

Оптимизация маршрутов на дорожной сети

77-48211/369475

05, май 2012

Степанов В. П.

УДК 681.3

Россия, МГТУ им. Н.Э. Баумана

vapals@yandex.ru

Постановка задачи. Известными для решения задачи являются: дорожная сеть города, место отправления и место назначения. Необходимо выбрать оптимальный и все близкие к нему маршруты проезда из места отправления до места назначения, которые учитывают реальную обстановку на дорогах: возможные заторы на дорогах и варианты их объезда, задержки перед светофорами на перекрестках, различные скорости движения транспорта на отдельных участках дорожной сети города.

В качестве критерия для выбора маршрута может служить время, пройденный путь и т. д. В реальности кратчайший по расстоянию путь не всегда является оптимальным по времени, расходу топлива и т.п. Свести эти критерии к одному весьма затруднительно. Часто критерий оптимальности пути достаточно трудно формализовать. С другой стороны, любая математическая модель не учитывает многих внешних факторов, влияющих на результат, так как ситуация на дороге может изменяться. Поэтому рассматривается не только оптимальный путь, но и все близкие к оптимальному маршруты в заданном диапазоне, чтобы пользователь при окончательном выборе смог выбрать наиболее подходящий.

Существует ряд реализованных программ, позволяющих искать кратчайший путь проезда по дорожной сети на электронной карте [1-5]. В рассмотренных системах используются различные критерии оптимальности пути - от критерия кратчайшего расстояния до сложных критериев оценки времени с учетом информации о пробках. В отмеченных работах в основном рассматриваются алгоритмы поиска только оптимального пути. Только в работе [6] описан алгоритм поиска **K** маршрутов отклонения от оптимального на основе оптимального маршрута, проходящего через **K** вершин графа.

Математическая модель. Множество всех возможных трасс поездки по улицам города представляется в виде ориентированного графа $G = (A, W)$, где A – множество вершин, W – множество дуг. Вершинам этого графа соответствуют: перекрестки на улицах города, место отправления $i \in A$ и место назначения $j \in A$. Вершины графа – это места дорожной сети, где имеются возможности выбора дальнейшего маршрута поездки по городу. Ребрам графа соответствуют магистрали и улицы между двумя вершинами. Для ребер графа задаются матрица расстояний $L = |l_{sd}|$ и матрица возможных скоростей движения $C = |c_{sd}|$, $s, d \in A$. Для каждой вершины графа $s \in A$ с учетом наличия или отсутствия светофора задаются значения z_s – время задержки на перекрестке. Тогда t_{sd} – время движения по ребру (s, d) определяется по формуле

$$t_{sd} = l_{sd} / c_{sd} + z_s, \quad s, d \in A. \quad (1)$$

Для заданных начальных и конечных вершин графа i и j требуется определить маршрут проезда R_{ij} , затрачивающего минимальное время, а также множество всех близких к оптимальному маршрутов, которые отличаются от оптимального на заданную величину E [7, 8]. Определение множества всех близких к оптимальному маршрутов позволяет при окончательном выборе учесть дополнительные неформализованные требования.

Алгоритм решения. Алгоритм решения задачи состоит из двух этапов. Определения кратчайшего по времени маршрута проезда R_{ij} сводится к решению известной задачи кратчайшего пути на ориентированном графе. Для ее решения применяются известный алгоритм [6], основанный на расстановке пометок на вершинах графа. Для определения множества всех близких к оптимальному маршрутов применяется алгоритм, основанный на методе последовательного анализа вариантов [9] и использовании правила отбраковки бесперспективных вариантов до получения окончательного решения.

Как известно, в алгоритме Дейкстры для поиска кратчайшего пути вершинам графа приписывают временные или постоянные пометки. Пометки определяют для вершины верхнюю границу длины пути от i вершины до текущей. Величины временных пометок вершин постепенно уменьшаются. Значение пометки определяет возможную длину пути от начальной до этой вершины. На каждом шаге алгоритма только одна из пометок с минимальным значением на рассматриваемом уровне выбирается в качестве постоянной. Это значит, что значение пометки является длиной кратчайшего пути из i вершины в текущую вершину.

Введем следующие обозначения: $V(s)$ – множество ребер, входящих или выходящих из вершины s ($V(s) \subseteq W$); $|V(s)|$ – мощность множества; p_{sj} – пометки вершины $s \in A$, $j = 0, 2, \dots, |V(s)|-1$; B – множество вершин с постоянными пометками для $j = 0$.

Алгоритм первого этапа поиска оптимального маршрута состоит из шести шагов:

Шаг 1. Присвоить пометке начальной вершины $p_{i0} = 0$ и считать постоянной. Для всех остальных вершин $s \in A \setminus \{i\}$ установить $p_{s0} = \infty$ и считать эти пометки временными. Установить $d = i$; $B = \{i\}$. Обновить метки.

Шаг 2. Для всех вершин $s \in V(d)$ вычисляются новые значения

$$p_{sj} = p_{dj} + t_{ds}, \quad j = 0, 2, \dots, |V(s)|-1. \quad (2)$$

Таким образом, для каждой вершины будет определено время проезда для всех возможных путей от исходной вершины i до вершины s .

Для $j = 0$ временные пометки вычисляются, используя выражение

$$p_{s0} = \min[p_{s0}, (p_{d0} + t_{ds})] \quad (3)$$

и превращаются эти пометки в постоянные.

Шаг 3. Среди всех вершин с временными пометками $s \in A \setminus B$ найти такую вершину k , для которой значение пометки минимально $p_{k0} = \min p_{s0}$.

Шаг 4. Считать пометку p_{k0} постоянной и установить $d = k$. В множество B добавить вершину k .

Шаг 5. Если $d \neq j$, то перейти к шагу 2. Если $d = j$, то p_{k0} является длиной кратчайшего пути R_{ij} из вершины i в вершину j .

Шаг 6. Если все пометки всех вершин постоянные, т.е. $B = A$, то на этом определение времени оптимального пути завершается.

После этого происходит восстановление маршрута проезда.

На втором этапе алгоритма задается допустимое значение отклонения от оптимального значения E . На первом этапе, в отличие от алгоритма Дейкстры, для каждой вершины в зависимости от мощности множества $V(s)$ вычисляются по формуле (2) и запоминается не одно, а ряд значений пометок. Затем для каждой вершины отбрасываются те значения пометок, для которых выполняется соотношение

$$p_{sj} > (p_{k0} + E), \quad s \in V(d), \quad j = 0, 2, \dots, |V(s)|-1. \quad (4)$$

Рис. 1. Схема работы алгоритма

Карта улиц города Москвы загружается из внешнего файла. Исходные данные генерируются на основе данных программы Mosmap [5]. Затем с помощью указателя мыши на перекрестках улиц назначаются вершины графа. При этом имеется возможность ввода характеристики вершин графа – перекрестков: название и время задержки. После этого указывается схема соединений вершин графа между собой для дорожной сети города. Далее клавишами мыши указываются начальная и конечная вершины ребра. С помощью контекстного меню можно задать характеристики дорог - ребер графа: название улицы, длина, средняя скорость движения. ПК дает возможность оперативного изменения характеристик вершин и ребер графа на карте города, а также показать карту города с требуемой подробностью. Главное окно ПК при его запуске имеет вид, приведенный на рис. 2.

В ПК реализованы три алгоритма: алгоритм Дейкстры, алгоритм нахождения **К** кратчайших маршрутов и предлагаемый алгоритм нахождения всех **Е** близких к оптимальному маршрутов.

В верхней части окна расположено главное меню и кнопка «Режим редактирования». В центре окна находится карта, а справа - окно с закладками «Результаты» и «Свойства». В расположенном внизу окна статус строке отображается информация о находящейся рядом с курсором мыши маршруте и времени поиска оптимальных маршрутов.

Главное меню ПК содержит следующие пункты:

- 1 Файл – «Загрузить» - открывает диалоговое окно для выбора файла карты.
- 2 Файл – «Сохранить» - сохраняет текущий граф карты.
- 3 Вид – «Очистить маршрут» - стирает отображенный на карте маршрут.
- 4 Вид – «Показывать расстояния» - для каждого участка дороги отображает расстояние или время движения с заданной на этом участке скоростью.

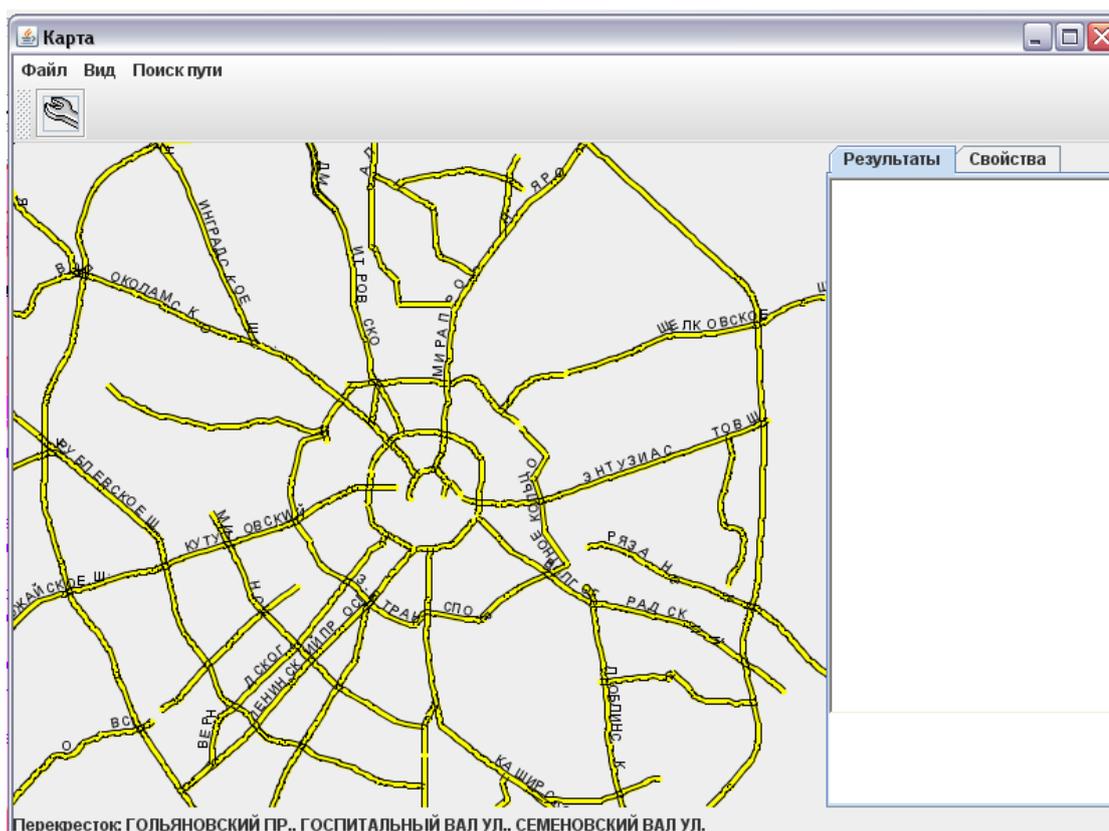


Рис. 2. Главное окно программного комплекса

- 5 Поиск пути – «Переключатель расстояние» - при выбранном переключателе параметром отображения на карте или поиска путей является расстояние.
- 6 Поиск пути – «Переключатель время»- при выбранном переключателе параметром отображения на карте или поиска путей является время.
- 7 Поиск пути – «Алгоритм Дейкстры» - ищет кратчайший маршрут между 2-мя точками, поставленными на карте.
- 8 Поиск «**К** кратчайших путей» - выводит диалоговое окно для запроса требуемого количества маршрутов и производит поиск маршрутов между двумя точками.
- 9 Поиск пути – «**Е** близкие к оптимальному маршруты» - выводит диалоговое окно для запроса значение **Е** и производит поиск маршрутов между двумя точками.

Кнопка «Режим редактирования» служит для перевода карты в режим редактирования - поиска маршрута. В режиме поиска маршрута правой кнопкой мыши определяется начальный пункт на карте, выделяемая синим кружком, а левой кнопкой мыши – конечный пункт поиска, отображаемая на карте зеленым кружком. В режиме редактирования пользователь может выбрать дорогу для редактирования, которая отображается на

карте зеленым цветом. Все характеристики дорог можно изменять. Чтобы изменения были доступны для последующих запусков программы, их нужно сохранить в файле.

Вычисленные возможные варианты маршрутов проезда между этими пунктами различными цветами выделяются на карте. На рис. 3 приведен пример представления результатов поиска маршрутов между двумя перекрестками для дорожной сети г. Москвы.



Рис. 3. Представление результатов поиска маршрутов

Проведенные расчеты и анализ полученных результатов. С помощью разработанного ПК проведены расчеты с целью исследования производительности алгоритмов и зависимости скорости поиска маршрутов от входных данных. При проведении всех расчетов задавались параметры: $K = 5$, $E = 1$ км.

Расчеты зависимости времени поиска маршрутов от числа вершин в кратчайшем пути проводились на карте города Москвы, представляющей собой граф с 12 214 вершинами и 35 598 ребрами. Была выбрана одна начальная вершина и несколько конечных вершин на разных расстояниях от начальной вершины. Для каждой такой пары вершин произведен поиск маршрутов с помощью трех алгоритмов, причем каждый из алгоритмов запускался несколько раз, а время поиска, измеряемое в миллисекундах, было усреднено.

В таблице 1 приведены результаты расчетов, выполненных для центра города. Начальный пункт выбирался на пересечении улицы Врубеля и Малого Песчаного переулочка, а конечный пункт назначался на пересечениях:

- Бумажного проезда и 5-й улицы Ямского Поля (6.6 км, 35 вершин);

- Цветного бульвара и Садово-Сухаревской улицы (9.2 км, 67 вершин);
- Басманного и Рязанского переулков (11.6 км, 88 вершин);
- Красноказарменной улицы и Энергетического проезда (15.7 км, 117 вершин);
- Городецкой улицы (33.4 км, 172 вершины).

Табл.1. Зависимость времени поиска от числа вершин графа

Число вершин графа	35	67	88	117	172
Алгоритм Дейкстры	0,01	0,03	0,13	0,24	0,27
К маршрутов	3,94	7,41	27,34	66,41	117,63
Е близкие	0,36	0,99	4,37	5,78	6,93

Результаты расчетов показывают, что время поиска всех **Е** близких к оптимальному маршрутов имеет хорошую производительность. Эффективность алгоритма объясняется тем, что дорожная сеть Москвы представляется весьма неполным графом. Поэтому отбраковка заведомо неоптимальных маршрутов происходит на ранних шагах построения вариантов. При этом полученное оптимальное решение позволяет эффективно отбрасывать те частичные решения, для которых не выполняются условия (4). Значительное возрастание времени поиска **К** маршрутов происходит потому, что алгоритм осуществляет поиск всех возможных отклонений по вершинам предыдущего маршрута и только после этого выбирает нужное число маршрутов.

Для реализованных алгоритмов исследовались зависимости времени поиска маршрутов от размерности графа. Для проведения этих расчетов использованы четыре карты Москвы разного размера:

- весь город – граф с 12214 вершинами и 35598 ребрами;
- центр города внутри третьего транспортного кольца – граф с 4756 вершинами и 14446 ребрами;
- центр города в пределах Садового кольца – граф с 2002 вершинами и 6116 дугами;
- южная часть центра города – граф с 186 вершинами и 453 дугами.

Для расчетов выбраны две вершины, присутствующие на всех четырех картах. Произведен поиск пути между этими вершинами на каждой из четырех карт с помощью трех алгоритмов. Усредненное время поиска маршрутов при проведенных многочисленных расчетах, измеряемое в секундах, сведено в таблицу 2.

Табл. 2. Зависимость времени поиска маршрутов от размерности графов

Число вершин графа	186	2002	4756	12214
Алгоритм Дейкстры	0,005	0,006	0,009	0,012
К маршрутов	0,056	0,41	0,954	2,348
Е близкие	0,058	0,28	0,505	0,548

Полученные расчеты показывают, что время работы всех трех алгоритмов зависит нелинейно от числа вершин графа. Результаты расчетов показывают, что изложенный в работе алгоритм нахождения всех **Е** близких к оптимальному маршрутов является весьма эффективным для графов значительных размеров.

Выводы.

1. Предложена математическая модель задачи выбора оптимального и всех близких к нему маршрутов в виде задачи дискретного программирования и алгоритм ее решения. Определение всех близких к оптимальному маршрутов позволяет при окончательном выборе маршрута учесть дополнительные неформализованные требования.
2. Предложенный алгоритм, являющийся модификацией известного алгоритма Дейкстры, реализован в виде программного комплекса на языке JAVA. Результаты проводимых расчетов представляются в удобном виде на электронной карте города.
3. Проведенные расчеты подтверждают, что алгоритм Дейкстры более эффективен для поиска оптимального пути. Предложенный алгоритм поиска всех близких к оптимальному маршрутов показывает хорошую производительность для графов дорог достаточно больших размеров.

Литература

1. <http://www.mapquest.com/routeplanner:Find the shortest path from A to B to Z>.
2. <http://www.pocketgis.biz>: Автоматическая трассировка маршрутов на электронной карте.
3. <http://www.mobimap.ru/item/2139302137/mobimap-moscow.jad>: Карта города для телефонов с Java.
4. <http://autosputnik.com/products/autosputnik/list/>: Электронная карта: Спутниковая навигация.
5. <http://mosmap.ru/software/gis-mosmap-integrator.htm>: Электронные карты городов России и планирование маршрутов.

6. Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир, 1978. – 432 с.
7. Степанов В.П. О математическом моделировании дорожной сети города для выбора маршрута проезда: Тез. док. науч. конф. МГТУ им. Н.Э.Баумана.- М, 2005, с.110-111.
8. Степанов В.П. О математическом моделировании дорожной сети. // Новые информационные технологии в автоматизированных системах: Материалы тринадцатого научно-практического семинара. – МИЭМ. М., 2010, с. 237 – 243.
9. Моисеев Н.Н. Численные методы в теории оптимальных систем. – М.: Наука, 1971. – 424 с.

Optimization of routes on road network

77-48211/369475

05, May 2012

Stepanov V.P.

Russia, Bauman Moscow State Technical University
vapals@yandex.ru

The author proposes an algorithm of solving the problem of determination of optimal and all near-optimal routes from one point to another. The problem was turned into determination of multiple paths between two nodes in the graph representing the city. A modification of the Dijkstra algorithm was proposed to solve the problem. The algorithm was implemented in JAVA as a software system. The calculations carried out by means of a program complex showed the efficiency of the algorithm when solving good-sized problems.

Publications with keywords: [optimization](#), [algorithms](#), [routes](#), [roads](#), [the program](#)

Publications with words: [optimization](#), [algorithms](#), [routes](#), [roads](#), [the program](#)

References

1. *Find the shortest path from A to B to Z*. Available at: <http://www.mapquest.com/routeplanner> .
2. *Avtomaticheskaia trassirovka marshrutov na elektronnoi karte* [Automatic tracing of routes on an electronic map]. Available at: <http://www.pocketgis.biz> .
3. *Karta goroda dlia telefonov s Java* [Map of the city for phones with Java]. Available at: <http://www.mobimap.ru/item/2139302137/mobimap-moscow.jad> .
4. *Elektronnaia karta: Sputnikovaia navigatsiia* [Electronic Map: Satellite navigation]. Available at: <http://autosputnik.com/products/autosputnik/list/> .
5. *Elektronnye karty gorodov Rossii i planirovanie marshrutov* [Electronic maps of the cities of Russia and planning of routes]. Available at: <http://mosmap.ru/software/gis-mosmap-integrator.htm> .
6. Christofides N. *Graph theory. An algorithmic approach*. New York, Academic Press, 1975. (Russ. ed.: Kristofides N. *Teoriia grafov. Algoritmicheskii podkhod*. Moscow, Mir, 1978. 432 p.).
7. Stepanov V.P. *O matematicheskom modelirovanii dorozhnoi seti goroda dlia vybora marshruta proezda* [About mathematical modeling of the road network of the city to select the route of

travel]. *Tez. dokl. nauch. konf. MGTU im. N.E.Baumana* [Abstracts of reports of scientific conference of the Bauman MSTU]. Moscow, 2005, pp.110-111.

8. Stepanov V.P. O matematicheskom modelirovanii dorozhnoi seti [About mathematical modeling of the road network]. *Novye informatsionnye tekhnologii v avtomatizirovannykh sistemakh: Materialy 13 nauchno-prakticheskogo seminara* [New information technologies in automated systems: Proc. of the 13th scientific-practical seminar]. Moscow, MIEM Publ., 2010, pp. 237-243.

9. Moiseev N.N. *Chislennye metody v teorii optimal'nykh system* [Numerical methods in the theory of optimal systems]. Moscow, Nauka, 1971. 424 p.