

УДК 004.421

РАЗРАБОТКА АЛГОРИТМА ПОИСКА ЖЕЛЕЗНОДОРОЖНЫХ МАРШРУТОВ ПО КРИТЕРИЮ ВРЕМЕНИ

Гриневич И.Е.¹, Жевжик С.Е.², Зеленева И.Я.¹

¹ Донецкий национальный технический университет, Украина;

² Укрзалізниця ГП ПКТБ АСУ ЖТ, г. Донецк, Украина

В статье рассмотрена задача разработки алгоритма поиска железнодорожных маршрутов, гарантирующих прибытие к заданному времени. Приведено краткое описание состояния существующих железнодорожных интернет-ресурсов Украины. Приведены основные необходимые данные и критерии для выбора маршрута. Приведен предлагаемый алгоритм поиска маршрутов, а также существующие алгоритмы, которые могут быть использованы для оптимизации предлагаемого.

Введение

В наш скоростной и мобильный век, вне зависимости от преследуемых целей, человек сталкивается с необходимостью выбрать маршрут, оптимальный по тому или иному критерию. Широко распространены сервисы, призванные помочь в прокладке автомобильных и пешеходных маршрутов. Такие сервисы широко используются в компьютерах и гаджетах для автоматической прокладки маршрутов. Также человек имеет возможность проложить маршрут самостоятельно. Однако, в нашей стране все ещё не реализован сервис для прокладки оптимальных железнодорожных маршрутов.

Описание проблемы

При самостоятельной прокладке железнодорожного маршрута есть много сложностей: отсутствие прямых поездов, отсутствие свободных мест на конкретные поезда, вынужденные ожидания во время пересадок, и др. Существующие украинские интернет-сервисы допускают возможность поиска поезда с учетом указанных конечных станций, проверку наличия свободных мест и заказ билетов. Примерами таких сервисов являются: e-kvytok.com.ua, gd.tickets.ua, travel.tochka.net, poizd.aviakassa.net и др.

Хорошим примером ресурса для прокладки железнодорожных (а также автомобильных, автобусных, велосипедных, пешеходных) маршрутов с пересадками и возможностью выбора маршрута из нескольких найденных является немецкий интернет-ресурс vvs.de. Данный ресурс также позволяет заказать билеты, бронировать жилье.

Создание сервиса для прокладки железнодорожных маршрутов решит множество существующих проблем и будет полезно пользователям украинских железных дорог.

Основные данные для прокладки маршрута

Маршрут – это путь между двумя точками (станциями). Каждый маршрут имеет

расстояние и время в пути, связанные с ним [1].

Прежде всего, для поиска маршрутов необходимо указание обязательных параметров: конечных точек и времени, до которого необходимо прибыть в конечную точку. К дополнительным параметрам можно отнести общее время в пути, количество пересадок, время между пересадками, общую стоимость проезда.

Критерии выбора маршрута

В зависимости от нужд пользователя, критерии выбора маршрута могут быть следующими: время прибытия, время в пути, стоимость, количество пересадок, время ожидания между пересадками, и др. Критерии могут быть объединены условиями «И»/«ИЛИ». В случае отбора нескольких маршрутов, пользователю предоставляется выбор оптимального на его взгляд маршрута.

Для примера рассмотрим рисунок, показывающий движение поездов из города А в D (рис. 1).

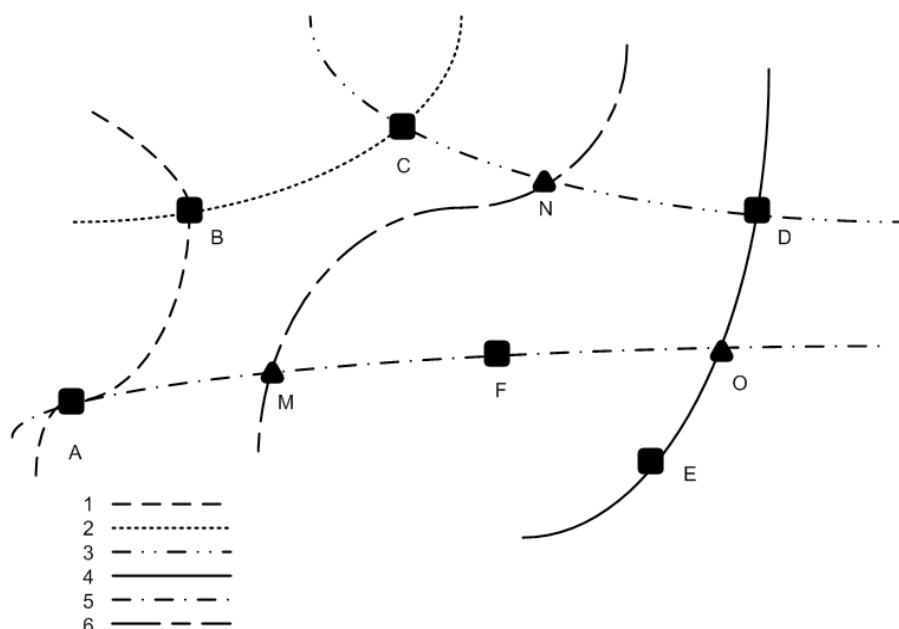


Рисунок 1. Схема движения поездов между станциями

Предположим, пользователю необходимо проложить маршрут из города А до города D. В данном примере у нас есть два возможных варианта:

- 1) ехать поездами 1 (до города В), 2 (до города С), 3 (до города D), совершая две пересадки в городах В и С;
- 2) ехать поездами 5 (до станции О), 4 (до города D), совершая одну пересадку на станции О;
- 3) ехать поездами 5 (до станции М), 6 (до станции N), 3 (до города D), совершая две пересадки в городах М и N.

В зависимости от характеристик маршрутов пользователь делает выбор в пользу одного из них. Главной задачей является избавление пользователя от необходимости сверки расписаний и предоставление готового списка возможных маршрутов с указанием всей необходимой информации по поездам.

Алгоритм поиска железнодорожных маршрутов, гарантирующих прибытие к заданному времени

Задачей алгоритма является формирование заданного количества каскадов поездов и соответствующего количества наборов маршрутов. Номер формируемого набора маршрутов соответствует номеру каскада поездов, добавляемых к маршрутам, т.е. номер набора равен количеству поездов. Количество каскадов зависит от глубины поиска. В подавляющем большинстве случаев количество пересадок не превышает двух, поэтому именно такое количество пересадок может быть задано по умолчанию, или же пользователь может задать свое значение. Всего может быть сформировано $N+1$ каскадов и наборов маршрутов, где N – максимальное количество пересадок.

Первым шагом является введение основных данных – станции направления и станции назначения (начальной и конечной точки), критического времени прибытия.

Перед формированием первого набора маршрутов создается «нулевой» набор маршрутов, куда добавляется новый полупустой маршрут с начальными данными (станция назначения и критическое время).

При формировании очередного набора маршрутов используется предыдущий. В ходе формирования осуществляется поиск поездов, идущих через заданную станцию, затем среди найденных – поиск проходящих через нужную станцию отправления. В случае нахождения поездов сформированные маршруты добавляются к найденным, в противном случае маршруты добавляются к группе маршрутов, которая будет использоваться для следующего поиска с учетом новой пересадки.

Алгоритм поиска маршрутов приведен на рис. 2.

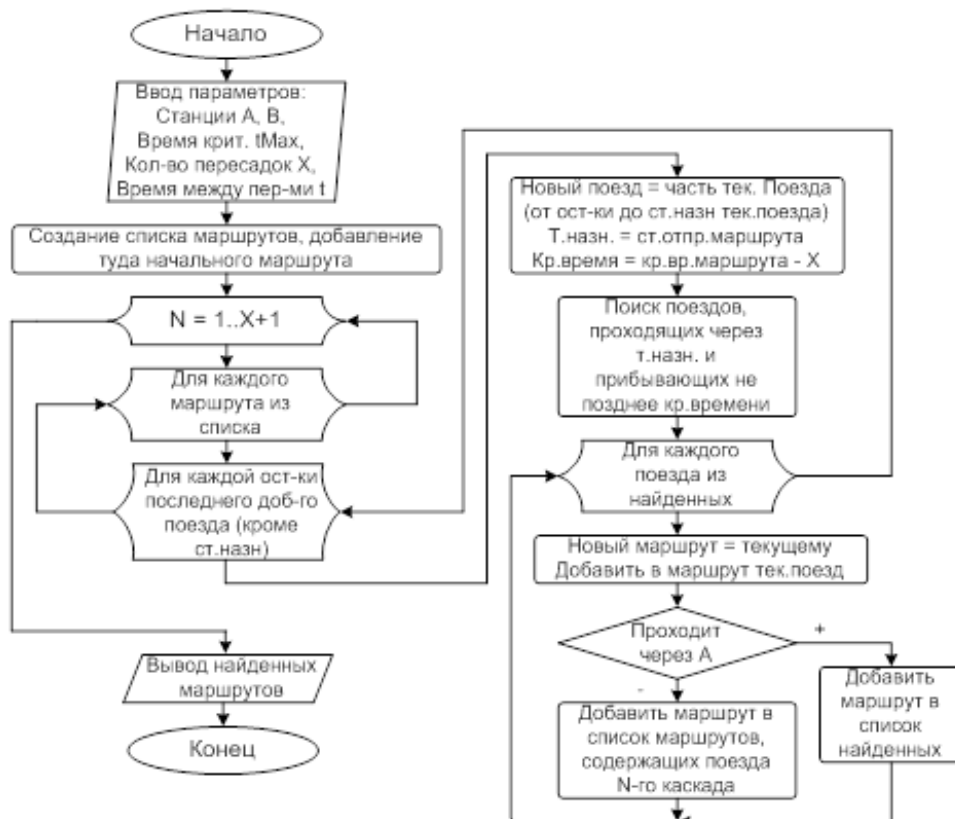


Рисунок 2. Алгоритм поиска маршрутов, гарантирующих прибытие к заданному времени

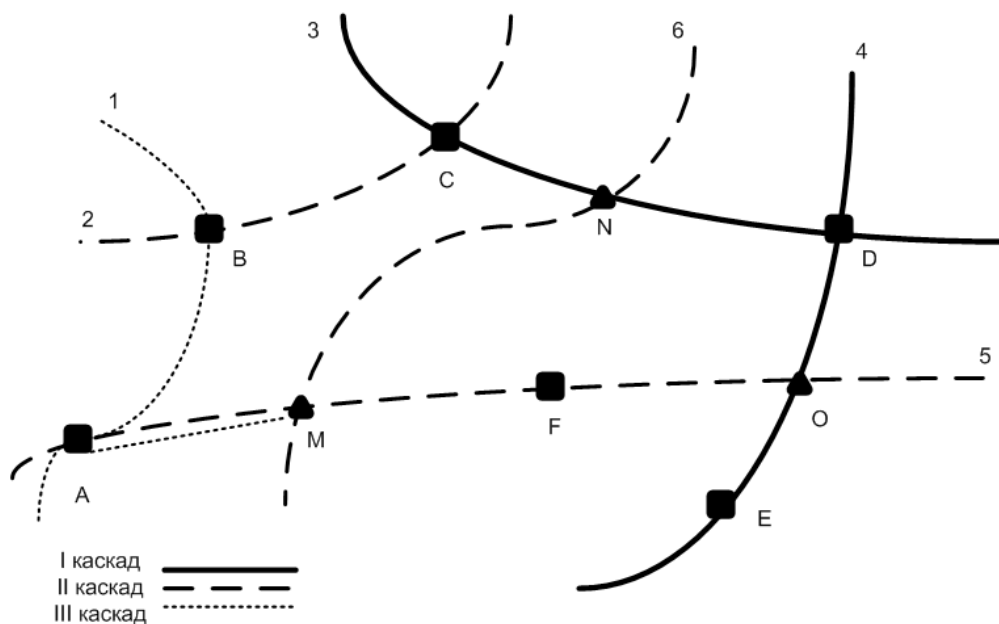


Рисунок 3. Принадлежность поездов каскадам

Для примера рассмотрим рисунок, показывающий принадлежность поездов каскадам (рис. 3).

Первый каскад состоит из поездов 3, 4, идущих через пункт назначения (D). В первом наборе имеем два маршрута: (3) и (4).

Второй каскад состоит из поездов 2, 5, 6, идущих через остановки поездов первого каскада. Количество маршрутов увеличивается, потому во втором наборе имеем уже три маршрута: (2-3), (6-3), (5-4). Как видно, поезд 5 маршрута (5-4) проходит через исходную точку, поэтому маршрут добавляется в список отфильтрованных.

Третий каскад состоит из поезда 1 и поезда 5, но идущего лишь до станции M, поэтому в третьем наборе имеем два маршрута: (1-2-3) и (5-6-3). Данные маршруты также добавляются в список отфильтрованных.

На основе полученных данных могут быть скомпонованы таблица связи каскадов с маршрутами (табл. 1), таблица принадлежности поездов каскадам (табл. 2), и таблица принадлежности поездов маршрутам с учетом порядка следования (табл. 3).

Нюансы разработки программного кода

Программная реализация алгоритма требует наличия специализированных моделей: маршрутов, поездов, наборов групп.

Модель маршрута может содержать массив поездов, время отправления (самого раннего поезда) и прибытия (самого позднего), а также станции отправления и назначения, критическое время, массив пересадок.

Модель поезда может содержать время отправления и прибытия, станции отправления и назначения, массив остановок.

Модель остановки может содержать ссылку на поезд, станцию остановки, время остановки поезда на станции и время его отправления.

Модель пересадки может содержать ссылки на поезда (с какого поезда на какой необходима пересадка), время прибытия первого поезда, время отправления второго, станцию пересадки.

Таблица 1 – Связь каскадов с маршрутами

Маршруты	Каскады		
	I	II	III
(1-2-3)	1	1	1
(5-6-3)	1	1	1
(5-4)	1	1	

Таблица 2 – Принадлежность поездов каскадам

Поезда	Каскады		
	I	II	III
1			1
2		1	
3	1		
4	1		
5		1	1
6		1	

Таблица 3 – Принадлежность поездов маршрутам

Поезда	Маршруты		
	(1-2-3)	(5-6-3)	(5-4)
1	1		
2	2		
3	3	3	
4			2
5		1	1
6		2	

Применение алгоритмов для оптимизации алгоритма поисков маршрутов

В целях оптимизации предлагаемого алгоритма поиска маршрутов могут также применяться существующие алгоритмы. К наиболее часто используемым относятся такие, как алгоритм Дейкстры, метод ближайшего соседа, алгоритм Ли. Однако, в случае отбора именно железнодорожных маршрутов с учетом их специфики, из перечисленных может быть полезным лишь алгоритм Дейкстры [2].

Алгоритм Дейкстры (Dijkstra's algorithm) — алгоритм, используемый для нахождения кратчайшего пути. В общем случае этот метод основан на приписывании вершинам временных весовых коэффициентов, причем коэффициент вершины дает верхнюю границу длины пути от некоторой вершины S к рассматриваемой вершине. Эти коэффициенты постепенно уменьшаются с помощью некоторой итерационной процедуры, и на каждом шаге итерации только один из временных коэффициентов становится постоянным. Последнее указывает на то, что коэффициент уже не является верхней границей, а дает точную длину кратчайшего пути к рассматриваемой вершине [2].

Данный алгоритм может быть применен для поиска маршрута с наименьшим временем и/или наименьшей стоимостью, что подразумевает внесение определенных корректив в алгоритм поиска маршрутов по времени, предложенный в данной статье.

Выводы

В статье рассмотрено возможное решение проблемы отсутствия сервисов для прокладки железнодорожных маршрутов, допускающих пересадки. Предлагаемый алгоритм поиска железнодорожных маршрутов гарантирует пользователю прибытие в пункт назначения не позднее заданного времени. Дальнейшие разработки

предусматривают оптимизацию предложенного алгоритма по критериям время/стоимость, а также разработку соответствующего программного комплекса, обладающего свойством кроссплатформенности.

Список источников

- [1] Train Routing Algorithms: Concepts, Design Choices, and Practical Considerations. L. Anderegg, S. Eidenbenz, M. Gantenbein, C. Stamm, D.S. Taylor, B. Weber, P. Widmayer. Proceedings 5th Workshop on Algorithm Engineering and Experiments (ALENEX03), 2003.
- [2] Алгоритм Дейкстры. Материал из интернет-университета ИНТУИТ. [Электронный ресурс] – Режим доступа: <http://www.intuit.ru/department/algorithms/ingrth/9>