# Algorithms of Multi-Modal Route Planning Based on the Concept of Switch Point

Lu Liu & Liqiu Meng, München

**Keywords:** Multi-Modal Navigation, Shortest Path Algorithm, Network Analysis

**Summary:** The paper addresses the task of generically finding the shortest path in multi-modal networks with the multi-modal route planning problem in transportation field as a special case. The multi-modal networks can be modelled by a data structure based on the core concept of Switch Point which abstracts the places where it is allowed for changing from one mode to another. Two routing algorithms Multi-Modal Bellman-Ford (MMBF) and Multi-Modal Dijkstra (MMD) were elicited which are respectively rooted in the classical label-correcting and label-setting methods. Both MMBF and MMD are capable of finding in multi-modal networks the shortest paths in spite of different computing complexity. The feasibility of the approach was verified in our prototype system. The results of our experiments conducted on real transportation networks showed the differences between the proposed algorithms in terms of computing performance.

**Zusammenfassung:** *Algorithmen zur multi-modalen Routenplanung auf Grundlage des Schaltpunktes*. Der Beitrag befasst sich mit der Aufgabe zum Auffinden der kürzesten Route in einem multi-modalen Netzwerk, wobei die multi-modale Routenplanung aus dem Bereich der Verkehrstechnik als eine spezielle Anwendung betrachtet wird. Die Modellierung eines multi-modalen Netzwerks basiert auf dem Kernkonzept Schaltpunkt. Bei einem Schaltpunkt handelt es sich um eine Stelle, wo eine Modalität auf eine andere umgeschaltet werden darf. Zwei Routenalgorithmen Multi-Modal Bellman-Ford (MMBF) und Multi-Modal Dijkstra (MMD) wurden dargestellt. Als Grundlage dazu dienen zwei klassische Methoden – „Label-Korrektur“ und „Label-Einstellung“. Beide MMBF und MMD sind in der Lage, die kürzeste Route in einem beliebigen multi-modalen Netzwerk aufzufinden. Die Machbarkeit des Ansatzes zur multimodalen Routenplanung wurde in einem Prototypensystem bestätigt. Ergebnisse aus unseren Untersuchungen der realen Verkehrsnetzwerke zeigen jedoch, dass sich die beiden Algorithmen hinsichtlich des Rechenaufwands voneinander unterscheiden.

## 1 Introduction

The purpose of multi-modal route planning is to provide the user with an optimal route between the source and the target of a trip. The route may utilize several transportation modes including car driving, public transportation, cycling, walking, etc. (Hochmair 2008). Taking several accessible transportation modes into account for the travel plan is a very common practice in everyday life.

A typical application scenario of multi-modal route planning, for example, is to identify the best route from *the garage exit of Technische Universität München* in *Hess street* to *the Chinese Tower of English Garden* for a tour with a car. In this case, it does not make sense to give the result of directly driving the car from the origin to the entrance of Chinese Tower even if some of the streets in the garden are allowed for cars. Since the user is not allowed to park his car at the door of the tower, he has to find an appropriate place along the route where parking is possible. As a result, the route is divided by the parking place into two segments with different transportation modes – by car and by foot. This is a simple case of multi-modal route planning problem with two involved travel modes.

DOI: 10.1127/1432-8364/2009/0031 1432-8364/09/0031 $ 3.50

Another typical scenario which is a little bit more complicated would be to find an optimal route in Munich from a suburban place such as *Seefeld* to *the Chinese Tower of English Garden* for a tour with a car after work. In this case, directly driving to a parking place and then walking to the destination may not be a best choice because there would be traffic jams in the downtown area of Munich. To stop the car at a suburban train station, and then travel to the destination by inner-city public transportation system may be a better solution. The task is not trivial because only an experienced inhabitant who is familiar with the traffic context in Munich would tend to make such a decision. This is a routing problem involving multiple networks with multiple objectives, constraints and some dynamic and fuzzy information.

A number of mature mono-modal navigation services such as car navigation, pedestrian navigation, public transportation information systems which can do route planning for a specific network are already available on the market. Some public transportation information systems can be envisaged as an embryonic form of multi-modal 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. Couckuyt et al. (2006) from Microsoft have patented the basic concepts of multi-modal navigation and some basic functions of such a system. And Rehrl et al. (2007) described the requirements of a multimodal transportation routing system in more detail. Hoel et al. (2005) from ESRI proposed their approach for efficient modeling of the multi-modal network, which was implemented as a tool in ArcGIS Network Analysis toolbox. The modeling issue of a multi-modal freight transportation network was discussed in (Southworth & Peterson 2000). Their proposed data models can be regarded as multi-layer graphs connected by transfer nodes or arcs, while some other researchers (Modesti & Sciomachen 1998, Boussedjra et al. 2004) built a single graph containing all the information of different transportation modes. The problem of finding the optimal path in a multi-modal network was also being investigated in the past two decades (Boardman et al. 1997, Modesti & Sciomachen 1998, Ziliaskopoulos & Wardell 2000, Lozano & Storchi 2001, Lozano & Storchi 2002, Boussedjra et al. 2004, Bielli et al. 2006, Hochmair 2008, Zografos & Androutsopoulos 2008). They proposed effective approaches for finding the best route in static or even dynamic multi-modal transportation networks. Nevertheless, most of the proposed routing algorithms are coupled with the specific transportation mode combinations, which may limit their acceptance in a variety of applications, although it is true that the path should respect a set of constraints on the sequence of the used modes, i. e., the path must be viable (Lozano & Storchi 2001). The method proposed by Frank (2008) is a novel approach. His idea is to combine the navigation graph and business graph together, and then apply the traditional shortest path algorithms in the product graph. In addition, the researchers from Universität Karlsruhe have also made substantial contribution to this topic. They proposed a multi-modal path finding method based on (regular-) language-constrained shortest path algorithm (Barrett et al. 2008, Pajor 2009) generalized from (formal-) language-constrained algorithm applied in transportation field by Barrett et al. (1998). Unfortunately, as they remarked in (Delling et al. 2009), using a fast routing algorithm in such a label-constrained scenario is very complicated and hence, another challenging task. In our previous work (Liu & Meng 2008), we gave a preliminary investigation of the multi-modal extension of Bellman-Ford algorithm, but lacked any theoretical analysis of the proposed algorithm. At that time, we didn’t realize that our method for generalizing mono-modal Bellman-Ford into multi-modal situation is generic enough that it can also be applied in the label-correcting algorithms (e. g., Dijkstra’s algorithm).

Our concern with the multi-modal route planning is more general. The main purpose of our work is to propose a solution which can find the optimal route in a static $N$-modal network for an arbitrary mode combination, where $N$ indicates the number of modes. More precisely, we treat the mode combination as a part of the input. To approach such a routing problem, we have scrutinized the input of the multi-modal route planning and developed a corresponding mathematic model. Based on this model, two algorithms which can solve a general problem of finding the multi-modal shortest path are implemented. The main contributions of this paper consist of: 1) a data model based on the concept of Switch Point for modeling the multi-modal networks; 2) two multi-modal shortest path algorithms generalized from the classical mono-modal shortest path algorithms, and the generalizations are based on the same basic principle; 3) a prototype system in which the proposed data model and algorithms are implemented.

The paper is organized as follows. In Section 2 we present the definition of Switch Point which is the core concept of our data model and its matrix expression, propose the multi-modal graph set containing the Switch Points, describe the multi-modal shortest path problem and present its mathematical formulation. A general multi-modal shortest path problem can be solved by *Multi-Modal Bellman-Ford* (MMBF) and *Multi-Modal Dijkstra* (MMD). The two proposed algorithms are described, demonstrated and analyzed in detail in Section 3. The experiment results related to the city of Munich in our prototype system are shown in Section 4. Finally, Section 5 gives the conclusions and an outlook.

## 2 Data Model

### 2.1 *Switch Point*

In modern urban transportation systems, many inter-modal facilities such as parking places, park and ride lots, transit hubs, trailheads, etc. are provided besides the basic transportation networks. These facilities make it easy to transfer between different transportation modes. With the collection and digitalization of inter-modal facilities from the real world and the integration of their information with navigational databases, it becomes possible to conduct automatic multi-modal route planning.

In our multi-modal data model, the inter-modal facilities are abstracted as points where a travel mode can switch from one to another. Therefore, we call them Switch Points. Generally speaking, given a set $M$ containing $N$ different modes, $m_i$ and $m_j$ as two distinctive elements of $M$, only the points satisfying some special conditions are eligible to be the Switch Points from $m_i$ to $m_j$ . All the special conditions can be expressed by a $N \times N$ Switch Point Matrix SPM (see Fig. 1). The matrix elements are denoted by $\lambda_{SP}^{m_i m_j}$, $\lambda_{SP}^{m_i m_j} =$ NIL when $m_i = m_j$. That means all the values on the diagonal of SPM are NIL because it is meaningless to switch from a mode to itself. $\lambda_{SP}^{m_i m_j}$, $m_i \neq m_j$ indicates the condition that should be satisfied for the switching from $m_i$ to $m_j$. A practical example of $\lambda_{SP}^{m_i m_j}$ is that if $m_i$ and $m_j$ indicate the modes of car driving and walking respectively, then $\lambda_{SP}^{m_i m_j}$, should be "the place is allowed to park a car".

$$
\begin{array}{cc}
 & \begin{array}{cccc} m_1 & m_2 & \cdots & m_N \end{array} \\
\begin{array}{c} m_1 \\ m_2 \\ \vdots \\ m_N \end{array} &
\begin{bmatrix}
\text{NIL} & \lambda_{SP}^{m_1 m_2} & \cdots & \lambda_{SP}^{m_1 m_N} \\
\lambda_{SP}^{m_2 m_1} & \text{NIL} & \cdots & \lambda_{SP}^{m_2 m_N} \\
\vdots & \vdots & \ddots & \vdots \\
\lambda_{SP}^{m_N m_1} & \lambda_{SP}^{m_N m_2} & \cdots & \text{NIL}
\end{bmatrix}_{N \times N}
\end{array}
$$

**Fig. 1:** Switch Point Matrix (SPM).

According to the graph theory, a graph is composed of a vertex set $V$ and an edge set $E$, which is denoted by $G = \{V, E\}$. $\lambda_{SP}^{m_i m_j}$ can be expressed by some special attributes of vertices in $G$. Therefore, a vertex $\upsilon$ is the Switch Point between two different modes $m_i$ and $m_j$, if and only if

1) $\upsilon$ is accessible by both $G_{mi}$ and $G_{mj}$;
2) $\lambda(\upsilon) = \lambda_{SP}^{m_i m_j}$.

where $G_{mi}$ and $G_{mj}$ denote the graphs of mode $m_i$ and mode$m_j$ respectively, $\lambda$ is a function and can return the attribute of a vertex.

The $\upsilon_{SP}^{m_i m_j}$ between mode $m_i$ and mode $m_j$ is not unique. All the $\upsilon_{SP}^{m_i m_j}$ constitute a set of Switch Points. The fact that $| V_{SP}^{m_i m_j} | > 1$ in most cases indicates the main difficulty of multi-modal route planning. If the Switch Points between any two modes can be uniquely determined, i. e., $| V_{SP}^{m_i m_j} | \equiv 1$, $\forall m_i, m_j \in M$, $i \neq j$, the problem can be reduced to finding the shortest paths that visit specified intermediate nodes in a multi-modal network. This is a typical kind

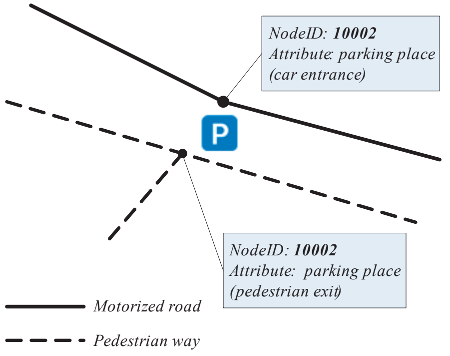



**Fig. 2:** Example of a Switch Point expressed by two geographically different nodes in two networks.

of constrained shortest path problem, which is considered by (Bajaj 1971) according to the taxonomy of shortest path algorithms (Deo & Pang 1984).

For the purpose of route planning in multimodal networks and their various combinations, we have to pay special attention to the Switch Point. Conceptually, Switch Points are similar to point features which can reside in one or more connectivity groups of the multimodal network model in ArcGIS Network Analysis toolbox (Hoel et al. 2005). However, in our Switch Point-based data model, the validity of a vertex as a Switch Point depends on the aforementioned two conditions. A $v_{SP}^{m_i,m_j}$ connecting two different modes may not be geometrically identical in the two corresponding networks. Taking urban transportation network for example, two nodes with distinct geographic coordinates are eligible to be the Switch Point connecting the two networks if they share some attribute values (e.g., node ID). Fig. 2 demonstrates a Switch Point which is a car entrance and a pedestrian exit of a parking place connecting the motorized network and the pedestrian network.

Switch Point is significant to the multi-modal route planning problem. A network dataset can not support multi-modal routing application without the information of Switch Point. The definition of a multi-modal shortest path problem and its solutions are all based on the concept of Switch Point.

From the transportation point of view, Switch Point is an abstraction of the place in the real world where people can change from one transportation mode to another. We can enumerate different Switch Points of different traffic mode-pairs in a table such as Tab. 1 which we call SPM-T (SPM in Transportation). Given an ordered transportation mode tuple $(m_i, m_j)$, we can get the values of SPM-T $(m_i, m_j)$ by looking up the matrix.

With Tab. 1 we do not attempt to enumerate all possible traffic modes which may for example include airplane, ship, inter-city train, bicycle, roller skating, etc. Instead, we focus on the most usual ones in our everyday life. The SPM-T has two properties:

***SPM-T is asymmetrical.***

Taking the car driving and walking modes for example, SPM-T $(D, W)$ is not equal to SPM-T $(W, D)$. The reason is obvious: a driver can park his car at any available parking place and transfer to pedestrian mode, however in the reverse situation, the driver must walk to a place (e.g., the parking place where his/her car is parked, or a car rental company) where there is a car he/she can use.

***SPM-T is scale-dependent.***

Tab. 1 just shows the case of inner-city map scale. At a more detailed level, we may find that there are no switch points between car

**Tab. 1:** An example of SPM-T. D, P and W denote car driving, public transportation and walking respectively.

| | *D* | *P* | *W* |
|---|---|---|---|
| *D* | NIL | P+R lots | parking lots for cars |
| *P* | P+R lots where a car is available | NIL | public transportation stations |
| *W* | some place where a car is available or can be rented | public transportation stations | NIL |

driving and public transportation because it is impossible to drive a car directly into a bus just like taking a bicycle into a suburban train. In other words, no Switch Point can exist that connects car driving mode and public transportation mode. There must be a pedestrian mode in between. On the other hand, when zooming out to a much smaller map scale, some traffic modes such as pedestrian, bicycle, inner-city public transportation, etc. may become meaningless and therefore should be ignored.

In our real life, a route planning problem with multiple travel modes can be very complicated. Still, it can be reduced to a shortest path problem or some of its variations. In our approach, we describe the original routing problem as a multi-modal shortest path problem.

## 2.2 *Formalized Description*

The input of a multi-modal shortest path problem contains five parts: 1) *Multi-Modal Graph Set* (MMGS); 2) attributes of vertices in the MMGS; 3) *Switch Point Matrix* (SPM); 4) a sequential list of modes which will be contained in the final route; 5) a source node and a target node.

The notational conventions used in this paper are identical with the second edition of the textbook *Introduction to Algorithms* (Cormen et al. 2001). Consequently, we are given a sequential list $M = \langle m_1, m_2, \ldots, m_N \rangle$, $N \geq 2$, $m_i \neq m_{i+1}$, $i \in [1, N-1]$ composed by $N$ modes, a set of vertices attributes $\Lambda$ and a vertex-labeled, non-negative weighted, acyclic, directed multi-graph $G_M = \{G_k = \{V_k, E_k\} \mid k \in M\}$ denoting MMGS. $\lambda\colon V \to \Lambda$, $V = \bigcup_{k \in M} V_k$ is the vertex label function, and $C_M = \{c_k : E_k \to R^+, k \in M\}$ is the set of cost functions that map the edges of different modes to positive real-valued costs. The source node and target node are denoted by $S$, $S \in V_{m1}$ and $T$, $T \in V_{mN}$ respectively. With the list $M$ as input, we can get the switch point value $\lambda_{SP}^{m_i m_{i+1}} = \mathrm{SPM}(m_i, m_{i+1})$, $i \in [1, N-1]$ by looking up in the SPM.

At first, we give the formalized description of double-modal shortest path problem as a simplified multi-modal shortest path problem. With the inputs listed above, in a double-modal shortest path problem, we are given a sequential modes list $M = \langle m_1, m_2 \rangle$, $m_1 \neq m_2$ composed by two different modes, a set of vertices attributes $\Lambda$ and a double-graph $G_M = \{G_{m1} = \{V_{m1}, E_{m1}\}, G_{m2} = \{V_{m1}, E_{m2}\}$, with vertex label function $\lambda\colon V \to \Lambda$, $V = V_{m1} \cup V_{m2}$, and cost functions $c_{m1} : E_{m1} \to R^+$ and $c_{m2} : E_{m2} \to R^+$. $C_M = \{c_{m1}, c_{m2}\}$ is the cost function set composed of the two functions. With the $M = \langle m_1, m_2 \rangle$ as a part of the input, we can get the $\lambda_{SP}^{m_1 m_2}$. The path cost of mode $k$ $p_k = \langle v_0, v_1, \ldots, v_t \rangle$ is the sum of the costs of its constituent edges (cf. (1)).
We define the double-modal shortest path cost from $S$ to $T$ with tow modes $\langle m_1, m_2 \rangle$ (2).

A double-modal shortest path from vertex $S$ to vertex $T$ with two modes $\langle m_1, m_2 \rangle$ involved in is then defined as any path $p_{\langle m_1, m_2 \rangle}$ with cost $c(p_{\langle m_1, m_2 \rangle}) = \delta(S, T, \langle m_1, m_2 \rangle)$.

Based on the formalized description of double-modal shortest path problem, we can generalize it to the multi-modal situation. The path cost of mode $m$ $p_k = \langle v_0, v_1, \ldots, v_t \rangle$ is the same as defined in double-modal shortest path problem by Eq. (1).

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

$$\delta(S, T, \langle m_1, m_2 \rangle) = \begin{cases} min\{c_{m1}(p_{m1}) + c_{m2}(p_{m2}) : S \overset{p_{m1}}{\rightsquigarrow} v_{SP}^{m_1 m_2} \overset{p_{m2}}{\rightsquigarrow} T \\ \qquad \mid v_{SP}^{m_1 m_2} \in V_{SP}^{m_1 m_2}\} & \text{if there is a path with the two modes } m_1, m_2 \text{ from } S \text{ to } T \\ \infty & \text{otherwise} \end{cases} \tag{2}$$

$$\delta(S, T, M) = \begin{cases} min\{\sum_{k=m_1}^{m_N} c_k(p_k) : \\ \qquad S \overset{p_{m1}}{\rightsquigarrow} v_{SP}^{m_1 m_2} \overset{p_{m2}}{\rightsquigarrow} v_{SP}^{m_2 m_3} \cdots v_{SP}^{m_{N-2} m_{N-1}} \overset{p_{mN-1}}{\rightsquigarrow} v_{SP}^{m_{N-1} m_N} \overset{p_{mN}}{\rightsquigarrow} T \\ \qquad \mid v_{SP}^{m_i m_{i+1}} \in V_{SP}^{m_i m_{i+1}}\} \\ \infty \end{cases} \tag{3}$$

We define the multi-modal shortest path cost from $S$ to $T$ with the mode list $M$ by (3).

An $N$-modal shortest path from vertex $S$ to vertex $T$ is then defined as any path $p_M$ with cost $c(p_M) = \delta(S, T, M)$.

If we restrict the discussion in urban transportation field, the shortest path with $N$ travel modes reveals some other interesting properties besides the lowest cost.

***The number of switch points in the multi-modal shortest path 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 modal switches. This property can be seen as a constraint on the number of modal transfers on a multi-modal shortest path.

***The sequence of switch points in the multi-modal shortest path has some regularity.***

The regularity of the switch points sequence can be set equal to the transportation mode sequence. A multi-modal 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 & Storchi 2001). However, finding the multi-modal shortest path and determining the relative logic of mode combination should be treated as two separate problems. The attempt to couple the algorithms with the concrete mode constraints may limit the acceptance level of routing algorithms themselves in applications other than transportation route planning, e. g., computer network routing, circuit designing, etc.

Our concern with the multi-modal route planning is more general in the sense that there are $N$ modes involved and the mode combination can be arbitrarily set. The solution is provided by two multi-modal shortest path algorithms described in the next section.

## 3 Multi-modal Shortest Path Algorithms

Our investigation starts with the double-modal shortest path problem which can be then generalized to cope with the multi-modal situation. We implemented two double-modal shortest path algorithms based on the classical label-correcting and label-setting algorithms, i. e., Bellman-Ford algorithm (Bellman 1958, Cormen et al. 2001) and Dijkstra's algorithm (Dijkstra 1959, Cormen et al. 2001) respectively, and extended them to *Multi-Modal Bellman-Ford* (MMBF) and *Multi-Modal Dijkstra* (MMD). A key step was to make a few modifications with respect to Switch Point during the initialization process.

### 3.1 *Multi-modal Bellman-Ford*

Bellman-Ford search is a well-tested label-correcting algorithm used to find the shortest path from a single source in a graph without negative-weighted cycles. Generally speaking, Bellman-Ford search procedure consists of two main steps: the initialization and the traversal on the whole graph (i. e., iteratively relaxation of all the edges). After the traversal step, the distance value on each vertex indicates the shortest distance value from the source to this vertex.

To solve the double-modal shortest path problem, we execute Bellman-Ford search twice, the first time in the $G_{m1}$ with the source $S$ and the second time in the $G_{m2}$. The initialization step in the second time is different from the first time. Instead of setting all the distance values on the vertices to infinity, we kept the values calculated on the Switch Points in $G_{m1}$. After this modified initialization, we did Bellman-Ford traversal in $G_{m2}$ and got the final shortest paths from $S \in V_{m1}$ to any vertex $\upsilon \in V_{m1}$ with mode $m_1$ and any vertex $\upsilon \in V_{m2}$ with modes $m_1$ and $m_2$ via Switch Points.

Fig. 3 demonstrates the work flow of the double-modal Bellman-Ford algorithm on two graphs with a single source-target pair. The first graph is $G_{m1} = \{V_{m1}, E_{m1}\}$ with 6 vertices and 9 edges of mode $m_1$, and the second one is $G_{m2} = \{V_{m2}, E_{m2}\}$ with 7 vertices and 18 edges of mode $m_2$. The source is vertex $S$ in $G_{m1}$. The

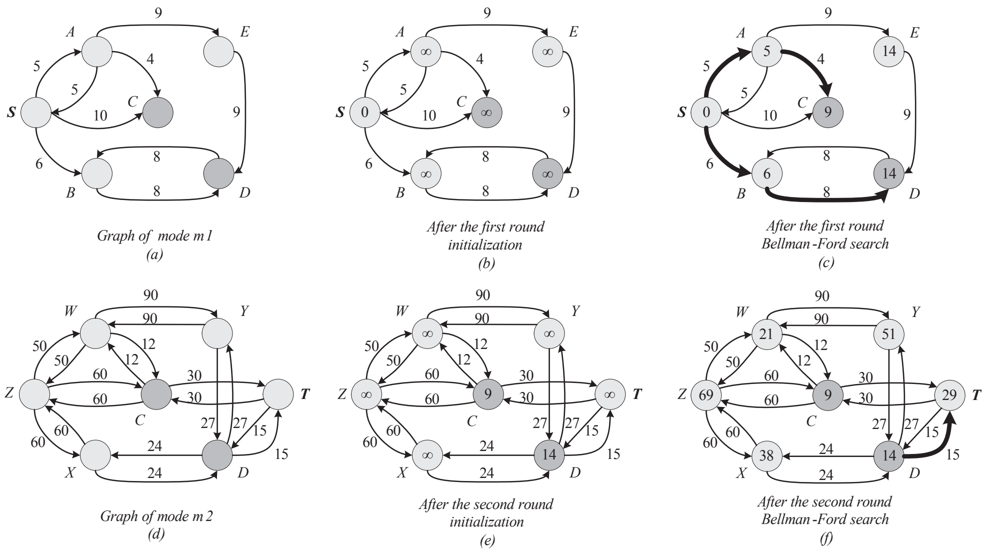



**Fig. 3:** Work flow of the double-modal Bellman-Ford algorithm.

target is vertex $T$ in $G_{m2}$. Fig. 3 (*a*) and (*d*) are the original graphs. The vertices in dark grey $C$ and $D$ indicate Switch Points from mode $m_1$ to $m_2$. The distance values after initialization and searching processes are shown within the vertices. The thick edges in Fig. 3 (*c*) indicate the shortest paths from $S$ to the Switch Points. The second initialization is illustrated in Fig. 3 (*e*). The thick edge in Fig. 3 (*f*) indicates the shortest path from Switch Point to the target. In this example, the double-modal shortest path from $S$ to $T$ is $S \rightarrow B \rightarrow D$ (Switch Point) $\rightarrow T$ with the final distance value of 29.

In a multi-modal situation with an ordered mode list $M$, we can also get the shortest paths by applying the modified initialization in the graphs from $G_{m2}$ and traverse graphs of further modes. Thus, the double-modal Bellman-Ford algorithm is generalized into MultiModalBellmanFord (MMBF). The whole process can be described as follows:

```
MultiModalBellmanFord(M, G_M, C_M, S)
for i ← 1 to N
    do if i = 1
        then MultiModalInitialize(G_M, m_i, NIL, S)
        else MultiModalInitialize(G_M, m_i, λ_SP^{m_{i-1}, m_i}, S)
    BellmanFordSearch(G_M, m_i, C_M)
```

where the routines MultiModalInitialize and BellmanFordSearch work as follows:

```
MultiModalInitialize(G_M, m, λ_SP, S)
if λ_SP = NIL
    then for each vertex v ∈ V[G_m]
            do distance[m][v] ← ∞
               predecessor[m][v] ← NIL
         distance[m][S] ← 0
    else for each vertex v ∈ V[G_m]
            do if λ(v) ≠ λ_SP
                  then distance[m][v] ← ∞
               predecessor[m][v] ← NIL
```

The computing complexity is $O(\sum_{k \in M} |V_k||E_k|)$ for the MMGS $G_m$ and the mode list $M$, since the MultiModalInitialize takes $\Theta(V_k)$ time and the BellmanFordSearch takes $\Theta((|V_k|-1)|E_k|)$ time in each of the $N$ passes.

***Theorem 3-1 (Effectiveness of the MMBF)***

Let MMBF be run on a vertex-labeled, non-negative weighted, acyclic, directed MMGS $G_M = \{G_k = \{V_k, E_k\} \mid k \in M\}$ where $M = \langle m_1, m_2, \ldots, m_N \rangle$, $N \geq 2$, $m_i \neq m_{i+1}$, $i \in [1, N-1]$ with source $S$ and cost function set $C_M = \{c_k: E_k \rightarrow R^+, k \in M\}$. We can have $distance[m_i][v] = \delta(S, v, \langle m_1, \ldots, m_i \rangle)$ for all vertices $v \in V_{mi}$.

***Proof***

After the first round BellmanFordSearch on $G_{m1}$, $distance\,[m_1]\,[V_{SP}^{m_1 m_2}] = \delta\,(S, V_{SP}^{m_1 m_2}, m_1)$. After the second round MultiModalInitialize on $G_{m2}$, all the $\delta\,(S, V_{SP}^{m_1 m_2}, m_1)$ are kept as the initial values of the vertices $v_{SP}^{m_1 m_2} \in V_{SP}^{m_1 m_2}, m_1$, i. e., $distance\,[m_2]\,[V_{SP}^{m_1 m_2}] = \delta\,(S, V_{SP}^{m_1 m_2}, m_1)$. To prove $distance\,[m_2]\,[v] = \delta\,(S, v, \langle m_1, m_2\rangle)$ after the second round BellmanFordSearch, we add a virtual source vertex $S_{m2}$ directly connecting to the vertices $v_{SP}^{m_1 m_2} \in V_{SP}^{m_1 m_2}$ in $G_{m2}$ which becomes $G'_{m2}$ (Fig. 4(*a*)). The costs of the edges from $S_{m2}$ to $V_{SP}^{m_1 m_2}$ are exactly $\delta\,(S, V_{SP}^{m_1 m_2}, m_1)$. As a result, there are $|\,V_{m2}\,| + 1$ vertices in $G'_{m2}$, and a normal Bellman-Ford's algorithm should take $|\,V_{m2}\,|$ iterations of the relaxes of all $|\,E_{m2}\,| + |\,V_{SP}^{m_1 m_2}\,|$ edges. In fact, the status of $G_{m2}$ after MultiModalInitialize is exactly the same as that after the initialization and the first iteration of relaxes all the edges in $G'_{m2}$. That means the algorithm just needs to do the remaining $|\,V_{m2}\,| - 1$ iterations in $G'_{m2}$, which is exactly the work of BellmanFordSearch in $G_{m2}$. After that, we can get the shortest distance values on all the vertices in $G_{m2}$, i. e., $distance\,[m_2]\,[v] = \delta\,(S, v, \langle m_1, m_2\rangle)$. In general, we can add virtual source vertex $S_{mi}$ in the graph of the $i$th mode $i \in [2, N]$, and the edges whose costs are $\delta\,(S, V_{SP}^{m_{i-1} m_i}, \langle m_1, \ldots, m_i\rangle)$ from $S_{mi}$ to the Switch Points from the last graph to $G_{mi}$. After the MultiModalInitialize and BellmanFordSearch in $G_{mi}$, which is equivalent to the normal Bellman-Ford search in $G'_{mi} = \{V_{mi} \cup \{S_{mi}\}, E_{mi} \cup \{S_{mi} \rightarrow V_{SP}^{m_{i-1} m_i}\}\}$, we can have $distance\,[m_i]\,[v] = \delta\,(S, v, \langle m_1, \ldots, m_i\rangle)$. ■

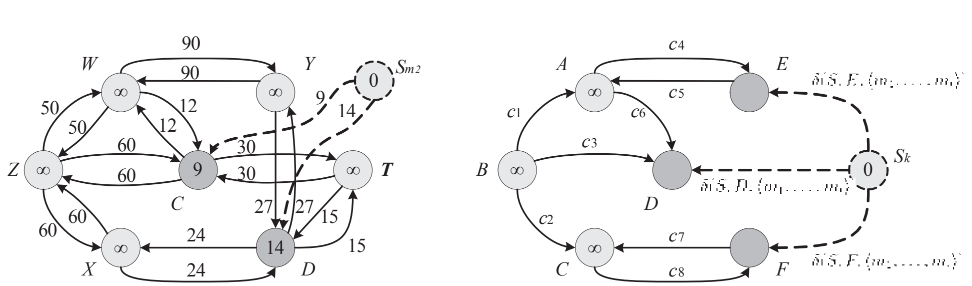
## 3.2 *Multi-modal Dijkstra*

In MMBF algorithm, the key action is the reservation of the distance values on Switch Points during the initialization step in each round of Bellman-Ford searching from the mode $m_2$ to $m_n$ through which the multi-modal shortest path can be finally found. This idea can also be applied in Dijkstra search which is a well-tested label-setting algorithm. Against the assumption made in our early study (Liu & Meng 2008) that only double-modal Bellman-Ford has the potential to be generalized to a multi-modal situation while the double-modal Dijkstra search can only find a good but not necessarily the optimal path. With our most recent experiments we have proved the possibility of finding the shortest path using double-modal Dijkstra or its further extension MultiModalDijkstra (MMD). The MMD algorithm can be described as follows.

```
MultiModalDijkstra(M, G_M, C_M, S)
for i ← 1 to N
    do if i = 1
        then MultiModalInitialize(G_mi, m_i, NIL, S)
        else MultiModalInitialize(G_mi, m_i, V_SP^(m_i-1 m_i), S)
    DijkstraSearch(G_mi, m_i, C_mi)
```

where the MultiModalInitialize is the same as in MMBF and DijkstraSearch works in the following way:

```
DijkstraSearch(G_mi, m_i, C_mi)
Q ← V[G_mi]
while Q ≠ ∅
    do u ← vertex in Q with the minimal distance[m_i][u]
        for each vertex v ∈ Adj[u]
            do if distance[m_i][v] > distance[m_i][u] + c_mi(u, v)
                then distance[m_i][v] ← distance[m_i][u] + c_mi(u, v)
                    predecessor[m_i][v] ← u
```




*Graph of mode $m_2$ (in Fig. 3) added by the virtual source and edges*
*(a)*

*Graph of mode $m_i$ added by the virtual source and edges*
*(b)*

**Fig. 4:** The effectiveness proof of MMBF by adding virtual sources and the corresponding edges.

When the min-priority queue $Q$ is implemented as an ordinary array, its computing time complexity is $O(\sum_{k \in M} |V_k|^2)$ for the MMGS $G_M$ and the mode list $M$, since the MultiModalInitialize takes $\Theta(V_k)$ time and the DijkstraSearch with an ordinary array as its min-priority queue takes $O(|V_k|^2)$ time in each of the $N$ passes.

The demonstration of MMD in a double-modal situation with a single source-target pair can also be expressed by Fig. 3. The statuses of the graphs after the two rounds MultiModalInitialize are exactly the same as that in MMBF. This means, the statuses of the graphs after DijkstraSearch are the same as that shown in Fig. 3 (*c*) and (*f*).

***Theorem 3-2 (Effectiveness of the MMD)***

MMD, run on a vertex-labeled, non-negative weighted, acyclic, directed MMGS $G_M = \{G_k = \{V_k, E_k\} \mid k \in M\}$ where $M = \langle m_1, m_2, \ldots, m_N \rangle$, $N \geq 2$, $m_i \neq m_{i+1}$, $i \in [1, N-1]$ with source $S$ and cost function set $C_M = \{c_k : E_k \rightarrow R^+, k \in M\}$, terminates with $distance[m_i][v] = \delta(S, v, \langle m_1, \ldots, m_i \rangle)$ for all vertices $v \in V_{mi}$.

***Proof***

The basic method used in the effectiveness proof of MMBF can also be used here. In general, we can add a virtual source vertex $S_{mi}$ in the graph of the $i$th mode $i \in [2, N]$ and the edges whose costs are $\delta(S, V_{SP}^{m_{i-1}m_i}, \langle m_1, \ldots, m_i \rangle)$ from $S_{mi}$ to the switch points from $G_{mi-1}$ to $G_{mi}$. $G_{mi}$ becomes $G'_{mi} = \{V_{mi} \cup \{S_{mi}\}, E_{mi} \cup \{S_{mi} \rightarrow V_{SP}^{m_{i-1}m_i}\}\}$. The MultiModalInitialize in $G_{mi}$ is equivalent to the normal Dijkstra's initialization together with the first round relaxes on the virtual edges emitted from $S_{mi}$ in $G'_{mi}$. As a result, each time the execution of MultiModalInitialize and DijkstraSearch in $G_{mi}$ is equivalent to the normal Dijkstra's algorithm in $G'_{mi}$. Therefore, when the MMD terminates, we have *distance* $[m_i][v] = \delta(S, v, \langle m_1, \ldots, m_i \rangle)$. ■

For any vertex in any $G_{mi}$, the distance value on it indicates the shortest distance from the source in$G_{m1}$ to this vertex via the mode sequence $(m_1, \ldots, m_i)$ after MMD. For the classical Dijkstra's algorithm with a single source-target pair input, the performance can be considerably improved by terminating the search when the target is reached. This improvement can also be used in MMD with a single source-target pair. If we are only interested in the shortest path between source and target via mode sequence $M$, the search process can be terminated and continued into the next graph when it has reached all the $v_{SP}^{m_i m_{i+1}} \in V_{SP}^{m_i m_{i+1}}$ from $G_{m1}$ to $G_{m_{N-1}}$, and be finally terminated when it has reached the target in $G_{m_N}$. In this case, the original MultiModalDijkstra becomes MultiModalDijkstra-Target depicted as follows.

```
MultiModalDijkstra-Target(M, G_M, C_M, S, T)
for i ← 1 to N
    do if i = 1
          then MultiModalInitialize(G_mi, m_i, {S}, S)
          else MultiModalInitialize(G_mi, m_i, V_SP^(m_i-1 m_i), S)
       if i = N
          then DijkstraSearch-Target(G_mi, m_N, ∅, C_mi, T)
          else DijkstraSearch-Target(G_mi, m_i, V_SP^(m_i m_i+1), C_mi, T)
```

where the DijkstraSearch should be changed into DijkstraSearch-Target:

```
DijkstraSearch-Target(G_mi, m_i, V_SP, C_mi, T)
Q ← V[G_mi]
while Q ≠ ∅
    do u ← vertex in Q with the minimal distance[m_i][u]
       remove u from Q
       if m_i = m_N
          then if u = T
                  then return
          else if u ∈ V_SP
                  then remove u from V_SP
               if V_SP = ∅
                  then return
       for each vertex v ∈ Adj[u]
           do if distance[m_i][v] > distance[m_i][u] + c_mi(u, v)
                 then distance[m_i][v] ← distance[m_i][u] + c_mi(u, v)
                      predecessor[m_i][v] ← u
```

This improvement by terminating the searching process ahead of time can not be applied in MMBF because it is based on label-correcting method which can get the shortest distance value on all the vertices only after the very last iteration.

According to our complexity analysis, MMD with an ordinary array as its min-priority queue is faster than MMBF. MMD-T should be faster than MMD practically although in the worst case they have the same running time. The performance differences between MMBF and MMD were verified in our experiments described in section 4. The average improvement of MMD-T can also be seen from the results.

## 4 Experiments

We implemented the two proposed algorithms in C# and developed a prototype system by

Visual Studio 2005 Professional. The programs run on a PC with a 2.2 GHz Intel Core2 Duo CPU and 2 GB physical memory under Windows XP Professional SP3. Our tests were conducted on the spatial datasets of a portion of Munich from two different sources. One is OpenStreetMap (OSM) which has been enriched with Switch Point information, the other is a navigation dataset provided by United Maps Co., Ltd. (UM) which integrates the

**Tab. 2:** The size of test networks.

| Data source | Area of coverage (width, length)* | Mode | \|V\| | \|E\| | \|E\|/\|V\| |
|---|---|---|---|---|---|
| UM | (20.34, 26.18) | $D$ | 19471 | 44979 | 2.31 |
| | | $W$ | 20516 | 57694 | 2.81 |
| | (9.829, 11.00) | $U$ | 64 | 132 | 2.06 |
| OSM | (4.970, 4.663) | $D$ | 4807 | 9125 | 1.90 |
| | | $W$ | 9077 | 22482 | 2.48 |

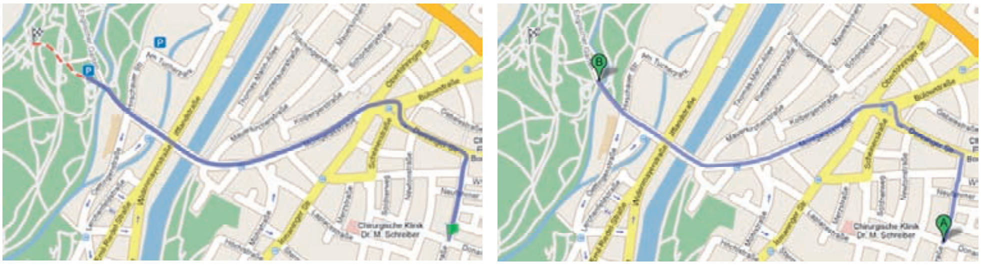
* The unit of width and length is km.

**Tab. 3:** Computing speed of MMBF and MMD.

| Data source | Mode list | | MMBF(s) | MMD(s) |
|---|---|---|---|---|
| UM | 1-modal | $D$ | 354.9 | 10.28 |
| | | $W$ | 536.7 | 12.27 |
| | | $U$ | 0.003125 | 0.001094 |
| | 2-modal | $\langle D, W\rangle$ | 862.7 | 24.81 |
| | | $\langle W, U\rangle$ | 467.3 | 12.21 |
| | | $\langle U, W\rangle$ | 467.7 | 13.13 |
| | 3-modal | $\langle D, W, U\rangle$ | 817.1 | 25.11 |
| | | $\langle W, U, W\rangle$ | 936.5 | 25.57 |
| | 4-modal | $\langle D, W, U, W\rangle$ | 1287 | 37.98 |
| OSM | 1-modal | $D$ | 15.99 | 0.4941 |
| | | $W$ | 76.42 | 2.239 |
| | 2-modal | $\langle D, W\rangle$ | 90.78 | 3.013 |

**Tab. 4:** Computing speed of MMD and MMD-T on UM dataset.

| Mode list | | MMD (s) | MMD-T (s) | Improvement. ratio (%) |
|---|---|---|---|---|
| 2-modal | $\langle D, W\rangle$ | 23.70 | 16.93 | 40.0 |
| | $\langle W, U\rangle$ | 14.96 | 12.02 | 24.5 |
| | $\langle U, W\rangle$ | 12.70 | 9.066 | 40.1 |
| 3-modal | $\langle D, W, U\rangle$ | 23.75 | 19.27 | 23.2 |
| | $\langle W, U, W\rangle$ | 23.83 | 19.91 | 19.7 |
| 4-modal | $\langle D, W, U, W\rangle$ | 35.99 | 27.65 | 30.2 |
| | | | | Avg: 29.6 |



Our multi-modal routing result drawn on Google's map
(*a*)

Google Maps' routing result
(*b*)

**Fig. 5:** Comparison between the routing results of our prototype system and Google Maps.

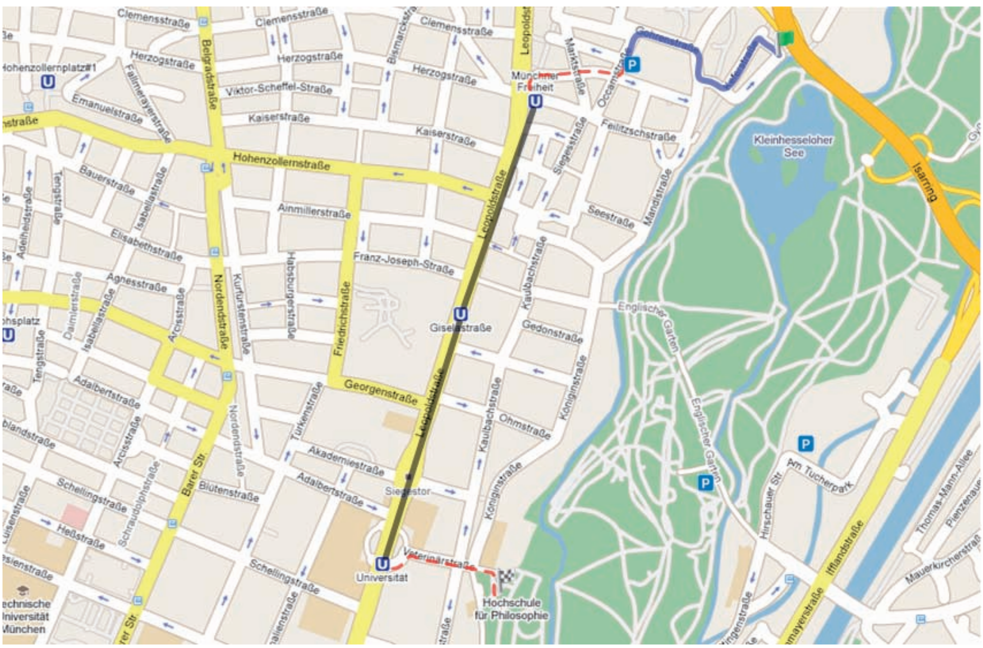



**Fig. 6:** A multi-modal routing result with three modes.

information from Navteq and ATKIS (the official topographic cartographic information system in Germany). The two datasets differ in the areas they cover as well as the size of their corresponding networks.

The basic information of the two datasets is listed in Tab. 2. There are three transportation modes of network in UM dataset: motorized ways denoted by *D*, pedestrian ways denoted by *W* and underground lines denoted by *U*. In OSM dataset, there are *D* and *W*.

The experiments consist of two groups: one is the test of single-source multi-modal shortest path algorithms for the purpose of comparing the performances between MMBF and MMD on the two datasets; the other is the test of single source-target-pair MMD to investigate how much performance improvement can be made by MMD-T. For a given combination of dataset, mode list and multi-modal shortest path algorithm, an average running time of 100 s execution of MMD with randomly selected source (and target for MMD-T) and 10 s for MMBF was recorded. The reported running times do not include data input or log output. The SPM (or, more precisely, SPM-T) used for the experiment is shown in Tab. 1 in Section 2. It should be noticed that the Dijkstra search can reveal considerably different computing performance if it is applied in a different data structure. For this reason, we used an ordinary array as the min-priority queue without using any built-in data structures or methods provided by .net framework class library. In this way, our test results are independent of the internal optimization made by Microsoft.

The experimental results of the first group are listed in Tab. 3 which shows that the MMBF is much slower than MMD, which has confirmed our analysis of theoretical complexities.

The experimental results of the second group are listed in Tab. 4 which shows that the average improvement made by MMD-T over MMD amounts to 29.6%.

Fig. 5 shows the routing result of our algorithm in comparison with Google's route from a crossing of motorized roads to a pedestrian junction in the *English Garden*. The destination is shown by a checkered flag. We can see that the car segments (blue solid line) of the two routes are exactly the same. However, the pedestrian segment (red dashed line) in our route is missing in Google's route. The reason for that is in Google Maps (and almost all the currently available routing systems) only one travel mode can be selected, e. g., either by car or by foot, but not both. Our routing algorithm allows the car-pedestrian combination with an appropriate parking place as the Switch Point.

Fig. 6 shows another multi-modal routing result calculated by our algorithms. Three modes are involved in the route: car driving (blue solid line), walking (red dashed line), underground train (black solid line) and walking again.

## 5 Conclusions and Outlook

The multi-modal route planning problem addressed in this paper originates from the transportation field. However, we developed a multi-modal routing strategy in a general sense, thus opened up its applicability for the fields beyond transportation. We proposed a data model with Switch Point as the core concept and gave the formal description of the multi-modal shortest path based on Switch Point. Two algorithms MMBF and MMD were developed on the basis of label-correcting and label-setting algorithms respectively. They reveal the complexities of $O(\sum_{k \in M} |V_k||E_k|)$ and $O(\sum_{k \in M} |V_k|^2)$ respectively. The effectiveness of the algorithms has been proved, which can ensure that the found paths are the shortest in terms of distances or weights. Our experiments were conducted on real transportation networks from two different data sources. MMBF is much slower than MMD, and MMD-T runs 29.6% faster than MMD on average in case of the single source-target pair. From the practical point of view, the MMBF is inefficient for real-time applications although it is able to find the multi-modal shortest path.

The conducted experiments have shown the convincing process of solving a multi-modal route planning problem. However, multi-modal routing in our real life can take various far more complicated forms which will challenge the computing power of routing algorithms. Therefore, our work for the next steps will focus on two points:

*1) To explore the further potential of computing performance of the routing algorithms*

The multi-modal routing planning algorithms introduced in this paper are based on the two earliest and well-tested shortest path algorithms. In fact, some researchers have proposed improved versions of mono-modal shortest path algorithms (Gallo & Pallottino 1988, Cherkassky et al. 1996). According to the evaluation done by (Zhan & Noon 1998) using real road networks and the comparison work between label-setting and label-correcting algorithms for computing single source-target pair shortest paths by (Zhan & Noon 2000), TWO-Q is the most efficient routing algorithm for shortest path finding in road networks (Pallottino 1984). We attempt to implement a new multi-modal shortest path algorithm based on TWO-Q and test its performance in the near future.

*2) To take the knowledge from specific field into consideration.*

There is a lot of context knowledge from the transportation field which should be embedded in the route planning. For example, the path should be "viable" which we mentioned in Section 2, the traffic rules in the transportation networks, the cost when changing the mode, the dynamic traffic information, etc. We will verify and specify our approach on multi-modal transportation networks enhanced with knowledge.

## Acknowledgements

The research work is sponsored by United Maps Co., Ltd. The high-quality and multimodal navigation dataset is jointly created by United Maps Co., Ltd. and Department of Cartography at Technische Universität München in the frame of a project on data integration. The authors would like to thank Mr. Hongchao Fan who gave useful suggestions about improving this paper.

## References

Bajaj, C.P., 1971: Some Constrained Shortest-Route Problems. – Mathematical Methods of Operations Research **15** (1): 287–301.

Barrett, C., Bisset, K., Holzer, M., Konjevod, G., Marathe, M. & Wagner, D., 2008: Engineering Label-Constrained Shortest-Path Algorithms. – Algorithmic Aspects in Information and Management: 27–37.

Barrett, C., Jacob, R. & Marathe, M., 1998: Formal Language Constrained Path Problems. – Algorithm Theory — Swat'98: 234–245.

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

Bielli, M., Boulmakoul, A. & Mouncif, H., 2006: Object Modeling and Path Computation for Multimodal Travel Systems. – European Journal of Operational Research **175** (3): 1705–1730.

Boardman, B.S., Malstrom, E.M., Butler, D.P. & Cole, M.H., 1997: Computer Assisted Routing of Intermodal Shipments. – 21st International Conference on Computers and Industrial Engineering: 311–314.

Boussedjra, M., Bloch, C. & El Moudni, A., 2004: An Exact Method to Find the Intermodal Shortest Path (Isp). – IEEE International Conference on Networking, Sensing and Control: 1075–1080.

Cherkassky, B., Goldberg, A. & Radzik, T., 1996: Shortest Paths Algorithms: Theory and Experimental Evaluation. – Mathematical Programming **73** (2): 129–174.

Coast, S., 2004: Openstreetmap Project. – www.openstreetmap.org

Cormen, T.H., Leiserson, C.E., Rivest, R.L. & Stein, C., 2001: Introduction to Algorithms. – The MIT Press and McGraw-Hill.

Couckuyt, J.D., McGrath, T.S. & Sutanto, H., 2006: Multi-Modal Navigation System and Method. – United States Patent, Application 11/058,057.

Delling, D., Sanders, P., Schultes, D. & Wagner, D., 2009: Engineering Route Planning Algorithms. – Algorithmics of Large and Complex Networks. Springer.

Deo, N. & Pang, C.-Y., 1984: Shortest-Path Algorithms: Taxonomy and Annotation. – Networks **14** (2): 275–323.

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

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

Gallo, G. & Pallottino, S., 1988: Shortest Path Algorithms. – Annals of Operations Research **13** (1): 1–79.

Hochmair, H.H., 2008: Grouping of Optimized Pedestrian Routes for Multi-Modal Route Planning: A Comparison of Two Cities. – The European Information Society, Springer, Berlin: 339–358.

Hoel, e. g., Heng, W.-L. & Honeycutt, D., 2005: High Performance Multimodal Networks. – Advances in Spatial and Temporal Databases, Springer, Berlin: 308–327.

Liu, L. & Meng, L., 2008: Algorithmic Concerns of Multi-Modal Route Planning. – 5th International Symposium on LBS & TeleCartography, 26–28 November, Salzburg, Austria.

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

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

Modesti, P. & 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.

Pajor, T., 2009: Multi-Modal Route Planning. Diplomarbeit, Karlsruhe, Universität Karlsruhe.

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

Rehrl, K., Bruntsch, S. & 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.

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

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

Zhan, F.B. & Noon, C.E., 2000: A Comparison between Label-Setting and Label-Correcting Algorithms for Computing One-to-One Shortest Paths. – Journal of Geographic Information and Decision Analysis **4** (2): 1–11.

Ziliaskopoulos, A. & 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. & Androutsopoulos, K.N., 2008: Algorithms for Itinerary Planning in Multimodal Transportation Networks. – IEEE Transactions on Intelligent Transportation Systems **9** (1): 175–184.

Address of the Authors:

M.Eng. Lu Liu, Prof. Dr.-Ing. Liqiu Meng, Technische Universität München, Department of Cartography, D-80333 Munich, Germany, Tel.: +49-89-2892-2586, -2825, Fax: +49-89-2809573, e-mail: liu.lu@bv.tum.de, meng@bv.tum.de

Manuskript eingereicht: Mai 2009
Angenommen: Juli 2009