Исследование ускорения метода определения надёжного кратчайшего пути в зависящей от времени стохастической транспортной сети

И.И. Абдулганиев[a], А.А. Агафонов[a]

[a] *Самарский национальный исследовательский университет имени академика С.П.Королёва, 443086, Московское шоссе, 34, Самара, Россия*

Аннотация

Целью работы является модификация и исследование алгоритма нахождения надежного кратчайшего пути в зависящей от времени стохастической транспортной сети. Под надёжностью понимается максимизация вероятности прибытия в пункт назначения в заданный заранее интервал времени. Модификация базового алгоритма нахождения надежного кратчайшего пути проводится с целью повышения скорости работы алгоритма и заключается в выборе определенного подмножества вершин и ребер графа, которые доступны для построения кратчайшего пути. Предложены два метода выбора подмножества вершин и ребер графа: на основе ограничивающего прямоугольника и на основе алгоритма k кратчайших путей. Проведены экспериментальные исследования эффективности базового и модифицированного алгоритмов на примере транспортной сети города Самары.

Ключевые слова: надёжный кратчайший путь; адаптивный маршрут; зависящая от времени сеть; алгоритм k кратчайших путей;

1. Введение

Дорожные заторы являются серьезной проблемой современного общества, которая имеет несколько существенных последствий. Для участников дорожного движения заторы снижают качество жизни, потребляя их свободное время. Для фирм заторы снижают производительность труда сотрудников и увеличивают грузовые транспортные расходы. Для общества в целом массовое скопление автомобилей приводят к нарушению работы экстренных служб, а так же негативно влияет на качество окружающей среды, вызывая большое количество выбросов выхлопных газов. Дорожные заторы ставят под угрозу безопасность движения, поднимая уровень стресса и усталости у водителей. Таким образом, становится всё более актуальным решение навигационных задач, связанных с выбором оптимального маршрута.

В последние десятилетия задаче нахождения кратчайшего пути в транспортных сетях (ТС) было посвящено большое количество работ. Однако большинство из них сосредотачивало своё внимание на нахождении наименьшего ожидаемого времени движения. В зависимости от типа веса дорожного сегмента различают несколько разновидностей моделей. В классических моделях [1,2,3] время прохождения дорожного сегмента рассматривается как константное или зависящее от времени. В тоже время реальная ситуация показывает, что время прохождения участка сети непрерывно меняется и зависит от многих факторов, таких как время суток, плохая погода, заторы на дорогах, строительные работы и т.д. Выше сказанное приводит к новому семейству моделей, в которых время прохождения является стохастической величиной. В таких моделях [4,5,6] время прохождения дорожного сегмента представляется случайной величиной с функцией распределения, зависящей от времени. В любом случае, условие оптимальности для маршрутизации может быть определено по-разному в зависимости от используемой целевой функции. Можно выделить следующие наиболее часто встречаемые виды целевых функций:

1) минимизация ожидаемого времени движения [4 – 7];
2) максимизация вероятности прибытия в пункт назначения в заданный заранее интервал времени [8 – 9];
3) максимизация вероятности времени нахождения в пути, меньше заданного порога [10,11];
4) минимизация наихудшего возможного времени в пути [12];
5) минимизация времени движения, необходимого для обеспечения прибытия в пункт назначения к выбранному моменту времени с указанной вероятностью [13].

Приведённые виды целевых функций можно разделить на две группы: задача наименьшего ожидаемого времени (least expected time – LET) (1) и задача надёжного кратчайшего пути (reliable shortest path – RSP) (2-5). LET проблема была хорошо изучена, существует много эффективных алгоритмов для различных вариантов задачи [6,7]. Тем не менее, в ряде практических задач путь с наименьшим ожидаемым временем может быть не подходящим, так как он не принимает во внимание дисперсию времени движения и не даёт никаких гарантий надёжности. Во многих случаях участники дорожного движения готовы пожертвовать временем в пути, чтобы выбрать более надёжный маршрут [8]. Задача нахождения надёжного кратчайшего пути была исследована в [10, 11]. В работе [14] был предложен алгоритм поиска кратчайшего маршрута, учитывающий текущую и прогнозную информацию о параметрах транспортных потоков в сети, являющий модификацией алгоритма из [8]. Однако описанные в работах алгоритмы являются вычислительно сложными и не могут быть использованы для определения кратчайшего пути в крупномасштабных сетях в режиме реального времени.

В данной работе исследуется метод определения надежного кратчайшего пути в зависящей от времени стохастической транспортной сети. Рассматривается критерий оптимальности, связанный с максимизацией вероятности прибытия в пункт назначения в течение заданного заранее интервала времени. Целью работы является ускорение существующего алгоритма поиска надежного кратчайшего пути. Ускорение работы алгоритма достигается за счет выбора некоторого подмножества вершин графа, используемых для нахождения кратчайшего маршрута движения. В качестве базового алгоритма используется алгоритм, предложенный группой авторов в работе [8].

Работа построена следующим образом. Во втором разделе вводятся основные понятия, постановка задачи и описание базового алгоритма ее решения. В третьем разделе представлены модификации алгоритма поиска надежного кратчайшего пути с целью повышения скорости работы алгоритма. В четвертом разделе приводятся экспериментальные исследования предложенных алгоритмов. В завершении работы представлены выводы, даётся список использованной литературы.

2. Основные обозначения, описание базового алгоритма

Транспортную сеть определим как ориентированный, зависящий от времени стохастический граф:

$G = (N, A, P)$,

где N – непустое множество вершин, представляющее собой разделяющие участки дорог узлы, $|N|$ – количество вершин, A – множество рёбер, которые соответствуют реальным участкам дорог (сегментам дорожной сети), $|A|$ – количество рёбер, P – вероятностное описание времени прохождения сегментов сети. Направление ребра определяет направления движения транспортного средства на соответствующем участке дороги. Будем считать, что граф имеет пространственную привязку, т.е. каждая вершина графа $i \in N$ имеет координаты $(x, y)_i$, которые определяются по физическому расположению соответствующего узла в реальной дорожной сети.

Введём следующие обозначения:

$T_{ij}(\tau)$ – время прохождения дорожного сегмента $(i, j) \in A$. Время представляется как случайная величина с функцией распределения $p_{ij}^{\tau}(t)$, зависящей от момента времени τ, в который данное транспортное средство попадает на данный участок;

$r \in N$ – начальная вершина или вершина отправления;

$s \in N$ – конечная вершина или вершина назначения;

T – максимальное количество времени, допустимое для достижения конечной вершины – временной бюджет.

Оптимальная стратегия навигации определяется как стратегия максимизации вероятности прибытия в вершину назначения s за время, меньшее чем T. В большинстве работ [8,14,15] данную проблему сокращенно обозначают как *SOTA* (*Stochastic On Time Arrival*).

В работах [8,19] для определения оптимальной стратегии навигации вводят дополнительное обозначение $u_i^{\tau}(t)$ – вероятность достижения конечной вершины s из вершины i в момент времени τ, при временном бюджете t (надежность пути).

Определение 1 [8]. Оптимальная стратегия навигации для проблемы SOTA формулируется следующим образом:

$$\begin{aligned} & u_i^{\tau}(t) = \max_j \int_0^t p_{ij}^{\tau}(\omega) u_j^{\tau+\omega}(t-\omega)\, d\omega \\ & \forall_i \in N,\ i \neq s,\ (i,j) \in A,\ t \in [0,T],\ \tau \geq 0; \\ & u_s^{\tau}(t) = 1\ t \in [0,T],\ \tau \geq 0. \end{aligned} \quad (1)$$

Дискретный алгоритм решения проблемы (1) был описан в [8]. Мы будем использовать его в качестве базового. В виде псевдокода алгоритм выглядит следующим образом:

Шаг 0. Инициализация.

$$\begin{aligned} & k = 0, \\ & u_i^k(x) = 0,\ \forall i \in N,\ i \neq s,\ x \in \mathrm{N},\ x \in [0, T/\Delta t], \\ & u_s^k(t) = 1,\ x \in \mathrm{N},\ x \in [0, T/\Delta t]. \end{aligned}$$

Шаг 1. Обновление.

ЦИКЛ по $k = 1, 2, ..., L = \lceil T/\Delta t \rceil$

$$\begin{aligned} & \tau^k = k\delta, \\ & u_s^k(t) = 1,\ x \in \mathrm{N},\ x \in [0, T/\Delta t], \\ & u_i^k(x) = u_i^{k-1}(x),\ \forall i \in N,\ i \neq s,\ (i,j) \in A,\ x \in \mathrm{N},\ x \in [0, (\tau^k - \delta)/\Delta t], \\ & u_i^k(x) = \max_j \sum_{h=0}^{x} p_{ij}(h) u_j^{k-1}(x-h),\ \ \forall i \in N,\ i \neq s,\ (i,j) \in A,\ x \in \mathrm{N},\ x \in [((\tau^k - \delta)/\Delta t + 1), (\tau^k/\Delta t)] \end{aligned}$$

КОНЕЦ ЦИКЛА

В алгоритме Δt – длина интервала дискретизации, δ – минимальное время прохождения сегмента транспортной сети.

Правило принятия решения о выборе следующей вершины j в графе транспортной сети при данном бюджете времени t и рассчитанном массиве вероятностей $u_i(x)$ выглядит следующим образом:

$$j = \arg\max_{i \in N} u_i(t) \tag{2}$$

В описанном алгоритме для построения кратчайшего маршрута итеративно просматриваются все вершины графа, что делает этот алгоритм вычислительно сложным и не применимым для нахождения кратчайшего маршрута в крупномасштабных сетях в режиме реального времени. В следующем разделе представлены модификации алгоритма, заключающиеся в выборе определенного подмножества вершин графа, которые будут использованы при нахождении кратчайшего маршрута.

3. Модифицированный алгоритм поиска кратчайшего маршрута

Основная идея модификации алгоритма состоит в том, чтобы добиться ускорения работы базового алгоритма путём уменьшения количества вершин, которые являются кандидатами в кратчайший путь. Для ускорения работы алгоритма предлагаются два метода выбора подмножества вершин и ребер графа, используемых для построения надежного кратчайшего маршрута движения: на основе ограничивающего прямоугольника и на основе алгоритма k кратчайших путей.

3.1. Выбор подмножества вершин на основе ограничивающего прямоугольника

Очевидным способом сокращения количества вершин, использующихся в поиске кратчайшего маршрута, является выбор только тех вершин, координаты которых попадают в ограничивающий прямоугольник между начальной и конечной вершинами.

Пусть $(x, y)_{r \in N}$ – координаты начальной вершины, $(x, y)_{s \in N}$ – координаты конечной вершины.

Координаты ограничивающего прямоугольника будут иметь вид:

$$\{x_{\min} = \min(x_r, x_s) - \Delta, y_{\min} = \min(y_r, y_s) - \Delta, x_{\max} = \max(x_r, x_s) + \Delta, y_{\max} = \max(y_r, y_s) + \Delta\},$$

где Δ - буферное расстояние, выбираемое экспериментально.

Тогда подмножества вершин и ребер графа могут быть определены следующим образом:

$$\overline{N} = \{i \in N : x_{\min} \le x_i \le x_{\max} \wedge y_{\min} \le y_i \le y_{\max}\},$$
$$\overline{A} = \{(i, j) \in A : i \in \overline{N} \wedge j \in \overline{N}\}.$$

Данный способ выбора подмножества вершин является вычислительно простым, однако получаемое подмножество вершин может быть избыточным или, наоборот, недостаточным, в зависимости от структуры сети и расположения начальной и конечной вершин.

3.2. Выбор подмножества вершин на основе алгоритма k кратчайших путей

Метод k кратчайших путей заключается в выборе нескольких путей между начальными и конечными вершинами, обычно наименьшей длины [16]. В зависимости от постановки задачи, кратчайшие пути могут содержать или не содержать общие вершины и ребра. Мы считаем, что вершины и ребра, которые входят в кратчайший путь, помечаются как пройденные и не могут быть использованы для построения следующих кратчайших путей. Для нахождения отдельных кратчайших путей могут быть использованы различные алгоритмы, в данной работе используется стандартный алгоритм Дейкстры[1].

Для формального описания метода выбора подмножества вершин на основе метода k кратчайших путей введём дополнительные обозначения:

Пусть $d_{rs}^k = \{v_1^k, v_2^k, ..., v_M^k\}$ – k-й кратчайший путь из вершины r в s,

где $v_m^k \in N, m = \overline{0, M^k - 1}$ – вершина графа, входящая в k-й кратчайший путь;

M^k – число вершин, входящих в кратчайший путь;

$k = \overline{1, K}$ – порядковый номер кратчайшего пути;

K – количество кратчайших путей, выбираемое экспериментально.

Тогда подмножества вершин и ребер графа могут быть определены следующим образом:

$$\overline{N} = \left\{ i \in \left\{ d_{rs}^{k} \right\}_{k=\overline{1,K}} \right\},$$
$$\overline{A} = \left\{ (i,\, j) \in A : i \in \overline{N} \wedge j \in \overline{N} \right\}.$$

Метод выбора подмножества вершин на основе алгоритма k кратчайших путей имеет б*о*льшую вычислительную сложность по сравнению с методом выбора вершин на основе ограничивающего прямоугольника, однако не зависит от структуры сети и позволяет настраивать размер подмножества выбором соответствующего параметра K .

3.3. Модифицированный алгоритм поиска надежного кратчайшего пути.

Модифицированный алгоритм поиска надежного кратчайшего пути, использующий подмножества вершин $\overline{N}$ и ребер $\overline{A}$ графа, в виде псевдокода может быть записан следующим образом:

Шаг 0. Инициализация.

$$k = 0,$$
$$u_i^k(x) = 0, \forall i \in N, i \neq s, x \in \overline{\mathrm{N}}, x \in [0, T / \Delta t],$$
$$u_s^k(t) = 1, x \in \overline{\mathrm{N}}, x \in [0, T / \Delta t].$$

Шаг 1. Обновление.

ЦИКЛ по $k = 1,2,..., L = \lceil T / \Delta t \rceil$

$$\tau^k = k\delta,$$
$$u_s^k(t) = 1, x \in \overline{\mathrm{N}}, x \in [0, T / \Delta t],$$
$$u_i^k(x) = u_i^{k-1}(x), \ \forall i \in N, i \neq s, (i, j) \in \overline{A}, x \in \overline{\mathrm{N}}, x \in [0, (\tau^k - \delta) / \Delta t],$$
$$u_i^k(x) = \max_j \sum_{h=0}^{x} p_{ij}(h) u_j^{k-1}(x-h), \ \forall i \in N, i \neq s, (i, j) \in \overline{A}, x \in \overline{\mathrm{N}}, x \in [((\tau^k - \delta) / \Delta t + 1), (\tau^k / \Delta t)]$$

КОНЕЦ ЦИКЛА

Подмножества $\overline{N}$ и $\overline{A}$ выбираются одним из вышеописанных способов.

4. Экспериментальные исследования

Цели проводимых экспериментальных исследований являлось сравнение базового и модифицированного алгоритмов по критериям скорости работы и надежности найденного маршрута движения $u_i^{\tau}(t)$.

Исследования алгоритмов проводились на транспортной сети г. Самары. Транспортная сеть состоит из 9721 вершин и 26088 сегментов. В качестве веса дорожного сегмента использовались данные о времени прохождения сегмента, усреднённые за десятиминутный интервал. Для распределения времени движения на конкретном сегменте (i, j) было выбрано логнормальное распределение, как наиболее целесообразное, по мнению авторов. Ниже приведены результаты экспериментов.

Для тестирования алгоритмов были выбраны 10 начальных и 10 конечных вершин транспортной сети, расположенных на значительном расстоянии друг от друга. Для каждой пары вершин решалась задача нахождения надёжного кратчайшего пути базовым и модифицированными алгоритмами. Для исследования алгоритма на основе ограничивающего прямоугольника были выбраны следующие значения буферного расстояния: $\Delta = \{200;\ 300;\ 400;\ 500;\ 600\}$. Исследование алгоритма на основе k-кратчайших путей проводилось в зависимости от количества кратчайших путей $K = \{3;4;5;6;7\}$.

В первую очередь сравнение алгоритмов было проведено по критерию надежности (вероятности достижения конечной вершины при заданном временном бюджете). Результаты сравнения представлены в таблице 1.

Таблица 1. Результаты сравнения алгоритмов по надёжности

Алгоритм	Базовый алгоритм	Ограничивающий прямоугольник					*k*-кратчайших путей				
		200	300	400	500	600	3	4	5	6	7
Надежность	0.99836	0.99568	0.99567	0.99624	0.996328	0.99642	0.99494	0.99512	0.99512	0.99523	0.995239

Все алгоритмы показали схожие результаты по выбранному критерию для заданных временных бюджетов поездки, модифицированные алгоритмы незначительно уступают базовому.

Следующим этапом экспериментального анализа являлось сравнение алгоритмов по критерию скорости работы. Время работы базового алгоритма составило в среднем 10 минут в зависимости от используемого временного бюджета поездки T . На рисунке 1 представлена гистограмма ускорения времени работы алгоритмов по сравнению с базовым алгоритмом для различных значений буферного расстояния и количества кратчайших путей.

Наилучшие результаты по критерию скорости показал алгоритм на основе k-кратчайших путей. Скорость работы по сравнению с базовым алгоритмом увеличилась в 10-35 раз в зависимости от количества кратчайших путей K. Модификация базового алгоритма на основе ограничивающего прямоугольника позволяет увеличить скорость работы в 8-10 раз. Заметим, что величина буферного расстояния оказывает не сильное влияние на скорость работы алгоритма, однако слишком большое значение может значительно увеличить время работы.

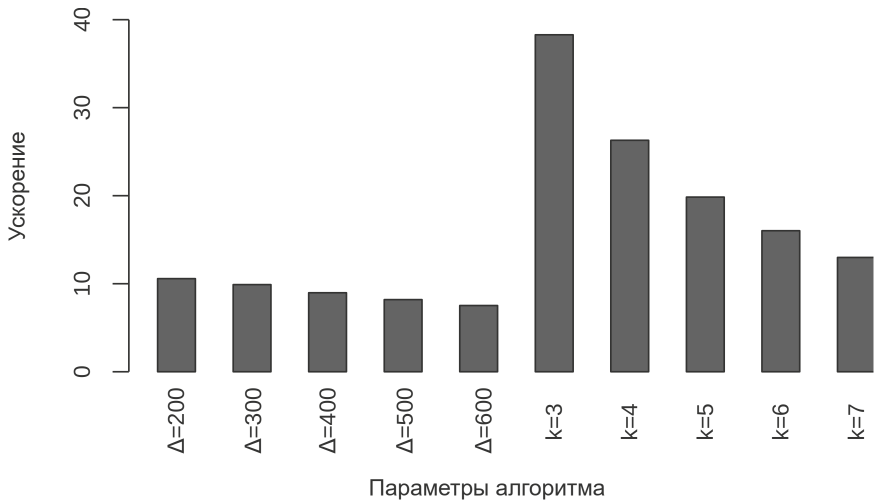

Рис. 1. Ускорение времени работы модифицированного алгоритма.

По результатам исследований можно сказать, что предложенные варианты модификации позволяют существенно уменьшить время выполнения работы базового алгоритма (в 8-35 раз), при этом надёжность построенного пути снижается незначительно. Это позволяет использовать данные модификации для нахождения кратчайшего маршрута в крупномасштабных сетях в режиме реального времени.

5. Заключение

В работе рассмотрен алгоритм нахождения надёжного кратчайшего пути в зависящей от времени стохастической транспортной сети. Под надёжностью понимается максимизация вероятности прибытия в пункт назначения в заданный заранее интервал времени. Предложены две его модификации с целью повышения скорости работы – на основе ограничивающего прямоугольника и на основе алгоритма k-кратчайших путей. Проведено сравнение разработанных модификаций с базовым алгоритмом на транспортной сети г. Самары.

Результаты экспериментальных исследований показали, что введённые модификации дают значительное ускорение времени работы (в 10-35 раз) при практически идентичных показателях надёжности.

Стоит отметить, что оба предложенных алгоритма обладают свойством, не присуще базовому алгоритму, а именно: позволяют находить кратчайший маршрут в крупномасштабных сетях в режиме реального времени.

Дальнейшие направления работ включают в себя:

- исследования, связанные с выбором маршрута движения в зависимости от значений параметров транспортных потоков (текущей дорожной ситуации);
- исследования, связанные с выбором временного бюджета поездки.

Благодарности

Работа выполнена при частичной финансовой поддержке гранта РФФИ 16-37-00055- мол_а.

Литература

[1] Dijkstra, E.W. A Note on Two Problems on Connexion with Graphs / E.W. Dijkstra // Numerische Mathematik. – 1959. – Vol. 1. – P. 269-271.

[2] Bellman, R.E. On a routing problem / R.E. Bellman // Quarterly of Applied Mathematics. – 1958. – Vol. 16. – P. 87-90.

[3] Dreyfus, S.E. An appraisal of some shortest-path algorithm / S.E. Dreyfus // Operations Research. – 1969. – Vol. 17. – P. 395-412.

[4] Gao, S. Optimal routing policy problems in stochastic timedependent networks / S. Gao, I. Chabini // Transportation Research Part B. – 2006. – Vol. 40. – P. 93-122.

[5] Gao, S. Real-time traveler information for optimal adaptive routing in stochastic time-dependent networks / S. Gao, H. Huang // Transportation Research Part C. – 2012. – Vol. 21. – P. 196-213.

[6] Hall, R.W. The fastest path through a network with random time-dependent travel times / R.W. Hall // Transportation Science. – 1986. – Vol. 20(3). – P. 182-188.

[7] Fu, L. Expected shortest paths in dynamic and stochastic traffic networks / L. Fu, L.R. Rilett // Transportation Research Part B. – 1998. – Vol. 32(7). – P. 499-516.

[8] Samaranayake, S. A tractable class of algorithms for reliable routing in stochastic networks / S. Samaranayake, S. Blandin, A. Bayen // Transportation Research Part C. – 2012. – Vol. 20. – P. 199-217.

[9] Fan, Y. Optimal routing for maximizing the travel time reliability / Y. Fan, Y. Nie // Networks and Spatial Economics. – 2006. – Vol. 6(3-4). – P. 333-344.
[10] Frank H. Shortest paths in probabilistic graphs / H. Frank // Operations Research. – 1969. – Vol. 17(4). – P. 583-589.
[11] Mirchandani P.B. Shortest distance and reliability of probabilistic networks / P.B. Mirchandani // Computers and Operations Research. – 1976. – Vol. 3(4). – P. 347-355.
[12] Montemanni R. An exact algorithm for the robust shortest path problem with interval data/ R. Montemanni, L. Gambardella // Computers and Operations Research. – 2004. – Vol. 31(10). – P. 1667-1680.
[13] Nie, Y. Shortest path problem considering on-time arrival probability / Y. Nie, X. Wu // Transportation Research Part B. – 2009. – Vol. 43(6). – P. 597-613.
[14] Агафонов, А.А. Метод определения надёжного кратчайшего пути в зависящей от времени стохастической сети и его применение в геоинформационных задачах управления транспортом / А.А. Агафонов, В.В. Мясников // Компьютерная оптика. – 2016. – Т. 40, № 2. – С. 275-283. – DOI: 10.18287/2412-6179-2016-40-2-275-283.
[15] Nie, Y. Arriving-on-time problem: Discrete algorithm that ensures convergence / Y. Nie, Y. Fan // Transportation Research Record. – 2006. – Vol. 1964. – P. 193-200.
[16] Yen J. Y. Finding the K shortest loopless paths in a network / J.Y. Yen // Management Science. – 1971. – Vol. 17(11). – P. 712-716.