation services the author has set the problem of multimodal route planning as the research focus in this thesis work. The goal is to find reasonable and optimal paths in the context of multimodal transportation networks involving motorized roads, pedestrian ways, underground, suburban, and tram lines. Four main tasks have been successfully performed in order to reach the goal:

- **Representation of multimodal transportation networks.** This task is divided into the modeling of 1). the transportation networks; and 2). the spots where people can change from one mode to another. The weighted digraph data structure consisting of vertices and edges proves sufficient to express the physical networks. Each network of one specific mode is modeled as a mode graph. The modeling processes of different mode graphs reveal some differences, e.g. the traveling time of a car on a road segment is obtained from historical statistics data, while this value of a public transit link is fixed in an official time table. On the other hand, the concept of Switch Point is adopted to describe the key places where mode-switching actions can happen, e.g. parking areas, park and ride lots, public transit stations, etc. With the support of a series of auxiliary concepts and structures, the proposed concept Switch Point in this work is flexible enough to include time-dependent information and attach user-defined constraints. Furthermore, Switch Point is case-dependent, which can be filtered by providing some Switch Conditions. This characteristic makes it possible to unite some mode graphs into one single graph for a concrete multimodal routing task. The combination process is carried out by generating a new edge directing from the "switch-from" vertex to the "switch-to" vertex within one Switch Point, just like the action of plugging into a power socket. Therefore, the final data structure is named Plug-and-Play Multimodal Graph Set (PnPMMGS). This modeling method separates the static physical networks from the dynamic mode-switching actions, and provides possibility and flexibility for the multimodal routing algorithms to find all the feasible optimal paths.

- **Formalization of multimodal route-planning problem.** The route-planning problem is categorized into two types depending on whether the mode sequence is known or not. The formalization is derived from the definition of shortest path problem, and strongly relies on the data model of a multimodal transportation network. For both types, the author addresses their single-source and point-to-point variations. Furthermore, the time-dependent versions of the problems are depicted as well. It turns out that in the Type II problem, the input Switch Condition Matrix can be interpreted as the adjacency matrix expression of another digraph whose vertices and edges are the involved modes and Switch Condition between mode pairs respectively. The definitions in the thesis are generic in nature, which means the multimodal shortest path problem is not limited in the transportation field. These descriptions have made full use of the results from the modeling task, abstracted the problem mathematically, and paved the way for the design of multimodal route-planning algorithms.

- **Design and analysis of multimodal route-planning algorithms.** With the aim to solve the two types of multimodal shortest path problems along with their variations and time-dependent versions, different algorithms have been proposed. For the Type I problem, an algorithmic framework Algorithm I with the key step of MULTIMODALINITIALIZE has been designed. Algorithm I can be implemented by substituting any traditional label-setting or label-correcting method with the initialization step removed into the ITERATIVERELAX routine. If there are Switch Constraints defined by users in the input, Algorithm I can handle it by embedding these constraints in the routing library during the runtime, and filter out all the semi-finished multimodal paths that do not meet the conditions. In the Type II problem where the optimal mode sequence should be calculated by the algorithm, the SCM-PLUG operation on the PnPMMGS is conducted first, and then the optimal path query with classic shortest path algorithms can be performed. The computational complexities of the solutions to both types of problem depend on the concrete labeling method they adopt, which means the multimodal shortest path algorithms in this work can benefit from the performance improvements for mono-modal algorithms. Looking into the details of the algorithms, the author found an interesting fact that the two solutions are equivalent if the Switch Condition list Φ is expressed in the form of a matrix SCM_{Φ}. To explain this fact, the author firstly proves the effectiveness of the Algorithm I based on two previously proposed theories, and then proposes a new theory stating the equivalence relation. The feasibility and validity with regard to the mode or mode combination are determined by a standalone component based on domain-specific knowledge and the context. In the filed of urban transportation area, the reasonable mode list is determined by an inferring process based on a series of pre-defined rules.

- **Evaluation and demonstration of the proposed approach.** In the experiments, the algorithmic framework was implemented by adopting a label-correcting method called TWO-Q that has a convincing performance in transportation network. The multimodal route-planning algorithm has been tested on real navigation datasets with desirable performance. This result convinced the author to implement the modeling and path-finding approaches in a prototype system which is web-based and available on the Internet. With the help of this system, two groups of case studies have been conducted for Munich and Berlin respectively. The system is able to create different routing plans along with the corresponding optimal paths that match users' diverse preferences. Worthwhile to mention is its unique capability of providing both mono-modal (e.g. by car, walking) and multimodal (e.g. driving and walking, driving and taking public transit) routing plans that satisfy user's options, which is missing in any other existing route planning systems or services. In short, the extensive experimental results have confirmed the robustness and performance of the system, and the effectiveness and feasibility of the underlying theories.

To sum up, multimodal route planning is one of the core supporting techniques of intelligent multimodal navigation service intended to substantially enhance human mobility. There are three main contributions from this research:

- **A novel modeling strategy that decouples the mode-switching action from the graph construction process of the physical networks.** This strategy prevents the complexities on Switch Point from being brought into the mode graph structures. If every Switch Point is compared to a pair of plug and socket, each mode graph looks like a component with some attached plugs and sockets, which makes these mode graphs pluggable to each other under some special conditions. This concept is termed as “Plug-and-Play Multimodal Graph Set”.
- **Generic solutions to route-planning problems.** Although route-planning problems originate from transportation field, they find affinities in other fields where there are usually some domain-specific restrictions on the mode sequence. The validation of a mode sequence is the responsibility of a rule-based inferring process which has been decoupled from the modeling method and path-finding algorithms. The rules set to determine the feasible mode sequence in urban transportation area is elaborated in the work.
- **A web-based prototype system that demonstrates the theories and methodologies proposed in the thesis.** The development of the system is based on the extensive evaluations with regard to the feasibility, reliability and performance of the proposed approach. According to the case studies for real-world scenarios, the multimodal routing results provided by the system can help people a lot to plan optimal paths in complex urban transportation networks.

6.2 Future work

There is still a large unexplored area in the research field of multimodal route planning. Several interesting topics on the basis of this thesis work need an in-depth investigation in the future.

- **Dynamic multimodal route planning.** The time-dependent issue in the multimodal route planning is considered in this work in terms of time-variant cost on Switch Points that gives the sense of the waiting time. In reality, however, the dynamics of multimodal route planning is not only reflected in mode-switching actions, but in some attributes of the physical networks, e.g. the current speed of a moving object, the real-time traffic information, etc. Taking such sort of information into account may lead to more accurate and realistic routing results and requires therefore further improvement of the graph data structure, the integration of mode graphs and Switch Points, and the path-finding algorithms.
- **Multimodal route planning for long-distance trips.** The test beds in this work are

only small navigation datasets compared with state- or even continent-level datasets. When planning a multimodal route for long-distance trips, more complicated issues have to be considered: 1). the physical networks are so large that they have to be re-organized by means of smart horizontal and vertical partitioning techniques; 2). for international long-distance trips, more transportation modes (e.g. intercity trains, ships and flights) and Switch Types (e.g. train stations, passenger transport wharves and airports) may be involved. This may require further innovative strategies.

- **Multimodal route planning based on Volunteered Geographic Information (VGI).** Aside from the commercial navigation dataset, the emerging VGI datasets (e.g. OpenStreetMap) have been growing fast in recent years. The data quality of such VGI datasets, from the route-planning point of view, is rather high and will become even better. In some urban areas, the OpenStreetMap data is even more detailed and up-to-date than some commercial products. Furthermore, VGI datasets have many usable features (e.g. pedestrian and cycling ways, parking lots, etc.) for multimodal route planning, and cannot be provided in the conventional navigation datasets dedicated to cars. Making full use of VGI datasets in the modeling and path-finding approach proposed in this thesis is an emerging issue for the subsequent work.

- **Smart visualization of multimodal routing results.** A multimodal route contains rich information including several path segments belonging to different modes, the descriptions of mode-switching actions, and diverse costs. How to visualize this result properly on a map is an interesting cartographic task. One possible way, for instance, is to visualize the multimodal route with some emphasis (e.g. by applying lens effect) on all the involved Switch Points along this route.

- **Multimodal route planning in other complex networks.** Besides the transportation network, there are other networks that have the similar multimodal property. The social network is such an example. Different people belong to different social sub-networks with each being comparable to a mode. Two people can share one or many modes or they can be reached each other across several different modes possibly at different costs. And the person who is at the tangent point of two neighboring sub-networks can be regarded as Switch Point. Optimal paths that satisfy one or many constraints in the social network are required in order to get right mindsets involved who can collectively perform a highly complex task. Figure 6.1 shows an example of multimodal social relationship networks in which the involved people represented by vertices are connected by edges with thicknesses indicating the familiarities between them. There is a relation between *Carpenter A* and *Judge F* across three different modes of relationship networks: *Carpenter A*, *Handyman B*, *Manager C*, *Chef D*, *Policeman E* and *Judge F* where the *Manager C* and *Chef D* are two “Switch Points”.

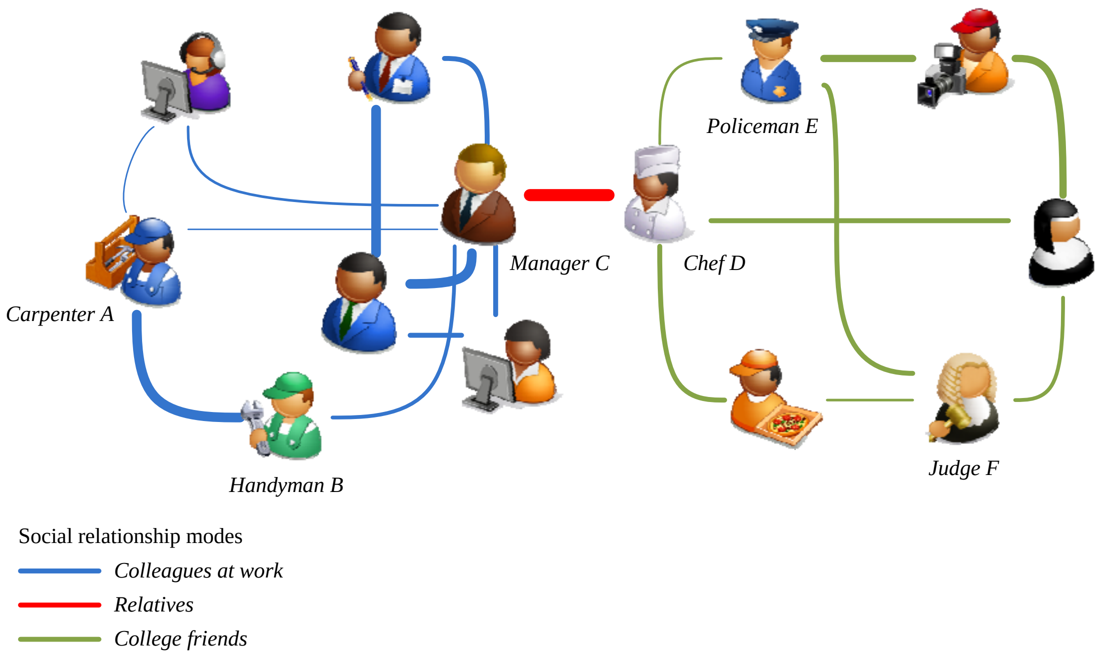


Figure 6.1 Multimodal social relationship networks.

Bibliography

ABBASPOUR, R. A. and SAMADZADEGAN, F. 2009. A Solution for Time-Dependent Multimodal Shortest Path Problem. *Journal of Applied Sciences,* 9(21), 3804-3812.

AHUJA, R. K., MAGNANTI, T. L. and ORLIN, J. B., 1993. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall.

AHUJA, R. K., *et al.* 2003. Dynamic Shortest Paths Minimizing Travel Times and Costs. *Networks,* 41(4), 197-205.

ARIEL, O. and RAPHAEL, R. 1990. Shortest-Path and Minimum-Delay Algorithms in Networks with Time-Dependent Edge-Length. *Journal of the ACM,* 37(3), 607-625.

ARIKAWA, M., KONOMI, S. and OHNISHI, K. 2007. Navitime: Supporting Pedestrian Navigation in the Real World. *IEEE Pervasive Computing,* 6(3), 21-29.

AYED, H., *et al.* 2010. Solving Time-Dependent Multimodal Transport Problems Using a Transfer Graph Model. *Computers & Industrial Engineering,* In Press, Corrected Proof.

AYED, H., *et al.*, 2008. Transfer Graph Approach for Multimodal Transport Problems. *Modelling, Computation and Optimization in Information Systems and Management Sciences.* Springer Berlin Heidelberg, 538-547.

AYED, H., *et al.*, 2009. Hybrid Algorithm for Solving a Multimodal Transport Problems Using a Transfer Graph Model. *Proceedings of the Second international conference on Global Information Infrastructure Symposium.* Hammamet, Tunisia: IEEE Press.

BARRETT, C., *et al.*, 2007. *Label Constrained Shortest Path Algorithms: An Experimental Evaluation Using Transportation Networks.*

BARRETT, C., *et al.*, 2008. Engineering Label-Constrained Shortest-Path Algorithms. *Algorithmic Aspects in Information and Management.* 27–37.

BARRETT, C., *et al.*, 2002. Classical and Contemporary Shortest Path Problems in Road Networks: Implementation and Experimental Analysis of the TRANSIMS Router. *Algorithms — Esa 2002.* 313-319.

BARRETT, C., JACOB, R. and MARATHE, M. 2000. Formal-Language-Constrained Path Problems. *SIAM Journal on Computing,* 30(3), 809-837.

BAST, H., *et al.* 2007. Fast Routing in Road Networks with Transit Nodes. *Science,* 316(5824), 566.

BAUER, R., *et al.* 2010. Combining Hierarchical and Goal-Directed Speed-up Techniques for Dijkstra's Algorithm. *Journal of Experimental Algorithmics (JEA),* 15, 2.1-2.31.

BAUER, R., DELLING, D. and WAGNER, D. 2011. Experimental Study of Speed up Techniques for Timetable Information Systems. *Networks,* 57(1), 38-52.

BAUMANN, D., TORDAY, A. and DUMONT, A.-G., The Importance of Computing Intermodal Roundtrips in Multimodal Guidance Systems. ed. *Proceedings of the 4th STRC Swiss Transport Research Conference*, 2004 2004 Monte Verità, Ascona, Switzerland.

BECKER, T., NAGEL, C. and KOLBE, T. H., 2009. A Multilayered Space-Event Model for Navigation in Indoor Spaces. *3d Geo-Information Sciences.* 61-77.

BELLMAN, R. 1958. On a Routing Problem. *Quarterly of Applied Mathematics*, 16(1), 87–90.

BERRY, J. W. and GOLDBERG, M. K. 1999. Path Optimization for Graph Partitioning Problems. *Discrete Appl. Math.*, 90(1-3), 27-50.

BERTSEKAS, D., GUERRIERO, F. and MUSMANNO, R. 1996. Parallel Asynchronous Label-Correcting Methods for Shortest Paths. *Journal of Optimization Theory and Applications*, 88(2), 297-320-320.

BERTSEKAS, D. P. 1993. A Simple and Fast Label Correcting Algorithm for Shortest Paths. *Networks*, 23(7), 703-709.

BIELLI, M., BOULMAKOUL, A. and MOUNCIF, H. 2006. Object Modeling and Path Computation for Multimodal Travel Systems. *European Journal of Operational Research*, 175(3), 1705–1730.

BMW, 2010. *Mini E Specifications* [online]. Available from: http://www.miniusa.com/minie-usa/pdf/MINI-E-spec-sheet.pdf.

BOARDMAN, B. S., *et al.*, Computer Assisted Routing of Intermodal Shipments. ed. *Proceedings of the 21st International Conference on Computers and Industrial Engineering*, 1997, 311–314.

BOILE, M. P., Intermodal Transportation Network Analysis-a GIS Application. ed. *Electrotechnical Conference, 2000. MELECON 2000. 10th Mediterranean*, 2000, 660-663.

BOUSQUET, A., 2009. Routing Strategies Minimizing Travel Times within Multimodal Transport Networks. *ECTRI-YRS09.* Torino, Italy.

BOUSQUET, A., SOPHIE, C. and NOUR-EDDIN, E. F., 2009. On the Adaptation of a Label-Setting Shortest Path Algorithm for One-Way and Two-Way Routing in Multimodal Urban Transport Networks. *International Network Optimization Conference.* Pisa, Italy.

BOUSSEDJRA, M., BLOCH, C. and EL MOUDNI, A., An Exact Method to Find the Intermodal Shortest Path (Isp). ed. *Proceedings of the IEEE International Conference on Networking, Sensing and Control*, 2004, 1075–1080.

BRUNTSCH, S. and REHRL, K., 2005. Vienna-Spirit Combining Intermodal Traveller Information with Onboard/Offboard Navigation Services. *The European Navigation Conference GNSS.* Munich, Germany.

CAR, A. and FRANK, A., 1994. Modelling a Hierarchy of Space Applied to Large Road Networks. *IGIS '94: Geographic Information Systems.* 15-24.

CAREY, N., 2005. *Establishing Pedestrian Walking Speeds* [online]. Available from: http://www.westernite.org/datacollectionfund/2005/psu_ped_summary.pdf.

CASHIN, R. D., LIU, P. and CHEN, J. 2002. Deploying the Advanced Public Transportation Systems

(Apts) Technologies in Los Angeles: A Summary and Analysis. *Journal of Advanced Transportation,* 36(1), 107-129.

CHANG, E., FLOROS, E. and ZILIASKOPOULOS, A., 2007. An Intermodal Time-Dependent Minimum Cost Path Algorithm. *Dynamic Fleet Management.* 113-132.

CHERKASSKY, B., GOLDBERG, A. and RADZIK, T. 1996. Shortest Paths Algorithms: Theory and Experimental Evaluation. *Mathematical Programming,* 73(2), 129–174.

CHIU, D. K. W., *et al.*, A Multi-Modal Agent Based Mobile Route Advisory System for Public Transport Network. ed. *38th Annual Hawaii International Conference on System Sciences* January 3-6 2005 Big Island, Hawaii, 92b-92b.

CHO, J., KIM, H. and CHOI, H. 2010. An Intermodal Transport Network Planning Algorithm Using Dynamic Programming—a Case Study: From Busan to Rotterdam in Intermodal Freight Routing. *Applied Intelligence.*

CHO, J., *et al.*, 2007. An Intermodal Transport Network Planning Algorithm Using Dynamic Programming. *New Trends in Applied Artificial Intelligence.* 1012-1021.

CORMEN, T. H., *et al.*, 2009. *Introduction to Algorithms.* 3rd ed.: MIT Press.

COUCKUYT, J. D., MCGRATH, T. S. and SUTANTO, H., 2006. *Multi-Modal Navigation System and Method.* United States application 11/058,057.

CRAINIC, T. G. and ROUSSEAU, J.-M. 1986. Multicommodity, Multimode Freight Transportation: A General Modeling and Algorithmic Framework for the Service Network Design Problem. *Transportation Research Part B: Methodological,* 20(3), 225-242.

DAVID, E. and MICHAEL, T. G., 2008. Studying (Non-Planar) Road Networks through an Algorithmic Lens. *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems.* Irvine, California: ACM.

DECHTER, R. and PEARL, J. 1985. Generalized Best-First Search Strategies and the Optimality of A*. *Journal of the ACM,* 32(3), 505-536.

DELLING, D., *et al.*, 2010. Chapter 9. Case Studies. *Algorithm Engineering.* Springer Berlin / Heidelberg, 389-445.

DELLING, D., PAJOR, T. and WAGNER, D., Engineering Time-Expanded Graphs for Faster Timetable Information. ed. *ATMOS 2008 - 8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems*, September 24 2008 Dagstuhl, Germany.

DELLING, D., PAJOR, T. and WAGNER, D., 2009a. Accelerating Multi-Modal Route Planning by Access-Nodes. *Algorithms - Esa 2009.* 587-598.

DELLING, D., *et al.*, 2009b. Engineering Route Planning Algorithms. *Algorithmics of Large and Complex Networks.* Springer-Verlag, 117-139.

DENNIS DE, C. 1983. Bidirectional Heuristic Search Again. *Journal of the ACM,* 30(1), 22-32.

DIESTEL, R., 2006. *Graph Theory.* New York: Springer-Verlag Berlin Heidelberg.

DIJKSTRA, E. W. 1959. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1(1), 269–271.

ENGINEER, F., 2001. Fast Shortest Path Algorithms for Large Road Networks. *36th Annual ORSNZ Conference.* University of Canterbury, Christchurch, New Zealand.

ESRI, 1998. *ESRI Shapefile Technical Description* [online]. Available from: http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf.

FAWCETT, J. and ROBINSON, P. 2000. Adaptive Routing for Road Traffic. *Computer Graphics and Applications, IEEE*, 20(3), 46-53.

FILIPOV, L. and DAVIDKOV, B., 2006. Development of Network Data Model for Urban Transportation System in GIS Environment for the City of Plovdiv, Bulgaria. *International Conference on Cartography and GIS.* Borovets, Bulgaria.

FLINSENBERG, I. C. M., 2004. *Route Planning Algorithms for Car Navigation.* (Doctor). Technische Universiteit Eindhoven.

FOHL, P., *et al.*, 1996. A Non-Planar, Lane-Based Navigable Data Model for ITS. *Seventh International Symposium on Spatial Data Handling.* Delft, 7B.17-17B.29.

FORD, L., 1956. *Network Flow Theory.* Santa Monica: The RAND Corporation.

FORD, L. and FULKERSON, D., 1962. *Flows in Networks.* Princeton University Press.

FOWLER, M., 2004. *Inversion of Control Containers and the Dependency Injection Pattern* [online]. Available from: http://martinfowler.com/articles/injection.html [Accessed Sep. 21 2010].

FRANK, A. U. 2008. Shortest Path in a Multi-Modal Transportation Network: Agent Simulation in a Product of Two State-Transition Networks. *Künstliche Intelligenz*, 3, 14–18.

FRIDRICH, M., 1998. A Multi-Modal Transport Model for Integrated Planning. *8th World Conference on Transport Research.* Antwerp, Belgium.

FU, L., SUN, D. and RILETT, L. R. 2006. Heuristic Shortest Path Algorithms for Transportation Applications: State of the Art. *Computers & Operations Research*, 33(11), 3324-3343.

GALVEZ-FERNANDEZ, C., *et al.* 2009. Distributed Approach for Solving Time-Dependent Problems in Multimodal Transport Networks. *Advances in Operations Research*, 2009, 15.

GARAIX, T., *et al.* 2010. Vehicle Routing Problems with Alternative Paths: An Application to on-Demand Transportation. *European Journal of Operational Research*, 204(1), 62-75.

GEISBERGER, R., *et al.*, 2008. Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks. *Experimental Algorithms.* 319-333.

GERAETS, F., *et al.*, 2007. Timetable Information: Models and Algorithms. *Algorithmic Methods for Railway Optimization.* Springer Berlin / Heidelberg, 67-90.

GHIANI, G., *et al.* 2003. Real-Time Vehicle Routing: Solution Concepts, Algorithms and Parallel Computing Strategies. *European Journal of Operational Research*, 151(1), 1-11.

GLOVER, F., GLOVER, R. and KLINGMAN, D. 1986. The Threshold Shortest Path Algorithm. *Networks*, 14(1), 12-37.

GOLDBERG, A. V. and HARRELSON, C., 2005. Computing the Shortest Path: A* Search Meets Graph Theory. *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms.* Vancouver, British Columbia: Society for Industrial and Applied Mathematics.

GOLDBERG, A. V. and RADZIK, T. 1993. A Heuristic Improvement of the Bellman-Ford Algorithm. *Applied Mathematics Letters,* 6(3), 3-6.

GOODCHILD, M. F. 1998. Geographic Information Systems and Disaggregate Transportation Modeling. *Geographical Systems,* 5, 19-44.

GOODCHILD, M. F. 2000. GIS and Transportation: Status and Challenges. *Geoinformatica,* 4(2), 127-139-139.

GRÄBENER, T., BERRO, A. and DUTHEN, Y., 2009. What Algorithms for Urban Routing on Mobile Devices? *9th Workshop on Multimedia Metadata (WMM'09).* Toulouse, France.

GUTIÉRREZ, E. and MEDAGLIA, A. 2008. Labeling Algorithm for the Shortest Path Problem with Turn Prohibitions with Application to Large-Scale Road Networks. *Annals of Operations Research,* 157(1), 169-182.

GUZOLEK, J. and KOCH, E., Real-Time Route Planning in Road Networks. ed. *Vehicle Navigation and Information Systems Conference,* 11-13 Sep 1989, 165-169.

HAHNE, F., NOWAK, C. and AMBROSI, K., 2008. Acceleration of the A*-Algorithm for the Shortest Path Problem in Digital Road Maps. *Operations Research Proceedings 2007.* 455-460.

HANIF, D. S., ANTOINE, G. H. and SASIKUL, K. 2003. Time-Dependent, Label-Constrained Shortest Path Problems with Applications. *Transportation Science,* 37(3), 278-293.

HANSON, S., 1995. Getting There: Urban Transportation in Context. *In:* HANSON, S. ed. *The Geography of Urban Transportation.* New York: Guilford, 3-25.

HART, P. E., NILSSON, N. J. and RAPHAEL, B. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics,* 4(2), 100-107.

HASSELBERG, S., 2000. *Some Results on Heuristical Algorithms for Shortest Path Problems in Large Road Networks.* Doctor thesis (Doctor). Universität zu Köln.

HERMANN, K. and GERHARD, K. 1997. Bidirectional Heuristic Search Reconsidered. *Journal of Artificial Intelligence Research,* 7(1), 283-317.

HOCHMAIR, H., 2004. Decision Support for Bicycle Route Planning in Urban Environments. *Proceedings of the 7th AGILE Conference on Geographic Information Science.* Heraklion, Greece: Crete University Press.

HOEL, E. G., HENG, W.-L. and HONEYCUTT, D., 2005. High Performance Multimodal Networks. *Advances in Spatial and Temporal Databases.* Berlin / Heidelberg: Springer, 308–327.

HOLZER, M., 2003. *Hierarchical Speed-up Techniques for Shortest-Path Algorithms.* Diplomarbeit

(Diplom). Universität Konstanz.

HOLZER, M., SCHULZ, F. and WAGNER, D. 2009. Engineering Multilevel Overlay Graphs for Shortest-Path Queries. *Journal of Experimental Algorithmics (JEA),* 13, 2.5-2.26.

HOLZER, M., *et al.* 2005. Combining Speed-up Techniques for Shortest-Path Computations. *Journal of Experimental Algorithmics (JEA),* 10, 2.5.

HONG, K. L., YIP, C. W. and BRIAN, M. 2005. Passenger Route Guidance System for Multi-Modal Transit Networks. *Journal of Advanced Transportation,* 39(3), 271-288.

HORN, M. E. T. 2002. Multi-Modal and Demand-Responsive Passenger Transport Systems: A Modelling Framework with Embedded Control Systems. *Transportation Research Part A: Policy and Practice,* 36(2), 167-188.

HUANG, B., WU, Q. and ZHAN, F. B. 2007. A Shortest Path Algorithm with Novel Heuristics for Dynamic Transportation Networks. *International Journal of Geographical Information Science,* 21(6), 625-644.

HUANG, R. and PENG, Z. R. 2008. A Spatiotemporal Data Model for Dynamic Transit Networks. *International Journal of Geographical Information Science,* 22(5), 527-545.

HUANG, Y.-W., JING, N. and RUNDENSTEINER, E. A., 1996. Effective Graph Clustering for Path Queries in Digital Map Databases. *Proceedings of the fifth international conference on Information and knowledge management.* Rockville, Maryland, United States: ACM.

JACOB, R., MARATHE, M. V. and NAGEL, K., 1999. *A Computational Study of Routing Algorithms for Realistic Transportation Networks.* Los Alamos National Laboratory.

JAGADEESH, G. R., SRIKANTHAN, T. and QUEK, K. H. 2002. Heuristic Techniques for Accelerating Hierarchical Routing on Road Networks. *IEEE Transactions on Intelligent Transportation Systems,* 3(4), 301-309.

JIANG, B. and YAO, X., 2010. Geospatial Analysis and Modeling of Urban Structure and Dynamics: An Overview. *Geospatial Analysis and Modelling of Urban Structure and Dynamics.* 3-11.

JING, N., HUANG, Y.-W. and RUNDENSTEINER, E. A. 1998. Hierarchical Encoded Path Views for Path Query Processing: An Optimal Model and Its Performance Evaluation. *IEEE Transactions on Knowledge and Data Engineering,* 10(3), 409-432.

JOEL, B., *et al.*, 2009. A Data Model for Trip Planning in Multimodal Transportation Systems. *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology.* Saint Petersburg, Russia: ACM.

JOYNER, D., NGUYEN, M. V. and COHEN, N., 2010. *Algorithmic Graph Theory.* 0.4 ed.: GNU-FDL.

KELLER, *et al.*, 2001. *Struktur Und Bewertung Des Verkehrsinformationsnetzwerks Bayerninfo.* Hamburg, ALLEMAGNE: Deutscher Verkehrs.

KENYON, S. and LYONS, G. 2003. The Value of Integrated Multimodal Traveller Information and Its Potential Contribution to Modal Change. *Transportation Research Part F: Traffic Psychology and Behaviour,* 6(1), 1-21.

KHEIRIKHARZAR, M., 2010. Shortest Path Algorithm in Multimodal Networks for Optimization of Public Transport. *FIG Congress 2010 Facing The Challenges- Building the Capacity.* Sydney, Australia.

KIM, D. and BARNHART, C. 1997. Multimodal Express Shipment Service Design: Models and Algorithms. *Computers & Industrial Engineering,* 33(3-4), 685-688.

KIM, D., *et al.* 1999. Multimodal Express Package Delivery: A Service Network Design Application. *Transportation Science,* 33(4), 391-407.

KLUNDER, G. A. and POST, H. N. 2006. The Shortest Path Problem on Large-Scale Real-Road Networks. *Networks,* 48(4), 182-194.

LENIE, S. and DENNIS DE, C. 1977. An Improved Bidirectional Heuristic Search Algorithm. *Journal of the ACM,* 24(2), 177-191.

LETCHNER, J., KRUMM, J. and HORVITZ, E., Trip Router with Individualized Preferences (Trip): Incorporating Personalization into Route Planning. ed. *18th Conference on Innovative Applications of Artificial Intelligence*, 2006.

LI, Q. and KURT, C. E., GIS-Based Itinerary Planning System for Multimodal and Fixed-Route Transit Network. ed. *Proceedings of the MID-Continent Transportation Symposium*, 2000, 47-50.

LI, Y. and TSUKAGUCHI, H. 2005. Relationships between Network Topology and Pedestrian Route Choice Behavior. *Journal of the Eastern Asia Society for Transportation Studies,* 6, 241-248.

LIN, C., YANG, Z. and GONG, B., Multimodal Traffic Information Service System with K-Multimodal Shortest Path Algorithm. ed. *Intelligent Computation Technology and Automation, 2009. ICICTA '09. Second International Conference on*, 2009, 652-655.

LIU, C.-L., Best-Path Planning for Public Transportation Systems. ed. *Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems*, 2002, 834-839.

LIU, C.-L., *et al.*, Path-Planning Algorithms for Public Transportation Systems. ed. *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, 2001, 1061-1066.

LIU, L. and MENG, L. 2009. Algorithms of Multi-Modal Route Planning Based on the Concept of Switch Point. *Photogrammetrie Fernerkundung Geoinformation,* 2009(5), 431-444.

LIU, L. and MENG, L., 2010. Applying Traditional Speed-up Techniques into Switch-Point-Based Multimodal Shortest Paths Algorithms. *Sixth international conference on Geographic Information Science.* Zurich, Switzerland.

LONGLEY, P. A., *et al.*, 2010. *Geographic Information Systems and Science.* 3rd ed. Hoboken: John Wiley & Sons.

LOPE, P. J., 2009. Assessment of Time Use Constraints in Multimodal Trips. Madrid Case Study. *EC-TRI-YRS09.* Torino, Italy

LOZANO, A. and STORCHI, G. 2001. Shortest Viable Path Algorithm in Multimodal Networks. *Transportation Research Part A: Policy and Practice,* 35(3), 225–241.

LOZANO, A. and STORCHI, G. 2002. Shortest Viable Hyperpath in Multimodal Networks. *Transportation Research Part B: Methodological*, 36(10), 853–874.

LUBY, M. and RAGDE, P. 1989. A Bidirectional Shortest-Path Algorithm with Good Average-Case Behavior. *Algorithmica*, 4(1), 551-567.

MACHARIS, C. and BONTEKONING, Y. M. 2004. Opportunities for Operational Research in Intermodal Freight Transport Research: A Review. *European Journal of Operational Research*, 153(2), 400-416.

MALM, A., 2009. *Personal Navigation Devices.* Gothenburg: Berg Insight.

MANDLOI, D. and THILL, J.-C., 2010. Object-Oriented Data Modeling of an Indoor/Outdoor Urban Transportation Network and Route Planning Analysis. *Geospatial Analysis and Modelling of Urban Structure and Dynamics.* 197-220.

MARATHE, M. V. 2002. Routing in Very Large Multi-Modal Time Dependent Networks: Theory and Practice. *Electronic Notes in Theoretical Computer Science*, 66(6), 1-7.

MARIA, F., 2005. Navigation and Multimodal Transportation with Easytransport. *In:* ALEX, D. ed. *IEEE Intelligent Systems.* 54-61.

MARTIN-ANDERSON, B., 2007. *Graphserver* [online]. Available from: http://bmander.github.com/graphserver/.

MAY, A., *et al.* 2003. Pedestrian Navigation Aids: Information Requirements and Design Implications. *Personal and Ubiquitous Computing*, 7(6), 331-338.

MENG, F. H., *et al.*, A Multi-Criteria, Multi-Modal Passenger Route Advisory System. ed. *Proceedings of the IES-CTR International Symposium*, 1999 Singapore.

MODESTI, P. and SCIOMACHEN, A. 1998. A Utility Measure for Finding Multiobjective Shortest Paths in Urban Multimodal Transportation Networks. *European Journal of Operational Research*, 111(3), 495–508.

MÖHRING, R. H., *et al.* 2006. Partitioning Graphs to Speedup Dijkstra's Algorithm. *Journal of Experimental Algorithmics (JEA)*, 11, 2.8.

MOLIN, E. and VAN GELDER, M. 2008. Freeway Access to Public Transport: A Hierarchical Multimodal Choice Model. *Transportation Research Record: Journal of the Transportation Research Board*, 2076, 106-113.

MOUNCIF, H., BOULMAKOUL, A. and CHALA, M. 2006. Integrating GIS-Technology for Modelling Origin-Destination Trip in Multimodal Transportation Networks. *The International Arab Journal of Information Technology*, 3(3), 256-264.

NANNICINI, G., *et al.*, 2008. Bidirectional A* Search for Time-Dependent Fast Paths. *Experimental Algorithms.* Springer Berlin / Heidelberg, 334-346.

NEUHERZ, M., *et al.*, 2000. User Acceptance and Impacts of New Multimodal Traffic Information Services in Bayerninfo. *7th World Congress on Intelligent Transport Systems.* Turin, Italy.

NGUYEN, S., PALLOTTINO, S. and MALUCELLI, F. 2001. A Modeling Framework for Passenger Assignment on a Transport Network with Timetables. *Transportation Science,* 35(3), 238-249.

PAJOR, T., 2009. *Multi-Modal Route Planning.* Diplomarbeit (Diplom). Universität Karlsruhe (TH).

PALLOTTINO, S. 1984. Shortest-Path Methods: Complexity, Interrelations and New Propositions. *Networks,* 14(2), 257–267.

PALLOTTINO, S. and SCUTELLA, M. G., 1997. *Shortest Path Algorithms in Transportation Models: Classical and Innovative Aspects.* University of Pisa.

PAPE, U. 1974. Implementation and Efficiency of Moore-Algorithms for the Shortest Route Problem. *Mathematical Programming,* 7(1), 212-222.

PEREIRA, H. B. D. B. and VIDAL, L. P., 2001. Multimodal Networks for Urban Public Transportation Route Planning in Barcelona: A Decision-Making Problem from the Individual's Perspective. *XV ANPET - Congresso de Pesquisa e Ensino em Transportes.* Campinas, Brazil.

PIJLS, W. and POST, H. N. 2006. Bidirectional A*: Comparing Balanced and Symmetric Heuristic Methods. *Econometric Institute Report EI,* 41.

PYRGA, E., *et al.* 2008. Efficient Models for Timetable Information in Public Transportation Systems. *Journal of Experimental Algorithmics (JEA),* 12, 1-39.

REHRL, K., BRUNTSCH, S. and MENTZ, H.-J. 2007. Assisting Multimodal Travelers: Design and Prototypical Implementation of a Personal Travel Companion. *IEEE Transactions on Intelligent Transportation Systems,* 8(1), 31–42.

REHRL, K., *et al.*, 2005. Combined Indoor/Outdoor Smartphone Navigation for Public Transport Travellers. *3rd Symposium on Location Based Services & Telecartography.* Vienna, Austria.

RODRIGUE, J.-P., COMTOIS, C. and SLACK, B., 2009. *The Geography of Transport Systems.* 2nd ed. New York: Routledge.

RÜETSCHI, U.-J. and TIMPF, S., 2005. Modelling Wayfinding in Public Transport: Network Space and Scene Space. *Spatial Cognition Iv. Reasoning, Action, and Interaction.* 24-41.

SANDERS, P. and SCHULTES, D., 2005. Highway Hierarchies Hasten Exact Shortest Path Queries. *Algorithms – Esa 2005.* 568-579.

SANDERS, P. and SCHULTES, D., 2007. Engineering Fast Route Planning Algorithms. *Experimental Algorithms.* Springer, 23-36.

SANTOS, J. L., 2009. Real-World Applications of Shortest Path Algorithms. *In:* DEMETRESCU, C., GOLDBERG, A. V. and JOHNSON, D. S. eds. *Shortest Path Problem.* Providence: Amer Mathematical Soc, 1-17.

SCHULZ, F., WAGNER, D. and ZAROLIAGIS, C., 2002. Using Multi-Level Graphs for Timetable Information in Railway Systems. *Algorithm Engineering and Experiments.* 43-59.

SONG, H. and CHEN, G., Minimum Cost Delivery Problem in Intermodal Transportation Networks. ed. *2007 IEEE International Conference on Industrial Engineering and Engineering Management,* 2007, 1502-1506.

SOUTHWORTH, F. and PETERSON, B. E. 2000. Intermodal and International Freight Network Modeling. *Transportation Research Part C: Emerging Technologies*, 8(1-6), 147–166.

SU, J.-M. and CHANG, C.-H. 2010. The Multimodal Trip Planning System of Intercity Transportation in Taiwan. *Expert Systems with Applications*, In Press, Uncorrected Proof.

SUNG, K., *et al.* 2000. Shortest Paths in a Network with Time-Dependent Flow Speeds. *European Journal of Operational Research*, 121(1), 32-39.

TAN, Z., *et al.* 2007. Modeling the Effects of Multi-Modal Traveler Information Systems. *Journal of the Eastern Asia Society for Transportation Studies*, 7, 328-343.

THORUP, M. 2004. Integer Priority Queues with Decrease Key in Constant Time and the Single Source Shortest Paths Problem. *Journal of Computer and System Sciences*, 69(3), 330-353.

TRANSAFETY 1997. Study Compares Older and Younger Pedestrian Walking Speeds. *Road Management & Engineering Journal*.

TSCHELIGI, M. and SEFELIN, R., 2006. Mobile Navigation Support for Pedestrians: Can It Work and Does It Pay Off? *Interactions*. 31-33.

TURNER, D., 2009. *OpenTripPlanner* [online]. Available from: http://opentripplanner.org/.

VAN ESSEN, R. and HIESTERMANN, V., 2005. "X-GDF" - the ISO Model of Geographic Information for ITS. *ISPRS Workshop on Service and Application of Spatial Data Infrastructure*. Hangzhou, China.

VAN NES, R., 2002. *Design of Multimodal Transport Networks : A Hierarchical Approach*. Doctor thesis (Doctor). DUP Science.

VITTER, J., *et al.*, 1999. Dijkstra's Algorithm on-Line: An Empirical Case Study from Public Railroad Transport. *Algorithm Engineering*. Springer Berlin / Heidelberg, 110-123.

VOLKER, L., 2008. *Route Planning in Road Networks with Turn Costs*. Student research project. Universität Karlsruhe.

WAGNER, D. and WILLHALM, T., 2007. Speed-up Techniques for Shortest-Path Computations. *Stacs 2007*. Springer Berlin / Heidelberg, 23-36.

WANG, X.-B., *et al.*, Modeling and Implementing Research of Multimodal Transportation Network. ed. *1st International Conference on Information Science and Engineering (ICISE)*, 26-28 Dec. 2009, 2100-2103.

WINTER, S. 2002. Modeling Costs of Turns in Route Planning. *Geoinformatica*, 6(4), 345-361.

YANG, L., 2008. *Study on Multimodal-Supported Composite Transportation Network Model and Key Technologies*. Doctor thesis (Doctor). China University of Geosciences.

YU, H. and LU, F., 2010. A Multi-Modal Route Planning Approach with an Improved Genetic Algorithm. *Joint International Conference on Theory, Data Handling and Modelling in GeoSpatial Information Science*. Hong Kong.

ZENG, W., CHANG, X.-J. and LV, J.-J. 2010. Design of Data Model for Urban Transport GIS. *Journal of Geographic Information System,* 2(2), 106-112.

ZENG, W. and CHURCH, R. L. 2009. Finding Shortest Paths on Real Road Networks: The Case for A*. *International Journal of Geographical Information Science,* 23(4), 531-543.

ZHAN, F. B. 1997. Three Fastest Shortest Path Algorithms on Real Road Networks: Data Structures and Procedures. *Journal of Geographic Information and Decision Analysis,* 1(1), 69-82.

ZHAN, F. B. and NOON, C. E. 1998. Shortest Path Algorithms: An Evaluation Using Real Road Networks. *Transportation Science,* 32(1), 65–73.

ZHANG, M., 2009. *Methods and Implementations of Road-Network Matching.* (Doctor). Technische Universität München.

ZILIASKOPOULOS, A., AIFADOPOULOU, G. and CHRISOHOOU, E. 2007. Multiobjective Optimum Path Algorithm for Passenger Pretrip Planning in Multimodal Transportation Networks. *Transportation Research Record: Journal of the Transportation Research Board,* 2032, 26-34.

ZILIASKOPOULOS, A. and WARDELL, W. 2000. An Intermodal Optimum Path Algorithm for Multimodal Networks with Dynamic Arc Travel Times and Switching Delays. *European Journal of Operational Research,* 125(3), 486–502.

ZOGRAFOS, K. G. and ANDROUTSOPOULOS, K. N. 2008. Algorithms for Itinerary Planning in Multimodal Transportation Networks. *IEEE Transactions on Intelligent Transportation Systems,* 9(1), 175–184.

ZOGRAFOS, K. G., ANDROUTSOPOULOS, K. N. and SPITADAKIS, V. 2009. Design and Assessment of an Online Passenger Information System for Integrated Multimodal Trip Planning. *IEEE Transactions on Intelligent Transportation Systems,* 10(2), 311-323.

Acknowledgements

Before the summer of 2007, I had never thought of coming abroad. It is Prof. Ning Jing from National University of Defense Technology (NUDT) and Prof. Liqiu Meng from Technische Universität München (TUM) who provided me such an invaluable opportunity to study in Germany. The opportunity was a challenge to me as well. I was not sure whether I could accomplish this job because there were several barriers I had to overcome back then. It was hard for me to judge the level of difficulty of the new research topic, the required foreign language skill, and the totally new living environment before I came to Germany.

As soon as I started working in the Department of Cartography, TUM, a lot of people here offered me great help. My supervisor, Prof. Dr.-Ing. Liqiu Meng, encouraged me to try my various ideas in the research all the time, which made me become more and more self-confident. She always knew how to guide me to the correct direction in the academic world. During the past three years, we had numerous insightful discussions. She taught me how to manage a research project, how to write scientific papers, how to promote my soft skills, etc. She helped me a lot to overcome the difficulties I met. I owe my deepest gratitude to her.

I would like to show my gratitude to my co-supervisor, Prof. Dr. Alexander Zipf, for reviewing my work and giving me valuable comments.

I am indebted to the whole team in the Department of Cartography, TUM and many of my fellow colleagues. In particular, Dr.-Ing. Hongchao Fan gave me so much warm-hearted help in the past three years. I cannot imagine what my life in Germany would be like if there were no support from him. Dr.-Ing Wei Yao is so open-minded and shared a lot of research ideas, experiences and information with me, which helped me a lot broaden my academic vision. I am truly grateful to them.

It was only one month after I got married that I left for Germany. The three years' separation was really tough for my wife Dr. Kangle Li and me. I appreciate her endless support and understanding. I am lucky to grow up in a loving family that is always a source of strength and fun. This thesis is dedicated to my family which has constantly supported me.

The research work in this dissertation is partly funded by Corp. United Maps. I would like to thank Andreas Wiedmann, Carsten Recknagel, and Stefan Knecht for their valuable advice and significant support with test data.

Curriculum Vitae

Fundamental information

Name:	Lu LIU
Nationality:	P.R. China
Date of Birth:	14 March 1982
Place of Birth:	Tianjin, P.R. China

Education

1988/09-1994/07	Kunming Lu elementary school, Tianjin
1994/09-2000/07	Tianjin No.1 middle school, Tianjin
2000/09-2004/07	Communication Engineering at National University of Defense Technology in Changsha, China (Bachelor of Engineering)
2004/09-2007/12	Information and Communication Engineering at National University of Defense Technology in Changsha, China (Master of Science). Master thesis: Research on Distributed Management Technology of Global Mass Remote Sensing Image Data.
2008/01-	PhD student in the Department of Cartography, Technische Universität München

Experiences

2008/01-	Scientific assistant in the Department of Cartography, Technische Universität München