

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

1	Постановка задачи	5
2	Модели движения	7
2.1	Макроскопические модели	7
2.2	Микроскопические модели	8
3	Моделирование	9
4	Поиск оптимального пути	10
4.1	Перебор. Сложность	10
4.2	Альтернативный подход к решению	11
4.2.1	Модифицированный алгоритм Дейкстры	11
4.2.2	Применимость модифицированного алгоритма Дейкстры	13
	Литература	17

Введение

В данной работе рассматривается задача нахождения наилучшего в смысле временных затрат пути с учетом движения фиксированного количества участников по заданным маршрутам в рамках некоторой модели движения. Под моделью движения понимается некий набор правил, которые задают скоростной режим участника движения в зависимости от его взаимодействия с другими автомобильными транспортными средствами (АТС). Маршруты участников могут пересекаться, что приводит к изменению скорости участников и образованию заторов. Целью работы является проложение оптимального маршрута в условиях возможности возникновения пробок на вариации подходящих путей.

Задачи на транспортную тематику никогда не теряют своей актуальности, в том числе и задача маршрутизации. Сложно найти человека, который в целях сбережения личного времени не пользуется какой-нибудь системой навигации, не говоря уже о сервисах такси, для которых крайне важно минимизировать временные затраты в пути. Однако данная задача решается построением оптимального маршрута с учетом картины заторов на момент составления этого маршрута. Образование заторов — часто непредсказуемое явление, и пробка может появиться прямо перед нами на участке пути, который при построении маршрута был свободен. Несмотря на то, что время и место заторов сложно предугадать, задачей их прогнозирования занимаются уже долгое время.

Помимо систем навигации прогнозы загруженности используются для автоматического управления дорожным движением в некоторых городах. Первые прототипы, в которых были применены прогнозы, появились в 1998 году в США. А первое пилотное использование системы, «заглядывающей в будущее», началось в 2006 году в Сингапуре в целях заблаговременного предупреждения водителей о дорожных условиях впереди и ценах, взимаемых за пробки в данный момент. Среди наших соотечественников задачей прогнозирования занимаются разработчики Яндекс.Пробок. Программисты 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 альтернативных 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 известны, и они не меняются с течением времени. Это не умаляет значимости и важности поставленной нами задачи. Путем моделирования движения можно получить более точные результаты, идеи в дальнейшем могут развиваться и открывать новые возможности для более широкой задачи, например, как у Яндекса. Также стоит отметить, что разработчики Яндекс.Пробок 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.

Отдельное внимание стоит уделить построению математической модели для описания движения АТС. Существует множество различных моделей движения, например, в модели Лайтхилла–Уизема (Уитема)–Ричардса (LWR) [1], [2] однополосный транспортный поток рассматривается как поток одномерной сжимаемой жидкости. Модель LWR была первой среди гидродинамических. Теория развивалась и появлялись новые модели, такие как модель Танака [3] — LWR-модель, где плотность потока зависит от некоторой дистанции видимости, зависящей от скорости потока. В микроскопической модели Ньюэлла [4] постулируется, что для каждого водителя существует безопасная скорость движения, зависящая от дистанции до лидера. В своей работе мы позаимствовали деление моделей движения на макро- и микроскопические [5] и разработали свои модели взаимодействия участников движения.

Задачи, которые мы ставили перед собой в рамках выбранной темы дипломной работы: формальная постановка задачи, анализ полученных ранее результатов в этой теме, разработка модели движения, которая была бы близка к естественному характеру взаимодействия АТС, исследование применимости различных алгоритмов к решению поставленной задачи. Методом исследования является моделирование движения.

В первом разделе вы сможете ознакомиться с определением модели движения как некоего автомата и деталями поставленной задачи. Во втором разделе будет представлена классификация моделей движения с примерами. Далее мы опишем процесс моделирования произвольной дорожной ситуации. Четвертый раздел посвящен описанию алгоритмов и исследованию их применимости. 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 $l : E \rightarrow \mathbb{R}$. Также мы говорим, что задана некоторая модель движения. Обратимся к теории автоматов, чтобы попытаться формализовать подразумеваемые под этим понятием правила движения АТС. Аналогичный подход использовали К. Нагель и М. Шрекенберг [6]. 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 [7], а также в исследовании пробок на дорогах [8]. 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, $t_i : S^n \rightarrow \mathbb{R}_{\geq 0}$ - функция критического момента движения участника i , $\varphi_i : S^n \times \mathbb{R}_{\geq 0} \rightarrow S$ - функция перехода состояния i -ого участника в некоторый момент времени t . Считаем, что, попав в заключительное состояние, мы не можем его покинуть:

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

Множество S описывает текущий характер движения 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 $t = t_i(s_1, \dots, s_n)$ для некоторого s_i из класса состояний. Непомеченные ребра соответствуют условию $t < t_i(s_1, \dots, s_n)$:

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

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

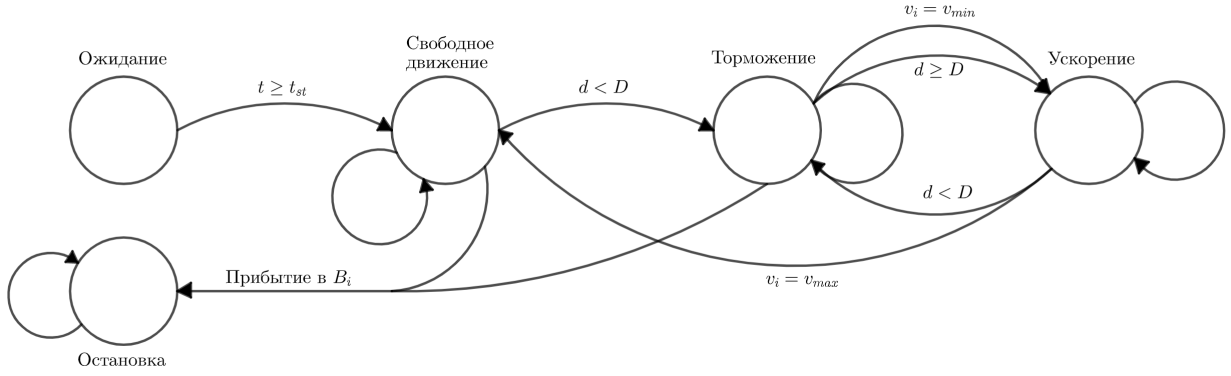


Рис. 1: Диаграмма для i -ого участника в модели движения, где d – расстояние до впереди идущего участника, D – максимальное расстояние взаимодействия с впереди идущим участником, v_{max} – максимально возможная скорость, v_{min} – минимально возможная скорость, t_{st} – время старта.

Перейдем от формального определения модели движения непосредственно к постановке задачи. Пусть имеется n участников, которые движутся по заранее заданным маршрутам: $p_i = \langle E_{j_1}^i, E_{j_2}^i, \dots, E_{j_{m_i}}^i \rangle, E_{j_k}^i \in E \quad i = 1, \dots, n$. 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

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

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, \dots, p_n требуется найти такой путь p^* из A в B , что движение нового участника по этому пути p^* будет *оптимально*, то есть

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

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, \dots, 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 $n(n-1) + 2n$.

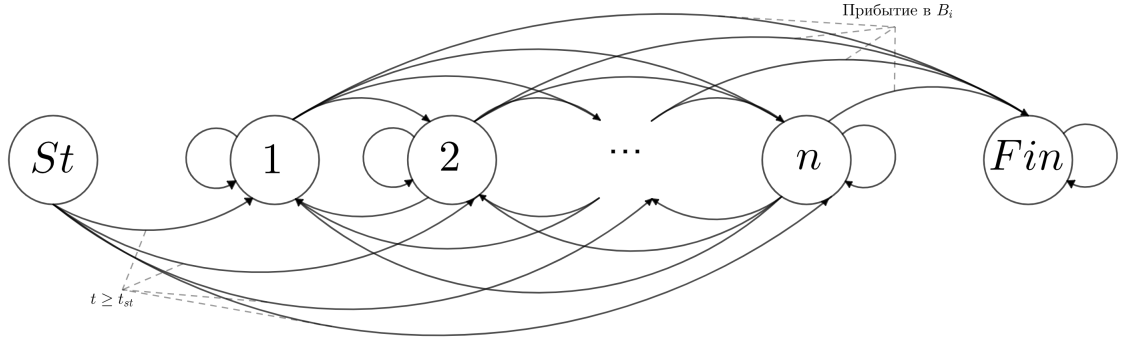


Рис. 2: Диаграмма для i -ого участника в макроскопических моделях движения

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

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

Для того, чтобы критические моменты $t_i, i = 1, \dots, n$ были определены, макроскопические модели требуют положительности скоростей $v_i, i = 1, \dots, n$. Условие $v_i \geq v_{\min} > 0, i = 1, \dots, n$ 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, $k \in \{1, \dots, n\}$, 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 разогнаться 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.

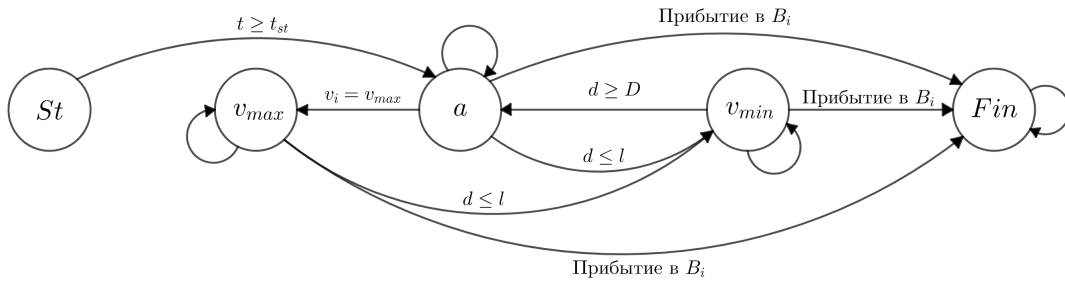


Рис. 3: Диаграмма для i -ого участника в модели следования за лидером

3 Моделирование

Под моделированием будем понимать воспроизведение движения $n + 1$ участника, движущихся по путям p_1, \dots, p_n, p , $p \in P(A, B)$ в модели движения $M = M(p_1, \dots, p_n, p)$. 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 $M = M(p_1, \dots, p_n, p)$, набор начальных состояний $s_1^\circ, \dots, s_{n+1}^\circ$

Output: $T(p)$

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

```
1:  $t = 0$ 
2: for  $i = 1, \dots, n + 1$  do
3:    $s_i \leftarrow s_i^\circ$ 
4: end for
5: while  $s_{n+1} \notin F$  do
6:   for  $i = 1, \dots, n + 1$  do
7:      $t_i^* \leftarrow t_i(s_1, \dots, s_{n+1})$ 
8:   end for
9:    $t^* \leftarrow \min(t_1^*, \dots, t_{n+1}^*)$ 
10:  for  $i = 1, \dots, n + 1$  do
11:     $s_i \leftarrow \varphi_i(s_1, \dots, s_{n+1}, t^*)$ 
12:  end for
13:   $t \leftarrow t^*$ 
14: end while
15:  $T(p) \leftarrow t$ 
```

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 $S_M(p)$ – сложность моделирования, т.е. нахождения функции $x_{n+1}^p(t)$, при выборе пути p . Тогда сложность перебора

$$S = \sum_{p \in P(A,B)} S_M(p) = |P(A,B)| * \bar{S}_M, \text{ где } \bar{S}_M - \text{средняя сложность.}$$

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

В качестве примера можем рассмотреть 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 Манхэттенна будет составлять $C_{|a_1-b_1|+|a_2-b_2|}^{|a_1-b_1|}$. Понятно, что путей $|P(A,B)|$ в таком графе гораздо больше. Таким образом, получаем оценку снизу для регулярного решеточного графа

$$C_{|a_1-b_1|+|a_2-b_2|}^{|a_1-b_1|} * \bar{S}_M \leq |P(A,B)| * \bar{S}_M = S.$$

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

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

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 Альтернативный подход к решению

Рассмотрим вспомогательную задачу. Пусть на каждом ребре $e \in E$ графа $G(V, E)$ определена функция *временных затрат* $\phi_e(t) : \mathbb{R}_+ \rightarrow \mathbb{R}_+$. 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 $\phi_e(t)$:

$$\phi_e(t) \leq \Delta + \phi_e(t + \Delta), \quad \Delta \geq 0 \quad (1)$$

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. Тогда если S_ϕ – сложность вычисления функций ϕ_e , $e \in E$, то сложность модифицированного алгоритма $S = |E|S_\phi$.

Для запуска алгоритма потребуется задать начальное время - минимальное время в точке старта. Это можно использовать в анализе маршрута.

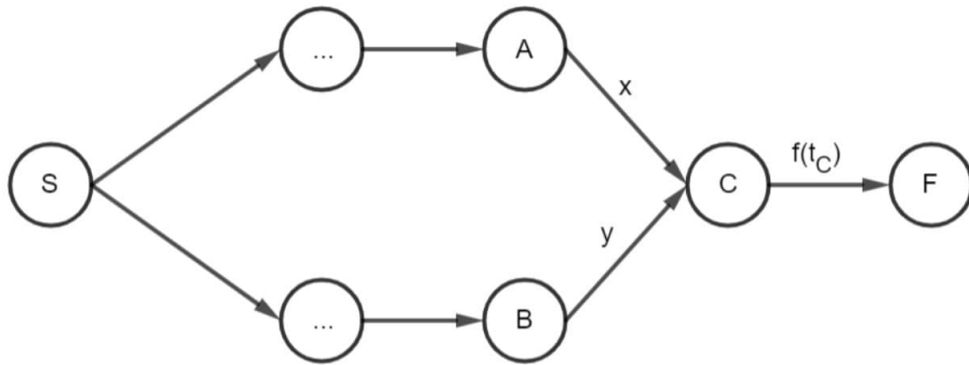
Утверждение 4.1. *Данный маршрут обладает наименьшим временем прохождения.*

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 $t_a = T(S - \dots - A - C)$, а время для части маршрута P из S в C , проходящего через точку B , за $t_b = T(S - \dots - B - C)$. 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 $P : S - \dots - B - C - F$ и $P' : S - \dots - A - C - F$. 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.

□

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. 4):

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

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 .

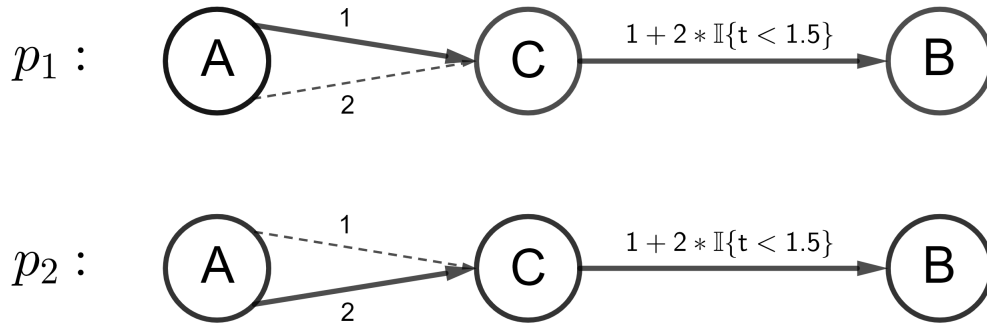


Рис. 4: Пример графа с невыполненным условием неравенства прохождения ребер

Отметим, что наша задача поиска оптимального маршрута сводится к вспомогательной задаче. Правила движения участников и их взаимодействий определяют функции ϕ_e , $e \in E$. Значения этих функций можно получить путем моделирования движения. Тогда $S_\phi = S_M$ и $S = |E|S_M$.

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.

4.2.2 Применимость модифицированного алгоритма Дейкстры

Поскольку неравенство прохождения ребер является гарантией того, что модифицированный алгоритм Дейкстры дает оптимальное решение, мы хотели бы исследовать выполнение

этого неравенства в интересующих нас моделях движения. Оказывается, что для любой макроскопической модели верно

Утверждение 4.2. *Если в модели движения M процесс моделирования движения зависит только от положения участника на ребре, то в M выполняется неравенство прохождения ребер.*

Доказательство. Докажем от противного. Рассмотрим одного участника и его копию, которая могла двигаться в другое время по другому ребру. И участник, и копия зависят от движений других участников, но друг на друга не влияют. Итак, пусть участник въезжает на ребро e в разные моменты времени t_1 и t_2 , $t_2 > t_1$ с ребер e_1 и e_2 соответственно. Пусть $\chi(t)$ – часть пройденного ребра e . Предположим, что $\phi_e(t_1) > (t_2 - t_1) + \phi_e(t_2)$, тогда в какой-то момент времени t $\chi_1(t) = \chi_2(t) \Rightarrow$

□

В микроскопических моделях условия выполнимости неравенства не найдены, а как мы уже выяснили на примере (см. рис. 4), модифицированный алгоритм Дейкстры может выдать не оптимальное решение задачи, если неравенство прохождения ребер не выполнено. Беря во внимание этот факт, мы предлагаем не отказываться от применения алгоритма и к микроскопическим моделям. Для них можно посчитать погрешность алгоритма.

Обратимся к описанной нами микроскопической модели (см. рис. 3) и рассмотрим случай, когда участник мог бы «обогнать» самого себя, если бы начал движение позднее. Опишем худший случай в нашей модели на бесконечном ребре. Участник оказывается на расстоянии $d \approx 0$ до лидера и теряет свою скорость до v_{min} . Пусть лидер и участник со сдвигом по времени двигаются со скоростями v_{max} на расстоянии $l + \varepsilon$, $\varepsilon > 0$ друг от друга и они не меняют своих скоростей. В таком случае участнику, близкому к лидеру, нужно отдалиться на безопасное расстояние D за $t_1 = \frac{D}{v_{max} - v_{min}}$ и разогнаться до скорости v_{max} за $t_2 = \frac{v_{max} - v_{min}}{a}$. За это время $t = t_1 + t_2$ участник со сдвигом преодолеет расстояние $s = v_{max}t$. Расстояние между участником и его копией через время t составит

$$\Delta s = D - l + \frac{(v_{max} - v_{min})^2}{2a}.$$

Практические результаты

Заключение

В ходе написания дипломной работы были совершены следующие шаги:

- Предложена автоматная форма определения модели движения АТС.
- Разработан и реализован алгоритм симуляции движения АТС в соответствии с заданной моделью движения.
- Сформулировано необходимое условие, при котором модифицированный алгоритм Дейкстры приводит к нахождению оптимального решения.
- Показано, что для модели следования за лидером возможно отклонение найденного решения от оптимального.

Исследование может иметь продолжение в различных направлениях: разработки более сложной и приближенной к реальности модели движения; рассмотрение случая, когда добавленный участник влияет на движение всех остальных АТС; проверки устойчивости найденного решения; поиска аналогий задачи или подзадач в разных областях математики.

Список литературы

- [1] *Lighthill M. J., Whitham G. B.* On kinematic waves: II. Theory of traffic flow on long crowded roads // Proc. R. Soc. London, Ser. A. 1955. V. 229. P. 281–345.
- [2] *Richards P. I.* Shock Waves on the Highway // Oper. Res. 1956. V. 4. P. 42–51.
- [3] *Иносэ Х., Хамада Т.* Управление дорожным движением. М.: Транс- порт, 1983.
- [4] *Newell G. F.* Nonlinear effects in the dynamics of car – following // Oper. Res. 1961. V. 9. P. 209–229.
- [5] *А. И. Гасников* “Введение в математическое моделирование транспортных потоков” — Издательство МЦНМО — 2013. — 427 с.
- [6] *Nagel K., Schreckenberg M.* A cellular automation model for freeway traffic // Phys. I France. 1992. V. 2. P. 2221–2229.
- [7] *Chowdhury D., Santen L., Schadschneider A.* Statistical physics of vehicular traffic and some related systems // Phys. Rep. 2000. V. 329. P. 199–329.
- [8] *Nagatani T.* The physics of traffic jams // Reports on Progress in Physics. 2002. V. 65. P. 1331–1386.