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

Графом дорожной сети назовем ориентированный граф $G(V, E, l)$, в котором вершины $v \in V$ осуществляют роль перекрестков, а ребра $e \in E$ - роль дорог, и каждое ребро имеет длину, т.е. задана функция $l : E \to \mathbb{R}$.

Пусть имеется n участников, которые движутся по заранее заданным маршрутам: $p_i = \langle E^i_{j_1}, E^i_{j_2}, \ldots, E^i_{j_{m_i}} \rangle$, $E^i_{j_k} \in E \quad i = 1, \ldots, n$. И пусть для каждого участника определена *скорость* движения $v_i^{p_i}(G, \cdot) : \mathbb{R}_+ \to \mathbb{R}_+$ — функция, интегрируемая на $[0, +\infty)$. Зададим систему *правил движения* АТС — набор ограничений на скорости $v_i^{p_i}$ и законов их изменения при взаимодействии участников друг с другом.

Пусть имеется $(n+1)$-ый участник, которому нужно добраться из пункта A в пункт B, $A, B \in V$. Определим $P(A, B)$ – множество всех простых путей из A в B. Правила движения позволяют определить *часть пройденного пути* для каждой АТС в зависимости от движения других участников, т.е. определить функции

$$x_i^{p_i}(G, t) = \frac{\int\limits_0^t v_i^{p_i}(G, \tau) d\tau}{\sum\limits_{e \in p_i} l(e)},\ i = 1, \ldots, n$$

$$x_{n+1}^{p}(G, t) = \frac{\int\limits_0^t v_{n+1}^{p}(G, \tau) d\tau}{\sum\limits_{e \in p} l(e)},\ p \in P(A, B),$$

где $v_{n+1}^p(G, \cdot) : \mathbb{R}_+ \to \mathbb{R}_+$ интегрируемая функция скорости $(n+1)$-ого участника. Ввиду однозначной определенности последовательности ребер, по которым осуществляется движение, индекс p_i, означающий путь, для заданных n участников можно опустить. Также для простоты записи не будем писать зависмость функций от графа G. Будем говорить, что функции x_i, $i = 1, \ldots, n$ и x_{n+1}^p, $p \in P(A, B)$ *задают модель движения* АТС.

На множестве путей $P(A, B)$ определим $T(p) = \inf\limits_t \{t : x_{n+1}^p(t) = 1\}$ – 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, на графе дорожной сети $G(V, E, l)$ при движении n участников, описанному функциями x_i, для заданных вершин A и B требуется найти путь

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

2 Возможные правила движения

Поведение движения АТС в нашей постановке задачи определяется некоторой системой правил. Представим примеры таких приавил, которые описывают естественный, близкий к реальности характер движения автомобилей.

Говоря о поведении движения автомобилей, мы имеем в виду изменение их скоростей в зависимости от ситуации на дороге. Обозначим через $v_m \in \mathbb{R}$ максимально возможную скорость, или *скорость свободного движения*: $v_i(t) \leq v_m, \forall t \in \mathbb{R}, i = 1, \ldots, n+1$.

2.1 Микроскопические модели

Опишем модель *следования за лидером* – модель, в которой поведение движения участника зависит от расстояния до впереди идущего АТС на ребре. Отметим, что лидер может влиять на движение участника, только если они находятся по одном ребре. Введем понятие *дистанции видимости* D — расстояние, на котором текущий участник начинает взаимодействовать с впереди идущим лидером. Также будем считать, что участники должны соблюдать *безопасное расстояние* до лидера l — тормозной путь, на котором нужно успеть сбросить скорость до нуля. Если в начале ребра участник оказался на расстоянии $d \leq l$ до лидера, то он мгновенно сбрасывает свою скорость до нуля. Скорости участников будут изменяться посредством ускорений, положим

$$a_+ = \frac{v_m^2}{D}$$

— ускорение такое, что позволяет разогнаться с нулевой до максимальной скорости за время, которое нужно для преодоления расстояния D с постоянной максимальной скоростью. Также пусть

$$a_- = -\frac{v_m^2}{2l}$$

— торможение такое, что позволяет снизить скорость с максимальной до нуля, проехав расстояние l.

Пусть i -ый и $(i+1)$ -ый участники, лидер и следующий за ним соотвественно, оказались в момент времени t на расстоянии d относительно друг друга со скоростями v_i и v_{i+1}. Опишем характер их движения.

1. $d \geq D$
   (a) $v_{i+1} = v_{max} \Rightarrow a_{i+1} = 0$, продолжаем ехать с максимальной скоростью.
   (b) $v_{i+1} < v_{max} \Rightarrow a_{i+1} = a_+$ на время $t = \frac{v_{max} - v_{i+1}}{a_+}$, после чего $v_{i+1} = v_{max}, a_{i+1} = 0$
2. $d \leq l \Rightarrow v_{i+1} = 0$ (в начале ребра)
3. $l < d < D$
   (a) $v_{i+1} \leq v_i$

i. $a_i = 0$, т.е. его движение равномерно, это может быть только в 2-х случаях:

A. $v_i = v_{max} \Rightarrow a_{i+1} = a_+$ на время $t = \frac{v_{max}-v_{i+1}}{a_+}$, после чего $v_{i+1} = v_{max}, a_{i+1} = 0$

B. $v_i = 0$, едем время $t = \frac{d-l}{v_{i+1}}$, не меняя скорость, после чего останавливаемся, $v_{i+1} = 0$.

ii. $a_i \neq 0 \Rightarrow a_{i+1} = a_i$

(b) $v_{i+1} > v_i$

i. $a_i = a_+ \Rightarrow a_{i+1} = a_-$ до момента, когда $v_i + a_i t = v_{i+1} + a_{i+1} t$ или v_{i+1} станет 0, т.е. до $t = min\left(\frac{2Dl(v_{i+1}-vi)}{v_{max}^2(l+D)}, \frac{2lv_{i+1}}{v_{max}^2}\right)$, после чего $a_{i+1} = a_i = a_+$.

ii. $a_i = a_- \Rightarrow a_{i+1} = a_i = a_-$.

Описанные правила движения диктуют непрерывность функций v_i, $i = 1, \ldots, n+1$ на ребрах, точки разрыва возникают только в вершинах. В графе дорожной сети $|V| < \infty \Rightarrow$ функции скорости интегрируемы и подходят под условия задачи.

2.2 Макроскопические модели

1.

3 Моделирование

Под моделированием будем понимать воспроизведение движения всех АТС по установленным правилам. Рассмотрим 2 различных способа моделирования с $(n+1)$-ым участником:

1. он влияет на движение n участников и вследствие на самого себя;

2. движение n участников не зависит от добавленного АТС.

3.1 Независимость движения участников от добавленного АТС

3.2 Взаимосвязь движений нового участника и группы АТС

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 $p = \langle V_0, e_1, V_1, e_2, V_2, \ldots, V_{k-1}, e_k, V_k \rangle$ и начало движения происходит в вершине V_0 в момент времени $t_0 = t$, тогда

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

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

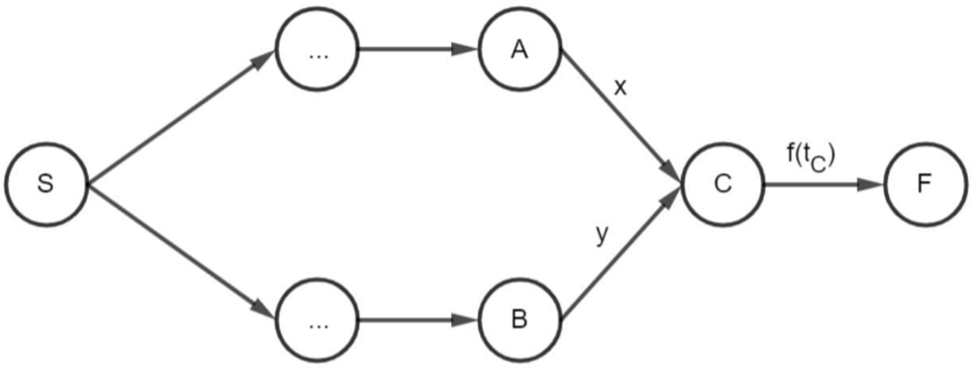
Алгоритм построения минимального маршрута

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.

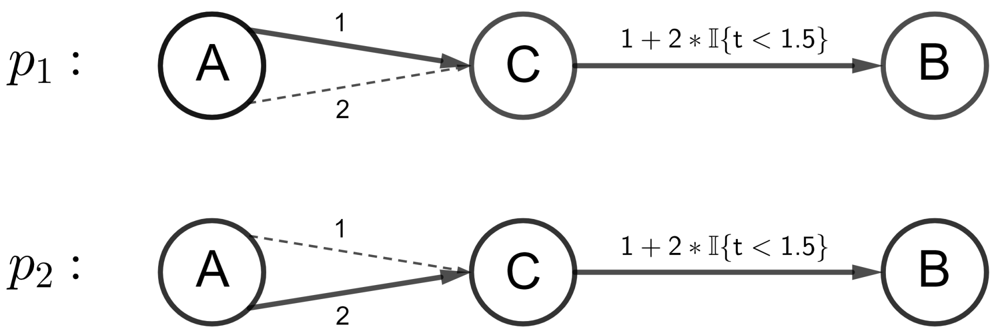


Рис. 1: Пример графа с невыполненным условием неравенства прохождения ребер

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