Proceedings of the 9th International Conference "Reliability and Statistics in Transportation and Communication" (RelStat'09), 21–24 October 2009, Riga, Latvia, p. 193-199. ISBN 978-9984-818-21-4
Transport and Telecommunication Institute, Lomonosova 1, LV-1019, Riga, Latvia

TIMETABLE INFORMATION SYSTEMS TO FIND OPTIMAL TRAVEL ON PUBLIC TRANSPORT

Vladimir Vishnevsky, Roman Zhelezov

ZAO Research & Development Company "Information and Networking Technologies"
Staroalekseevskaja Str. 5-215, Moscow, 129626 Russia
E-mail: vishn@inbox.ru, ironromeo@mail.ru

We present architecture of a developed timetable information system and describe an original algorithm to find an optimal travel route on public transport.

Keywords: *timetable information system, public transport, optimal travel route*

1. Introduction

The level of timetable information service for passengers in Russian Federation is not perfect. Even within a specific transportation system (except air transport) a customer is not provided with searching a transfer route. This problem for the railway transport is much more complicated than for the air one since the number of railway stations is hundreds of times greater than the number of airports.

The problem of searching an intermodal route (using different modes of transport in conjunction, such as ships, aircraft, etc) is of a huge complexity [1-3]. It should be noted that the transport network in Russian Federation is the largest in the world and strongly heterogeneous. There are places which can be reached by air only. There are places where water transport is preferable, and places that can be reached by railway only. Thus development of the information system allowing to unify an information from the automated systems operating for different transport modes to get a full information on tickets and a route with possible changes is badly needed.

In Europe, the mostly spread system is HAFAS [4] used on the rail transport in several countries. HAFAS provides the satisfactory search results but they are not generally optimal. The main reason is that the system search algorithm is based on the heuristic methods to narrow the search space. At first step the system uses a static algorithm with limited transport data which leads to a result generally non-optimal.

The automated system developed by authors of this paper avoid some disadvantages of the existing information systems and is based on a special search algorithm and on actual, dynamically updated information on the timetables of all kinds of transportation.

2. Automated Timetable Information System on the Public Transport

The developed timetable information system provides information obtained by Internet on the optimal route in public transport between any places within Russia. Information allows for possible change between routes, intermodal routes, timetable and tickets available. The timetable information system is a system of programs and program modules providing its operation [2].

The program system consists of the following components:

1. Specialized database for geographic information and timetables.
2. Data preparation aids for the timetable information system.
3. Aids to import data from the intermediate XML format.
4. System administration aids.
5. The core module with realization of a route search algorithm.
6. Service of integration with other information systems:
   a. Integration service operating on-line;
   b. Emulator of an EXPRESS system terminal.
7. Information Internet portal to access the information system.

The specialized database includes geographic information, regularly updated timetables of the public transport and realization of some algorithms to work with data programmed in the form of the stored T-SQL procedures. The features of the timetable information system made it necessary to develop special **data preparation aids** – a program of a transport graph visual description. The program presents Windows application to describe fragments of a transport graph. The program saves results as an XML file. For each region or transport company, we can create a separate XML file describing disposition of the transport nodal points. Thus, we can describe rail, bus or other type of a transport network by piecemeal.

After geographical data is prepared in an intermediate XML format, it is downloaded to the system by means of **the data import program**. The program processes data in XML format, connects to the system database and appends information on transport graph junctions. In details, the download process consists of the following steps: reading data from a file, data analysis, elimination of data ambiguity, the forming of an object structure, data saving. The import program consists of two functional modules:

1. Geographic data loader;
2. Timetable loader.

The algorithm to import geographic objects is programmed as follows:

1. Recognition of an object name and type;
2. Searching of objects with a similar name within a database;
3. Adding an object to the database;
4. Update of topological connections.

The algorithm to import timetables differs from object import by the following:

1. Definition of data structure;
2. Data reading into an intermediate table;
3. Data object structuring;
4. Saving structures into database;
5. Correction of supplementary data.

If a route involves a station belonging to a hyper-junction the database is added with supplementary timetable record. In the process of route timetable import, transfer subgraph is updated in a special way.

By means of graph creating program, the database has been uploaded with a number of transport graph fragments (the total number is more than 20 000 junctions). It involves the majority of railway stations and stops in Russia and abroad and the main bus stations in some Russian regions.

The information inquiry starts with route parameter determination (departure and destination points, mode of transport, travel date, etc). After a customer specified the inquiry parameters and pressed the "Search" button, the parameters are transmitted to the route search module. Using descriptors of departure and destination points, the modules receives necessary information on geographic objects. First, the algorithm tries to find a direct (non-stop) route between given points. If it fails, the system uses the heuristic algorithm to find change-over routes. After possible routes with minimal number of changes are found, the system checks if each of them satisfies additional conditions with taking into account search parameters. This step of algorithm implementation accounts for run days, border crossing and transfer duration. For each possible route, a customer is provided with information of departure and arrival time of the shortest travel.

On the final stage, the mostly preferable route is transmitted to a subsystem to interact with outer automated systems. The subsystem is intended to get information on tickets available. The ticket information is the most frequently updated part of the transport information. If throughput of the automated systems interconnection channel is limited, and data level is high, it is virtually impossible to import current ticket information from the outer automated systems. Thus, the tickets availability is checked on the final stage of the algorithm work and in the moment when the optimal route is found and it satisfies customer's criteria. The interactive subsystem is composed of:

1. The integration module within the route search module;
2. The outer systems integration service;
3. Emulator of an EXPRESS terminal;
4. The outer system interaction control program.

In the subsystem, data is searched in a cache of completed inquiries first. The cache collects the results of previous inquiries to the outer systems and keeps them for set time. If data is not found in cache, the subsystem sends a request to the outer automated systems. A way to send a request depends on features of an automated system. The request can be synchronous or asynchronous. If it is **synchronous**, the serving process performs it immediately. If the interaction is **asynchronous**, the request is queued for further service. When the queue is formed, the interaction service program starts processing it. A request from the queue is transformed into inquire in accordance with a protocol understandable for the outer automated system and is sent there. For example, to communicate with EXPRESS system, the special service is used, namely an emulator of an EXPRESS terminal. A service transforms the inquiry into BSC-3 packets, encapsulates them into TCP/IP packets and transmits them to the EXPRESS access gateway.

The gate communicates with a host PC and returns an answer to the terminal emulator. The process goes asynchronously; one inquiry may result in series of questions and answers between the emulator and the host PC. Several services can work simultaneously, and they can be run on several PCs looking through a common queue of inquiries.

To get timetable information via Internet, we have developed a portal to access the timetable information system. The portal gives information on a route by air or bus with possible changes of transport within Russia and abroad, including as follows:

- Searching for change points when there is no direct route between departure and arrival points;
- Searching the optimal travel date according to traffic days;
- Searching for routes between two points using different criteria (time in way, price and others);
- Searching for intermodal routes (with using different transport modes);
- Taking into account border crossings during route search;
- Providing information on a station timetable;
- Imaging the found routes on an interactive map;
- Imaging an interactive scheme of direct routes for a given station.

3. Algorithms for Searching Optimal Paths on Passenger Transport

The new algorithm is used in developed timetable information system. Two main approaches are well known for modelling timetable information as the shortest path problem: the time-expanded, and the time-dependent approach [5,6]. The common characteristic of both approaches is that a query is answered by applying some the shortest path algorithm to a suitably constructed graph. A time-expanded graph is constructed by assigning a specific time event (departure or arrival at a station) for every node. Edges between nodes represent either elementary connections between two events, or waiting within a station. Depending on the optimisation criterion, the construction assigns specific fixed lengths to the edges. This usually results in the construction of a huge graph. As far as time-dependent graph is concerned the idea is to refrain from the creation of a node for every event. In the time-dependent graph model based on stipulation that every node represents a station and two nodes are connected by an edge if the corresponding stations are connected by an elementary connection. The length on the edges is assigned in real time during request and depends on the time in which the particular edge will be used by the shortest path algorithm to answer the query. The timetable problem operate with objects of classes: stations, routes, arrival and departure times and traffic days schedule. The problem can be depicted by expression more formally: there are a set of routes *Z*, a set of stations *B* and a set of connections *C*. The connection is a combination of five values $C = (Z, S_1, S_2, t_d, t_a)$. This tuple is interpreted as train *Z* leaves station S_1 at time t_d, and the next stop of train *Z* is station S_2 at time t_a. A timetable is valid for a number of *N* traffic days, and every train is assigned a bit-field of *N* bits determining on which traffic day the train operates. At a station $S \in B$ it is possible to transfer from one train to another only if the time between the arrival and the departure at that station *S* is larger than or equal to a given, station-specific, minimum transfer time, denoted by transfer(*S*). Between stations that are located close to each other it is possible to walk by foot. Such data is available through the so-called foot-edges between stations. Each foot-edge is associated with a natural number representing the time in minutes needed for the walk. Formally, we treat a foot-edge like an elementary connection *c*, where the train *Z* and the departure and arrival times t_d and t_a are invalid, and length(*c*) is the associated walking time.

Let $P = (c_1, ..., c_k)$ be a sequence of elementary connections (and foot-edges) together with departure times $dep_i(P)$ and arrival times $arr_i(P)$ for each elementary connection $c_i, 1 \le i \le k$. We assume that the times $dep_i(P)$ and $arr_i(P)$ include data regarding also the departure/arrival day by counting time in minutes from the first day of the timetable. A time value *t* is of the form $t = a * 1440 + b$, where $a \in$ [0, 364] and $b \in$[0, 1439]. Hence, the actual time within a day is t (mod 1440) and the actual day is [t/1440]. Such a sequence *P* is called a consistent connection from station $A = S_1(c_1)$ to station $B = S_2(c_k)$ if it fulfils some consistency conditions: the departure station of $c_1 + 1$ is the arrival station of c_i, and the time values $dep_i(P)$ and $arr_i(P)$ correspond to the time values t_d and t_a, of the elementary connections (modulo 1440) and respect the transfer times at stations.

More formally, *P* is a consistent connection if the following conditions are satisfied:

$$c_i \; valid \; 1 \; day \, [dep_i(P)/1440],$$
$$S_2(c_i) = S_1(c_{i+1}),$$
$$dep_i(P) \equiv t_d(c_i) (\bmod 1440),$$
$$arr_i(P) = dep_i(P) + length(c_i),$$
$$dep_{i+1}(P) - arr_i(P) \geq \begin{cases} 0, & if \; Z(c_{i+1}) = Z(c_i) \; or \; c_i - foot \; edge, \\ transfer(S_2(c_i)), & othetwise. \end{cases}$$

The most well-known problem is a **problem of the earliest arrival**. The request (A, B, t_0) defines departure *A* and arrival *B* stations and departure time t_0. The goal is to minimize the difference between the arrival time and the given departure time. In another problem the goal is to find the route with the minimal number of transfers on the way between *A* and *B*. In this problem the time of arrival and departure do not matter.

Designated approaches can be used for modelling a real problem that takes into account the transfer time. To incorporate transfer times in the time-expanded model the realistic time-expanded graph is constructed as follows. Based on the time-expanded graph, for each station, a copy of all departure and arrival nodes in the station is maintained which we call transfer nodes. The stay-edges are now introduced between the transfer-nodes. For every arrival node there are two additional outgoing edges: one edge to the departure of the same train, and a second edge to the transfer-node with time value greater than or equal to the sum of the time of the arrival node and the minimum time needed to change trains at the given station. Many customers are interested in finding the cheapest connection from station A to B within a defined time interval. Unfortunately, a tariff rules in every countries are very complicated and a problem can not be solved exactly and efficiently at the same time. The multi-criteria optimisation is needed then passenger plans route using several criteria. If one need route with minimal time in way and minimal number of transfers the target is to solve MOSP (multi-objective the shortest path) problem. The problem is usually NP-hard, since the Pareto set is typically exponential in size even for simple graphs like chains. So, approximated methods are used in practice to find the solution for multi-criteria problem. One method to find Pareto optimal solution is using lexicographic order. With the simplified version of the time-expanded approach the lexicographically first solution can be computed for any *d*-tuples as edge weights. For example, if *d*=2, the first element being the travel time and the second one the number of transfers, among all fastest connections the one with the minimum number of transfers is computed. With the realistic version of the time-expanded approach only tuples can be used where the first criterion is travel time. Both approaches (time-extended and time-dependent) have been thoroughly investigated in many papers. It is evident, that finding all suitable routes taking into account time in way, ticket price, number of transfers is much more complex, than a problem of the earliest arrival. The investigations of performance for well-known methods show that performance is not enough to be used in real life tasks. The performance is a critical for timetable information systems that serve thousands requests in real time on web server.

Let enumerate some speed-up techniques that allow improving performance of pathfinder algorithm. The **angle restriction** technique uses geo data including position of nodes. At first stage during data preprocessing Dijkstra algorithm is used in order to compute all the shortest paths from every node. The results of computation are not saved, because task needs a huge amount of hard memory. Two angles for every node are saved instead $a(v, w)$ and $b(v, w)$. Let (v, w) be an edge from some event v to some other event *w*, and let S_v and S_w be the stations to which events *v* and *w* belong, respectively. Then the values $a(v, w)$ and $b(v, w)$ stored for edge (v, w) span a circle sector with centre S_v in the plane, where, say, $a(v, w)$ is the right leg. The values $a(v, w)$ and $b(v, w)$ are determined such that this circle sector has the following meaning: if the shortest path from event *v* to some event *u* at some station S_0 contains (v, w), then S_0 has to be in this circle sector.

A **station subset** technique is used in many travel route planners [6]. As a result of applying this method the number of tested nodes decreases several times. Let $G = (V, E)$ be a transport graph, $V^{\sim}$ - a set of selected nodes, *G*'- auxiliary graph built on nodes set $V^{\sim}$. Auxiliary graph is generated on pre-processing stage. After that, all requests can be computed using generated graph and resulting path lengths will be equal to lengths on transport graph. The **goal-directed search** is one more speed-up technique. This strategy is based on one real number $b(s)$ for each station s, which must be a lower bound on the minimal travel distance from s to the arrival station. Let λ – potential function (v, w) defined as $l(v,w) := l(v,w) - \lambda(v) + \lambda(w)$. The weight of edges changes to direct the search on graph to the target node.

The **hierarchical search** is similar to station subset technique, but uses several generated auxiliary graphs. Source graph $G = (V, E)$ split to l+1 level. Every level is extended with the shortest paths between some nodes [10]. The Dijkstra algorithm uses the smallest graph in hierarchy in order to find the shortest path. One more interesting is **reach** technique. Given a path P from s to t and a vertex v on P, the reach of v with respect to P is the minimum of the length of the prefix of P (the subpath from s to v) and the length of the suffix of P (the subpath from v to t). The reach of v, $r(v)$, is the maximum, over all the shortest paths P that contain v, of the reach of v with respect to P. If reaches are calculated on pre-processing stage then results can be used for pruning during search requests. Such data allows know in advance which edge to exclude.

4. Special Algorithm for Searching Optimal Passenger Route

Special algorithm that is used in timetable information system solve bi-criteria problem with minimal number of transfers and earliest arrival time [1, 2] in lexicographical order. Algorithm operates with converted data and allows solving the problem exactly. The search uses generated virtual transport graph that is similar to time-dependent model extended with auxiliary data. The advantage is auxiliary data can be recalculated locally. So, full pre-processing is not needed after some change in timetables and graph update can be done in real time. At the same time the above mentioned speed-up technique can not be used because the information about graph edges do not stored in system, but calculated in real time. Generally, Dijkstra algorithm can be applied to this graph, but the performance is not enough. For improving it special techniques were developed. **Transfer subgraph** technique is based on selecting potential transfer stations. During import source timetables, the data is analysed and subgraph is updated. Developed criteria allow making decision whether to add station to subgraph. This technique is similar to hierarchical technique but the subgraph can be updated in real time. One more effective technique is **bi-directional search balancing**. The idea of method is to make optimal decision whether to make direct or back step in bi-directional search. For special graph effective technique “**A***” can be adapted. Though it shows good practical efficiency, in general case this method does not guarantee the optimal solution. But then applied restriction on direct route lengths the method is **distance-preserving**. Distance-preserving speed-up technique has to guarantee optimal solution for problem.

Proposed approaches implemented in timetable information system “MARSHRUTY”. Adoption of new technology allows improving strongly the quality of information service for passengers; optimise public transport traffic and loading.

The pictures demonstrating the work of information system are shown on Figures 1-3.

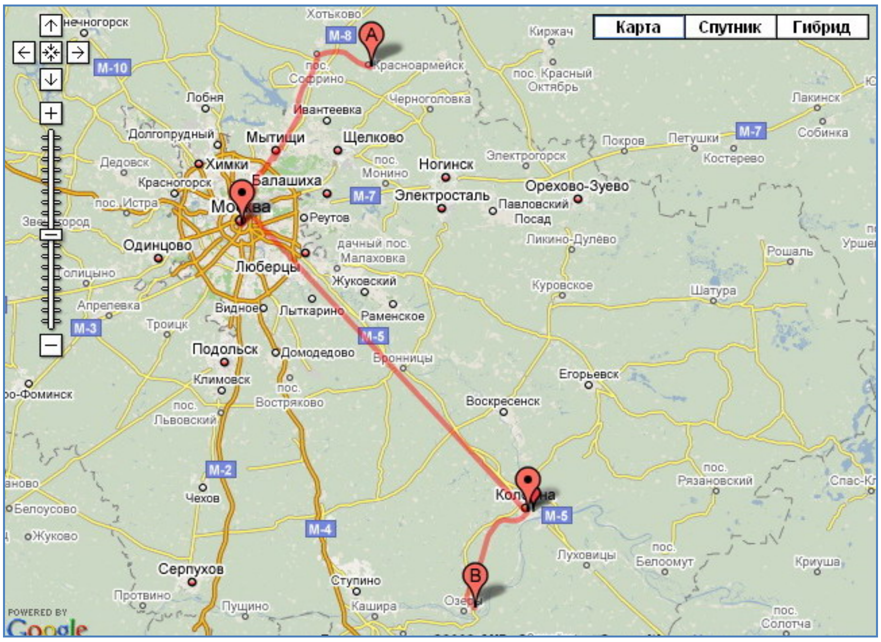

Figure 1

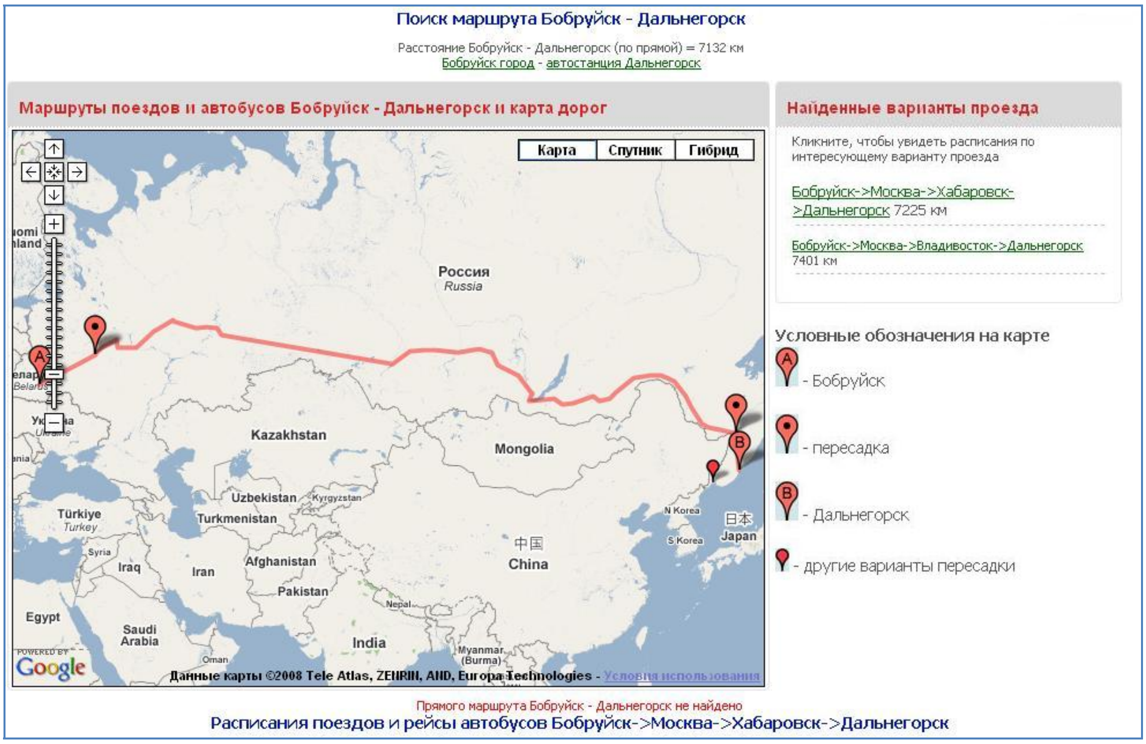

Figure 2

Figure 3

References

1. Vishnevsky, V.M., Zhelezov, R.V., Atanasova, T.N. Integrated timetable information system for railway transport in Russian Federation, *Distributed Computer and Communication Networks* (*DCCN-2005*), 2005, pp. 165-172.
2. Zhelezov, R.V. Realization of the integrated timetable information system on the base of Microsoft technologies, *Proceedings of the Conference «Microsoft Technologies in Programming Theory and Practice»*, Moscow, 2006, pp. 10-13.
3. Vishnevsky, M.V. *Theoretical design bases of computer networks*. Moscow: Technosphera, 2003. (In Russian)
4. HAFAS. A timetable information system by HaCon Ingenieurgesellschaft mbH, Hannover, Germany – http://www.hacon.de/hafas/.
5. Műller-Hannemann, M., Schulz, F., Wagner, D., Zaroliagis, C. *Timetable Information: Models and Algorithms in Algorithmic Methods for Railway Optimization*. Berlin, Heidelberg: Springer, 2007.
6. Brodal, G. S., Jacob, R. Time-dependent networks as models to achieve fast exact time-table queries. In: *Proceedings of the 3rd Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS 2003), vol. 92 of Electronic Notes in Theoretical Computer Science*. Elsevier, 2004.
7. Cooke, K.L., Halsey, E. The shortest route through a network with time-dependent intermodal transit times, *Journal of Mathematical Analysis and Applications*, Vol. 14, 1966, pp. 493-498.
8. Wagner, D., Willhalm, T. Geometric speed-up techniques for finding the shortest paths in large sparse graphs. In: *Proceedings of the 11th European Symposium on Algorithms (ESA 2003), Lecture Notes in Computer Science,* Vol. 2832, 2003, pp. 776-787.
9. Wagner, D., Willhalm, T. Speed-up techniques for shortest path computations. In: *Algorithmic Methods for Railway Optimization, Lecture Notes in Computer Science,* Vol. 4393, 2007, pp. 23-36.
10. Zhelezov, R.V. *Features of a transport route searching algorithm subject to timetable, Distributed Computer and Communication Networks. Theory and Applications (DCCN-2007).* Moscow, 2007, pp. 28-32. (In Russian)