UDC 004.421

Irina Zeleneva[1], PhD,
Sergii Zhevzhyk[2], master of Computer Engineering,
Iryna Grynevych[1]
[1]Donetsk National Technical University, Donetsk, Ukraine
[2]State Enterprice PCTB ACS RT Ukrzaliznytsia, Ukraine
irina.grinevich90@gmail.com

Development of an Algorithm for Searching Railroad Routes to Ensure Arrival at the Specified Time

The article considers the problem of developing an algorithm for searching railroad routes that ensures arrival at the specified time. The brief description of the existing railroad Internet resources in Ukraine is presented, necessary basic information and criteria for route selection is discucced. In this article the algorithm for route searching and its comparison with Dijkstra's algorithm is proposed.

Keywords: route, algorithm, Dijkstra's algorithm, graph, model, railway transport, internet resource

Introduction

In our high-speed and mobile life a man is often confronted with the need to choose the best route on a given criterion. There are widespread services to assist in the laying of roads and hiking trails. These services are widely used in computers and gadgets for automatic routing. Also human always has the ability to lay route by himself. However, our country still has no tool for the construction of optimal rail routes using computer.

Problem description

There are many difficulties when you are laying a route by yourself: absence of direct trains, absence of free places for needed trains, forced waiting during trip, etc. Existing Ukrainian internet services make it possible to search trains with given stations - endpoints, to check availability of free places and to buy tickets online. These are examples of such internet services: e-kvytok.com.ua, gd.tickets.ua, travel.tochka.net, poizd.aviakassa.net etc.

A good example of a resource for the laying of the railway (and car, bus, bicycle, pedestrian) routes with transfers and a possibility to choose one of several founded routes is a German internet resource vvs.de. This resource also allows you to book tickets, rent a car and to book accommodation.

Creating a service for laying railway lines will solve a lot of problems and will be useful to users of Ukrainian railways.

Basic data and choice criteria

Route is a path between two points (stations). Each route has a distance and travel time associated with it [1].

First of all, to find routes user should specify the required parameters: endpoint and time on which he wants to arrive at the end point. Additional parameters include total travel time, number of transfers, time between transfers, total cost of travel and other according to user's needs.

Depending on user's requirements, route selection criteria may be like: arrival time, travel time, cost, number of changes, the waiting time between transfers, and other. Criteria may be combined by "AND" / "OR". In case when a few routes were found, user may choose the optimal route to his mind.

For example, let's consider a drawing that shows the trains motion from city A to D (Fig. 1).

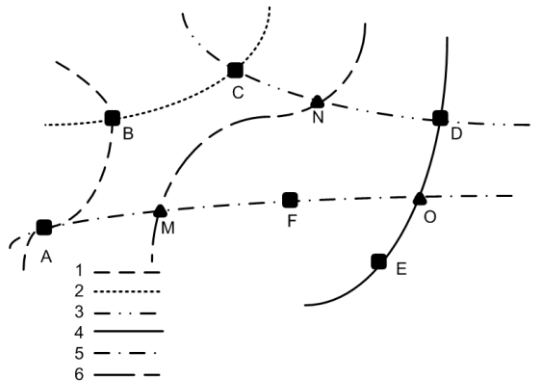

Figure 1 – Diagram of train's motion from city A to D

Let's suppose that user wants to get routes from A to D. In this example we have three possible routes:

1) train 1 (goes to B), 2 (to C) and 3 (to D);
2) train 5 (goes to O), 4 (to D) with one change on station O;

3) train 5 (goes to M), 6 (to N) and 3 (to D) with two changes on stations M and N.

Depending on the characteristics of the routes user has possibility to choose one of them on favor. The main task is to free the user from checking schedules and provide giving him finished list of possible routes with showing all the necessary information on train.

Algorithm for searching railroad routes to ensure arrival at the specified time

During the algorithm a specified number of cascades is formed and corresponding number of routes sets. A word "cascade" in this context means trains, which pass specified stations. Number of generated routes set corresponds to the number of cascade, which trains are added to routes and set number equals to trains count in route. The route contains trains from cascades, following one another. Number of cascades depends on the depth of the search. In most cases the number of transfers does not exceed two, so this number may be set by default, or user can set the value by himself. The total number of cascades and routes sets equals N+, where N is maximum count of changes.

The first step is to introduce the basic data – departure station and destination station (start and end points), the critical time of arrival. Before formation of the first set of routes the "zero" set of routes is created, which adds a new empty route with initial data (destination station and the critical time).

For creating each new set of routes the previous one is used. During the formation there are searching trains, which are going through a given station, then amount found routes are searched trains, that go through the required departure station. In case when needed trains are founded formed completed routes are added to result, in opposite case routes are added to the group of routes that will be used for next search with one new change.

Algorithm of search is shown in figure 2:

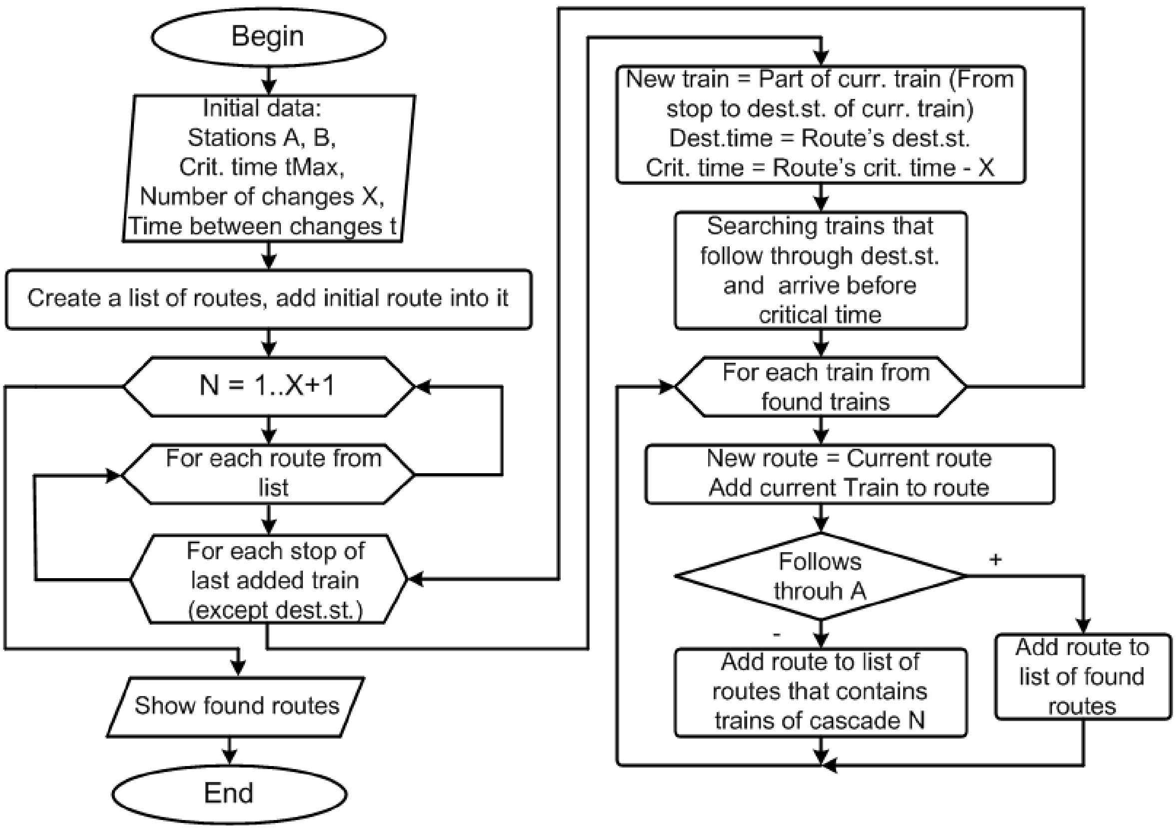

Figure 2 – The algorithm for searching railroad routes

For example let's consider drawing that shows the relation of the trains and cascades (Figure 3).

The first cascade consists of trains 3, 4, that are passing through the destination (D). In the first set we have two routes: (3) and (4).

The second cascade consists of trains 2, 5, 6, that pass stations, where trains from previous cascade makes stop. Routes count is increasing, that's why in the second set we have three routes: (2-3), (6-3), (5-4). As can be seen, the train 5 from route (5-4) passes the reference point, so the route is added to the list of filtered.

The third cascade consists of train 1 and train 5, but last one goes only to the station M, so in the third set we have two routes: (1-2-3) and (5-6-3). These routes are also added to the list of filtered.

With these data we can make up the table of relation of cascades and routes (Table 1), cascades and trains (Table 2), trains and routes (Table 3).

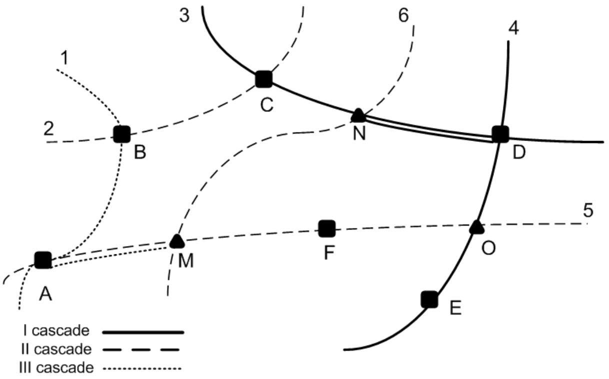

Figure 3 – Diagram of the relation of the trains and cascades

Table 1

Routes	Cascades		
	I	II	III
(1-2-3)	1	1	1
(5-6-3)	1	1	1
(5-4)	1	1	

Table 2

Trains	Cascades		
	I	II	III
1			1
2		1	
3	1		
4	1		
5		1	1
6		1	

Table 3

Trains	Routes		
	(1-2-3)	(5-6-3)	(5-4)
1	1		
2	2		
3	3	3	
4			2
5		1	1
6		2	

The nuances of software code development

Software implementation of the algorithm requires presence of specialized models: routes, trains, changes, etc.

The route model may contain an array of trains, time of departure and time of arrival, departure and destination stations, the critical time and an array of changes.

The train model may contain time of departure and time of arrival, departure and destination stations and an array of stops.

The stop model may contain a reference to train, station, where this train makes stop, time of stop and time of departure.

The change model may contain references to trains, first train arrival time, second train departure time, station where this change may be done.

The proposed algorithm will be implemented with programming language C#. Developed models will become a part of automated supporting system of reflected railway transport models. Such system is a specialized software for object-relational mapping objects. It means that for connection of models there may be used links which make easier transition from one model element to another [3].

Comparison of the algorithm with the existing analogue

For laying routes such algorithms as Dijkstra's algorithm, Lee algorithm, k-nearest neighbor algorithm and other are most often used. However, in case of searching exactly railroad routes taking into account their specific may be useful only Dijkstra's algorithm with condition that as distance to the point may be taken cost or time of a ride.

In general case, Dijkstra's algorithm is based on assigning to point some weight coefficients, and besides coefficient of point gives value of distance from reference point to current. Gradually these coefficients are decreasing with some iterative procedure and on each iteration step only one coefficient may become permanent. The latter indicates that coefficient is not upper bound anymore and equals minimum distance to current point [2].

Let us consider the figure, where trains motion from city A to D from previous examples is represented as a graph (Figure 4):

When applying Dijkstra algorithm during each iteration there is not only calculating distance from one point to another, but also there is a comparison of this distance with other possible values.

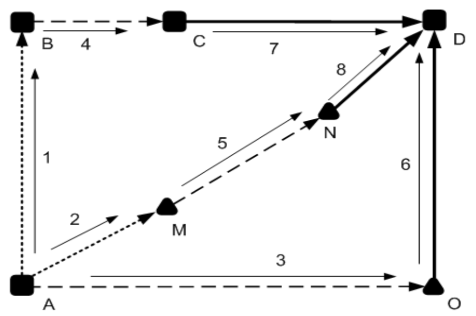

Figure 4 – Graph for Dijkstra's algorithm with cascades from developed algorithm

In this case the total number of iterations equals eight. It should be mentioned that significant disadvantage of Dijkstra's algorithm is that there must be ready graph with all points. So, this algorithm requires the prior formation of a graph through other means.

Conclusion

In contrast to Dijkstra's algorithm, algorithm for searching railroad routes does not require recalculation of the distance to a point for each iteration, and therefore has a smaller number of execution steps. Besides, during algorithm execution there is an automatically formed graph, which contains a certain number of cascades of trains. A user-specified search depth not only helps to find routes according to user's wishes, but also guarantees that graph will not be overloaded with unnecessary information.

The proposed search algorithm ensures the arrival at destination point at the specified time. The algorithm may be taken as the basis for development of corresponding Internet service. Currently, similar services for train routes searching are not present in Ukraine. Therefore, creation of such service can help to solve a lot of problems of Ukrainian railway users.

References

1. Train Routing Algorithms: Concepts, Design Choices, and Practical Considerations. [L. Anderegg, S. Eidenbenz, M. Gantenbein, C. Stamm etc.] // Proceedings 5th Workshop on Algorithm Engineering and Experiments (ALENEX03), 2003.

2. Dijkstra's algorithm. Material from internet university INTUIT [Электронный ресурс]. - Access mode: http://www.intuit.ru/department/algorithms/ingrth/9/

3. Iryna Grynevych. Supporting system of reflected models / Iryna Grynevych, Sergii Zhevzhyk, Irina Zeleneva // Electronic collection of works of VII International Scientific Conference of Students and Young Scientists [«Computer science and computer technology (ICT-2011)»]. – Donetsk: Donetsk National Technical University, 2011. - P. 22-26.

Надійшла до редакції 10.04.2012

І.Є. ГРИНЕВИЧ[1], С.Є. ЖЕВЖИК[2], І.Я. ЗЕЛЕНЬОВА[1]
[1] Донецький національний технічний університет
[2] ДП ПКТБ АСУ ЗТ «Укрзалізниця»

Розробка алгоритму пошуку залізничних маршрутів, що гарантують прибуття до заданого часу

У статті розглянута задача розробки алгоритму пошуку залізничних маршрутів, що гарантують прибуття до заданого часу. Наведено короткий опис стану існуючих залізничних інтернет-ресурсів України. Наведено основні необхідні дані і критерії для вибору маршруту. Наведено пропонований алгоритм пошуку маршрутів, а також порівняння алгоритму з алгоритмом Дейкстри.

Ключові слова: маршрут, алгоритм, алгоритм Дейкстри, граф-схема, модель, залізничний транспорт, інтернет-ресурс

И.Е. ГРИНЕВИЧ[1], С.Е. ЖЕВЖИК[2], И.Я. ЗЕЛЕНЁВА[1]
[1] Донецкий национальный технический университет
[2] ГП ПКТБ АСУ ЖТ «Укрзализныця»

Разработка алгоритма поиска железнодорожных маршрутов, гарантирующих прибытие к указанному времени

В статье рассмотрена задача разработки алгоритма поиска железнодорожных маршрутов, гарантирующих прибытие к заданному времени. Кратко описано состояние железнодорожных интернет-ресурсов Украины. Приведены необходимые данные и критерии для выбора маршрута. Предложен алгоритм поиска маршрутов, а также выполнено сравнение алгоритма с алгоритмом Дейкстры.

Ключевые слова: маршрут, алгоритм, алгоритм Дейкстры, граф-схема, модель, железнодорожный транспорт, интернет-ресурс