ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М. В. ЛОМОНОСОВА»

МЕХАНИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
специалиста

ПОСТРОЕНИЕ ОПТИМАЛЬНОГО МАРШРУТА ПРИ ЗАДАННОЙ МОДЕЛИ ДВИЖЕНИЯ ДРУГИХ УЧАСТНИКОВ ТРАНСПОРТНОЙ СЕТИ

Выполнил студент 610 группы
Разумова Любовь Евгеньевна

____
подпись студента

Научный руководитель:
доктор физико-математических наук
Афонин Сергей Александрович

____
подпись научного руководителя

Москва
2022

Содержание

Введение

В данной работе рассматривается задача нахождения наилучшего в каком-то смысле пути с учетом движения фиксированного количества участников по заданным ранее маршрутам в условиях ограниченности модели дорожной системы. Новый построенный маршрут должен отвечать выбранным критериям кратчайшести среди всевозможных путей на всем временном промежутке, но не обязательно в каждый момент времени. Знание маршрутов изначальных участников помогает определить плотность автомобильного потока на конкретных отрезках пути. Рассматриваемая модель приближена к реальной дорожной системе городов, поэтому на всех ее участках наложены ограничения по вместимости участников и скорости их движения. Такие ограничения влияют на показатели маршрутов участников, такие как итоговое время движения и длину пути.

Тема актуальна в наше время, так как она помогает решить проблему пробок на дорогах, а также призвана упростить водителям выбор маршрута, который займет у них наименьшее время. Задача имеет практический характер... Проблема пробок в Москве стоит очень остро, ученые решают ее не первый год..

В мире прогнозы загруженности используются для автоматического управления дорожным движением в некоторых городах. Первые прототипы, в которых были применены прогнозы, появились появились в 1998 году в США. А первое пилотное использование системы, «заглядывающей в будущее», началось в 2006 году в Сингапуре. Среди наших соотечественников похожей задачей занимается отдел навигации Яндекса. Разработчики собирают информацию по трекам движения автомобилей, осуществляют привязку треков к ребрам графа дорожной сети, вычисляют некоторую усредненную скорость на отдельных участках, рисуют карту прогноза дорожной ситуации на ближайший час и в связи с этим предлагают оптимальный по времени путь, а также еще пару алтернативных маршрутов. Сложность сбора информации и построения треков движения состоит в том, что, во-первых, не все пользуются сервисами Яндекса при построении своего маршрута, и во-вторых, в данных постоянно возникают лишние шумы, что приводит к выбросам на графиках, и с таким качеством информации работать очень сложно. Мы же рассматриваем более прозрачную и простую модель, когда все маршруты изначально проложены и нам известны, и они не меняются с течением времени. Это не умаляет значимости и важности поставленной нами задачи. Мы получим более четкие результаты, идеи в дальнейшем могут развиваться и открывать новые возможности для более широкой задачи, например, как у Яндекса. Также стоит отметить, что специалисты по навигации используют статистические методы и машинное обучение в качестве инструмента для решения своих задач, мы же подойдем к вопросу с другой стороны и применим другие алгоритмы. На данный момент отдел навигации Яндекса проводит улчшения своих методов и подходов к решению задач, а также придумывает какие-то новые метрики и способы оценки качества этих решений.

Задачи, которые мы ставили перед собой в рамках выбранной темы дипломной работы: формальная постановка задачи, анализ полученных ранее результатов в этой теме, разработка простого алгоритма решения на основе моделирования и его оценка, оценка устойчивости полученного решения, попытка обощения дорожной сети, ее расширение (или сужение?). Методы исследования включают в себя: построение графа с вершинами в концевых точках заданных маршрутов и ребрами, отображающими дороги между ними, определение функции веса-загруженности дорог,

Основа нашей задачи - нахождение наилучшего пути в условиях изменчивости плотности и скорости дорожного потока на участках в заисимости от времени.

В первой главе вы сможете ознакомиться с деталями поставленной задачи, далее мы решим ее путем моделирования дорожной ситуации и применением некоторых известных алгоритмов, в третьей главе поговорим о достоинствах и недостатках такого решения, его сложности и реализуемости в реальной жизни. Четвертая глава будет содержать описание некоторых модификаций графа дорожной сети, а также улучшений решения на таком графе. В завершении поделимся результатами проделанной работы, оценим их качество и сделаем выводы.

1 Постановка задачи

Пусть задан ориентированный *граф дорожной сети* $G(V, E, l)$ таким образом, что вершины $v \in V$ осуществляют роль перекрестков, а ребра $e \in E$ - роль дорог. Каждое ребро имеет длину, т.е. задана функция $l : E \to \mathbb{R}$. Также мы говорим, что задана некоторая модель движения. Обратимся к теории автоматов, чтобы попробовать формализовать подразумеваемые под этим понятием правила движения АТС. Аналогичный подход использовали К. Нагель и М. Шрекенберг [1]. В своей работе авторы рассматривают модель клеточных автоматов, которая предполагает разбиение дорог на клетки и использование дискретного времени. Эта идея нашла применение в описании движения физических частиц [2], а также в исследовании пробок на дорогах [3]. Мы же не будем ограничиваться клеточными автоматами и опишем случай непрерывного движения.

Моделью движения АТС назовем $M = (n, G, S, F, \{t_i\}_{i=1}^n, \{\varphi_i\}_{i=1}^n)$, где n - количество участников движения, G — граф дорожной сети, S — множество состояний, которые могут принимать участники, $F \subset S$ — множество заключительных состояний, $t_i : S^n \to R_{>=0}$ — функция критического момента движения участника i, $\varphi_i : S^n \times \mathbb{R}_{>=0} \to S$ — функция перехода состояния i-ого участника в некоторый момент времени t. Считаем, что, попав в заключительное состояние, мы не можем его покинуть:

$$\varphi_i(s_1, \ldots, s_i, \ldots, s_n, t) = s_i,\ s_1, \ldots, s_n \in S,\ s_i \in F,\ \forall t \in \mathbb{R}_{\geq 0}.$$

Множество S описывает текущий характер двжиения АТС. Это, в первую очередь, ребро, на котором едет участник, координата на этом ребре, скорость участника и его ускорение, если оно есть. Подразумевается, что состояния можно разбить на классы, например «свободное движение», «ожидание», «прибытие», «торможение», «ускорение» и тд. Функция t_i описывает время, когда участнику необходимо совершить переход из текущего состояния в состояние другого класса.

Модель движения АТС можно описать некоторой диаграммой, описывающей переходы между классами для каждого участника. Диаграмма представляет собой ориентированный граф, где вершины — классы состояний, а ребра являются переходами в другое состояние. Метка на ребре — условие перехода, который осуществляется, если $t = t_i(s_1, \ldots, s_n)$ для некоторого s_i из класса состояний. Непомеченные ребра соответствуют условию $t < t_i(s_1, \ldots, s_n)$:

$$\varphi_i(s_1, \ldots, s_n, t) = s_i,\ s_1, \ldots, s_n \in S,\ 0 \leq t < t_i(s_1, \ldots, s_n).$$

Например, правила движения, в которых поведение участника зависит от расстояния, до впереди идущего участника может быть задано следующей диаграммой (см. рис. 1):

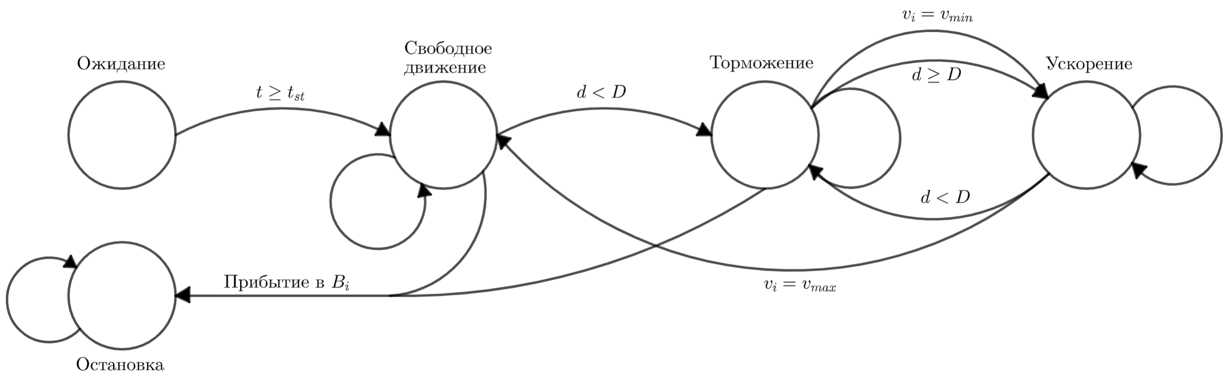


Рис. 1: Диаграмма для i-ого участника в модели движения, где d – расстояние до впереди идущего участника, D – максимальное расстояние взаимодействия с впереди идущим участником, v_{max} –максимально возможная скорость, v_{min} –минимально возможная скорость, t_{st} – время старта.

Перейдем от формального определения модели движения непосредственно к постановке задачи. Пусть имеется n участников, которые движутся по заранее заданным маршрутам: $p_i = \langle E^i_{j_1}, E^i_{j_2}, \ldots, E^i_{j_{m_i}} \rangle$, $E^i_{j_k} \in E \quad i = 1, \ldots, n$. Добавим к ним $(n+1)$-ого участника, которому нужно добраться из пункта A в пункт B, $A, B \in V$. Будем считать, что движение добавленного участника не влияет на движение n участников. Определим $P(A, B)$ – множество всех простых путей из A в B. Модель движения позволяет определить *часть пройденного пути* для каждого АТС в зависимости от движения других участников, т.е. определить непрерывные монотонные функции

$$x_i(G, p_1, \ldots, p_n, p, \cdot) : \mathbb{R} \to [0, 1],\ i = 1, \ldots, n+1,\ p \in P(A, B).$$

Ввиду однозначной определенности последовательности ребер p_j и графа дорожной сети G, явную зависимость x_i от данных параметров можно не указывать.

На множестве путей $P(A, B)$ определим $T(p) = \inf_t \{t : x_{n+1}(p, t) = 1\}$ – время прибытия $(n+1)$-ого участника в вершину B при движении по маршруту p. Требуется найти такой путь $p^* \in P(A, B)$, что $T(p^*)$ - минимальна. Другими словами, для заданной модели движения на графе дорожной сети $G(V, E, l)$ при движении n участников по путям $p_1, \ldots, p_n$ требуется найти такой путь p^* из A в B, что движение нового участника по этому пути p^* будет *оптимально*, то есть

$$p^* = \underset{p \in P(A,B)}{\operatorname{argmin}}\, T(p).$$

2 Модели движения

В первую очередь для решения задачи, нужно конкретизировать модели движения. Рассмотрим те из них, в которых изменения скорости участников базируются только на количестве участников на ребре в момент времени t. Назовем такие модели *макроскопическими*, а все остальные — *микроскопическими*.

2.1 Макроскопические модели

При движении n участников макроскопические модели можно параметризовать набором действительных чисел $v_1, \ldots, v_n$, где v_k – скорость при k участниках на ребре. Множество состояний в макроскопических моделях движения можно разбить на n классов по количеству участников на ребре (см. рис. 2). Общее количество всех возможных переходов между классами состояний для i –ого участника составит $n(n-1)+2n$.

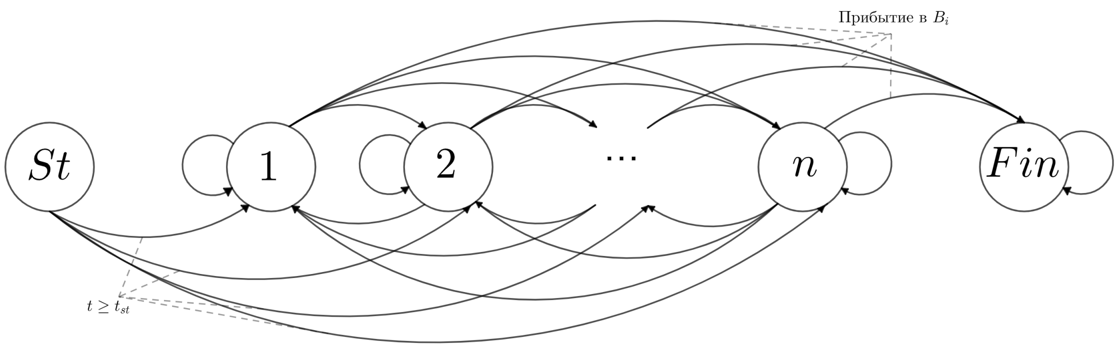


Рис. 2: Диаграмма для i-ого участника в макроскопических моделях движения

Пусть n_i – количество участников на ребре e_i, на котором находится участник i. Переходы из состояния k в состояние m осуществляются при выполнении сразу двух условий: $n_i \neq k$ и $n_i = m$. При этом критический момент для i –ого участника наступает, когда какой-либо участник въезжает на ребро e_i или съезжает с него. Так, если χ_i – часть пройденного i-ым участником ребра e_i, то

$$t_i = min\left\{\frac{l(e_j)\,(1-\chi_j)}{v_{n_j}}\;\middle|\; j=1,\ldots,n;\; e_j = e_i \text{ или следующее за } e_j \text{ ребро в } p_j \text{ это } e_i\right\}.$$

Для того, чтобы критические моменты $t_i, i = 1, \ldots, n$ были определены, макроскопические модели требуют положительности скоростей $v_i, i = 1, \ldots, n$. Условие $v_i \geq v_{min} > 0, i = 1, \ldots, n$ обеспечивает достижимость переходов в другие классы состояний:

$$t_i < T_i = \frac{max\{l(e_i) \mid e_i \in p_i\}}{v_{min}}.$$

Количество таких переходов для участника i будет ограничено количеством ребер в пути p_i, из чего можно сделать вывод, что все заключительные состояния достижимы.

Примером такой модели может послужить набор $v_k = \frac{v_{max}}{k}$ для k участников на ребре, $k \in \{1, \ldots, n\}$, где v_{max} – заданная максимальная скорость.

2.2 Микроскопические модели

Микроскопические модели, в отличие от макроскопических, предполагают другой набор другие зависимости скоростей. Опишем модель *следования за лидером* – модель, в которой поведение движения участника зависит от расстояния до впереди идущего АТС.

Пусть заданы максимальная $v_{max} > 0$ и минимальная $v_{min} > 0$ скорости участников, и ускорение $a = const, a > 0$, которое позволяет разгоняться от v_{min} до v_{max}. Будем считать, что участники подчиняются следующему правилу: оказавшись на расстоянии $d \leq l, l > 0$, которое мы будем называть *дистанцией торможения*, до лидера, участник мгновенно сбрасывает свою скорость до минимальной и продолжает движение с постоянной скоростью, пока не отдалится от лидера на *безопасное расстояние* $D > l$. Оказавшись на расстоянии D до лидера, участник с минимальной скоростью может перейти к равноускоренному движению. Диаграмма такой модели движения представлена на рис. 3.

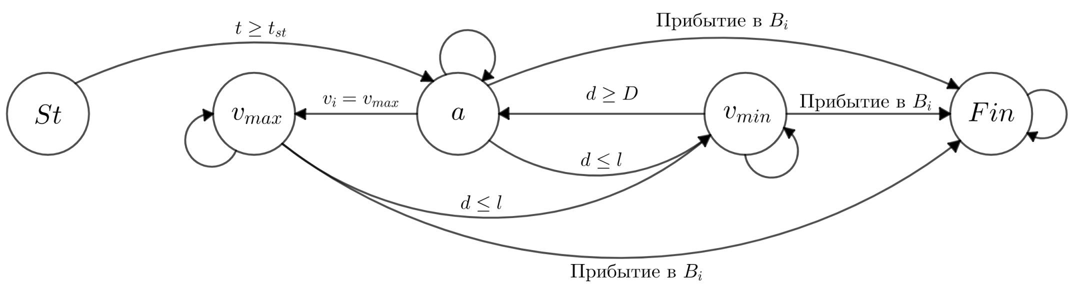


Рис. 3: Диаграмма для i-ого участника в модели следования за лидером

3 Моделирование

Под моделированием будем понимать воспроизведение движения $n+1$ участника, движущихся по путям $p_1, \ldots, p_n, p,\ p \in P(A, B)$ в модели движения $M = M(p_1, \ldots, p_n, p)$. Рассмотрим алгоритм нахождения времени, затраченного $n+1$-ым участником на путь p.

Algorithm 1 Моделирование движения участников

Input: количество участников $n+1$, граф дорожной сети G, модель движения $M = M(p_1, \ldots, p_n, p)$, набор начальных состояний $s_1^\circ, \ldots, s_{n+1}^\circ$

Output: $T(p)$

Data: текущее время t, критический момент движения i-ого участника t_i^*, критический момент t^*

$t = 0$
for $i = 1, \ldots, n+1$ **do**
    $s_i \leftarrow s_i^\circ$
end for
while $s_{n+1} \notin F$ **do**
    for $i = 1, \ldots, n+1$ **do**
        $t_i^* \leftarrow t_i(s_1, \ldots, s_{n+1})$
    end for
    $t^* \leftarrow min\left(t_1^*, \ldots, t_{n+1}^*\right)$
    for $i = 1, \ldots, n+1$ **do**
        $s_i \leftarrow \varphi_i(s_1, \ldots, s_{n+1}, t^*)$
    end for
    $t \leftarrow t^*$
end while
$T(p) \leftarrow t$

4 Поиск оптимального пути

Отметим, что количество путей конечно, поэтому первое, что приходит на ум в качестве решения, это перебор всех возможных путей и нахождение подходящего по затраченному времени. Посчитаем сложность этого алгоритма и сделаем вывод о его использовании в нашей задаче на практике.

4.1 Перебор. Сложность

Чтобы узнать, применим ли перебор в нашем случае, посчитаем сложность нахождения кратчайшего пути среди множества всех простых путей из A в B. Пусть $S_x(p)$ – сложность моделирования, т.е. нахождения функции $x^p_{n+1}(t)$, при выборе пути p. Тогда сложность перебора

$$S = \sum_{p \in P(A,B)} S_x(p) = |P(A,B)| * \overline{S}_x, \text{ где } \overline{S}_x \text{ – средняя сложность.}$$

Заметим, что S растет при увеличении количества возможных путей. Так, в полном графе на $|V|$ вершинах получим $S = 2^{|V|-2} * \overline{S}_x$.

В качестве примера можем рассмотерть также регулярный граф-решетку на $\mathbb{R}^2$ и на нем оценить снизу сложность поиска пути с минимальными тратами. Пусть точки A и B имеют координаты (a_1, a_2) и (b_1, b_2) соответственно. Тогда количество путей минимальной длины в метрике Манхэтенна будет составлять $C^{|a_1-b_1|}_{|a_1-b_1|+|a_2-b_2|}$. Понятно, что путей $|P(A,B)|$ в таком графе гораздо больше. Таким образом, получаем оценку снизу для регулярного решеточного графа

$$C^{|a_1-b_1|}_{|a_1-b_1|+|a_2-b_2|} * \overline{S}_x \leq |P(A,B)| * \overline{S}_x = S.$$

Например, в решетке-квадрате со стороной m при движении из угловой точки по диагонали в угловую точку напротив количество путей минимальной длины составит $C^m_{2m} = \frac{(2m)!}{m!m!}$. По формуле Стирлинга

$$C^m_{2m} = \frac{\sqrt{2\pi(2m)}\left(\frac{2m}{\exp}\right)^{2m}}{\left(\sqrt{2\pi m}\left(\frac{m}{\exp}\right)^m\right)\left(\sqrt{2\pi m}\left(\frac{m}{\exp}\right)^m\right)} = \frac{2\sqrt{\pi m}\left(\frac{2m}{\exp}\right)^{2m}}{2\pi m\left(\frac{m}{\exp}\right)^{2m}} = \frac{2^{2m}}{\sqrt{\pi m}}.$$

Понятно, что при увеличении m, количество путей экспоненциально растет.

С помощью перебора можно находить кратчайшие пути быстро, если $|P(A,B)|$ не велико. Однако изначально наша задача была сформулирована в терминах дорожной сети и предполагала графы с достаточно большим количеством вершин и ребер, что влияет на количество маршрутов для заданных точек. Таким образом, можно сделать вывод, что в общем случае перебор путей в нашей задаче на практике не применим.

4.2 Дейкстра

Рассмотрим вспомогательную задачу. Пусть на каждом ребре $e \in E$ графа $G(V, E)$ определена функция *временных затрат* $\phi_e(t) : \mathbb{R}_+ \to \mathbb{R}_+$. Если мы оказались в начальной вершине ребра e в момент времени t, то время преодоления ребра будет равняться $\phi_e(t)$. Рассмотрим путь $p = \langle V_0, e_1, V_1, e_2, V_2, \dots, V_{k-1}, e_k, V_k \rangle$ и начало движения происходит в вершине V_0 в момент времени $t_0 = t$, тогда

$$
\begin{aligned}
t_0 &= t \\
t_1 &= \phi_{e_1}(t) + t = \phi_{e_1}(t_0) + t_0 \\
t_2 &= \phi_{e_2}(\phi_{e_1}(t) + t) + \phi_{e_1}(t) + t = \phi_{e_2}(t_1) + t_1 \\
&\dots \\
t_i &= \phi_{e_i}(t_{i-1}) + t_{i-1} \\
&\dots \\
t_k &= \phi_{e_k}(t_{k-1}) + t_{k-1}
\end{aligned}
$$

Пусть $P(A, B)$ – множество всех простых путей из A в B в графе $G(V, E)$. Необходимо найти путь из A в B, который требует минимальных затрат, т.е.

$$T = \min_{p \in P(A,B)} t_{|p|}.$$

В общем случае функции временных затрат могут быть любыми. Давайте рассмотрим эту задачу с дополнительным условием на $\phi_e(t)$:

$$\phi_e(t) \leq \Delta + \phi_e(t + \Delta), \quad \Delta \geq 0$$

Назовем это условие *неравенством прохождения ребер.* Утверждается, что если для $\forall e \in E$ функции временных затрат ϕ_e удовлетворяют неравенству прохождения ребер, то задачу можно решить модифицированным алгоритмом Дейкстры.

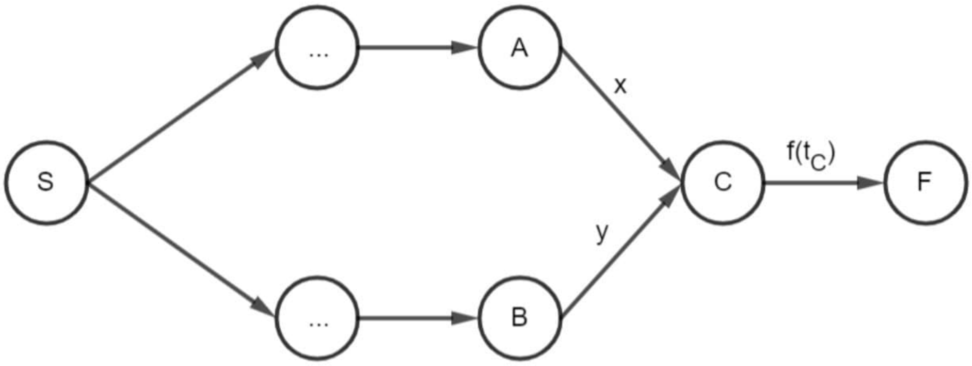
4.2.1 Модифицированный алгоритм Дейкстры

Для каждой вершины будем хранить два значения: минимальное время, за которое можно добраться до этой вершины, и ребро, через которое проходит кратчайший маршрут до вершины. Применяем стандартный алгоритм Дейкстры, с отличием, что при посещении вершины мы фиксируем время для нее и пересчитываем функции временных затрат на всех ребрах, исходящих из этой вершины.
Для запуска алгоритма потребуется задать начальное время - минимальное время в точке старта. Это можно использовать в анализе маршрута.

Утверждение 4.1. *Данный маршрут обладает наименьшим временем прохождения.*

Доказательство. Будем доказывать по индукции :
База индукции - в графе 2 вершины и несколько ребер между ними. Минимальным маршрутом будет то ребро, у которого наименьшее время прохождения.

Шаг индукции - считаем что в случае с m ($< n$) вершинами лемма справедлива. Рассмотрим граф, содержащий n вершин. Пусть $\exists$ маршрут P в этом графе, требующий меньше затрат, чем построенный нашим алгоритмом, тогда возьмем ближайшую к началу точку, обозначим ее C, в которой выбрано ребро, отличное от минимального по затратам. Очевидно, что если точка C совпадает с точкой F, концом маршрута, то P не является минимальным по времени прохождения.
Пусть ребро маршрута P в точку C выходит из точки B, а минимальное - из точки A. Построим маршрут по нашему алгоритму из S – начала маршрута в C. Заметим, что он проходит через точку A. Обозначим время этого маршрута за $t_a = T(S - ... - A - C)$, а время для части маршрута P из S в C, проходящего через точку B, за $t_b = T(S - ... - B - C)$. В подграфе $(P - C)$ вершин меньше чем n, а значит по индукции $t_a < t_b$.



Без ограничения общности, будем рассматривать часть маршрута P от точки C до F как одно ребро : C-F. Тогда время прохождения этого ребра $\phi(t_C) = \phi_{CF}(t_C)$, где t_C - время старта из точки C. Вспомним неравенство прохождения для ребер (см. выше) : $\phi(t_a) \leq (t_b - t_a) + \phi(t_b)$, где $\Delta = (t_b - t_a)$

Рассмотрим два маршрута $P : S - ... - B - C - F$ и $P' : S - ... - A - C - F$. Посчитаем время : $T(P) = t_b + \phi(t_b)$ и $T(P') = t_a + \phi(t_a)$ Используя неравенство, получаем : $T(P') = t_a + \phi(t_a) \leq t_a + (t_b - t_a) + \phi(t_b) = T(P)$ Значит маршрут P не является минимальным.

$\square$

Понятно, что если неравенство прохождения ребер не выполняется, то модифицированный алгоритм Дейкстры может построить не кратчайший маршрут в терминах временных затрат. Рассмотрим такой пример (рис. 1):

$$
\begin{aligned}
&p_1: \\
&\quad t_0 = 0 \\
&\quad \phi_{AC}(t) = 1 \\
&\quad \phi_{CB}(t) = 1 + 2 * \mathbb{I}\{t < 1.5\} \\
&p_2: \\
&\quad t_0 = 0 \\
&\quad \phi_{AC}(t) = 2 \\
&\quad \phi_{CB}(t) = 1 + 2 * \mathbb{I}\{t < 1.5\}
\end{aligned}
$$

Время прохождения пути p_1 будет составлять $t_{p_1} = t_0 + \phi_{AC}(t_0) + \phi_{CB}(\phi_{AC}(t_0)) = 1 + 3 = 4$. Время прохождения пути p_2 будет составлять $t_{p_2} = 2 + 1 = 3$. Очевидно, на путь p_2 потребуется меньше времени, чем на путь p_1, но алгоритм Дейкстры предложит в качестве решения задачи маршрут p_1.

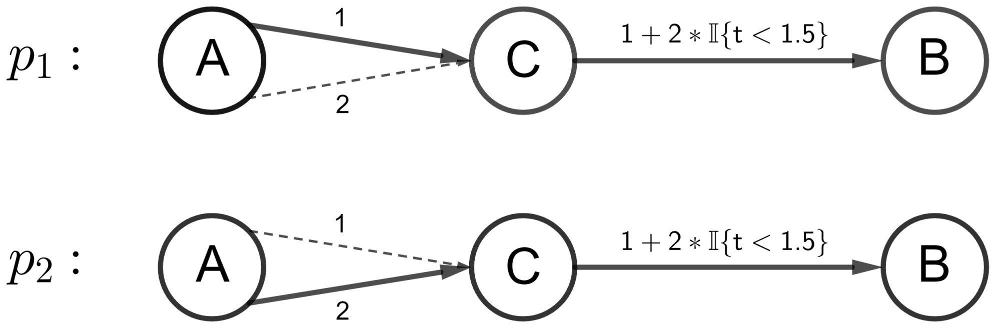


Рис. 4: Пример графа с невыполненным условием неравенства прохождения ребер

Отметим, что наша задача поиска оптимального маршрута сводится к вспомогательной задаче. Правила движения участников и их взаимодействий определяют функции ϕ_e, $e \in E$. Мы можем их получить путем численного моделирования. Неравенство прохождения ребер можно переформулировать так: дорожная сеть обладает условием FIFO — первый въехавший на дорогу первым ее покидает. Другими словами, если участники не обгоняют друг друга, то путь с минимальными затратами можно найти при помощи алгоритма Дейкстры.

Сложность модифицированного алгоритма Дейкстры

Устойчивость нашего решения

Практические результаты

Сложность решения

Альтернативные подходы

Заключение

Список литературы

[1] *Nagel K., Schreckenberg M.* A cellular automation model for freeway traffic // Phys. I France. 1992. V. 2. P. 2221–2229.

[2] *Chowdhury D., Santen L., Schadschneider A.* Statistical physics of vehicular traffic and some related systems // Phys. Rep. 2000. V. 329. P. 199–329.

[3] *Nagatani T.* The physics of traffic jams // Reports on Progress in Physics. 2002. V. 65. P. 1331–1386.