

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:
кандидат 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	Поиск кратчайшего пути в динамическом графе	11
4.2.1	Модифицированный алгоритм Дейкстры	12
4.2.2	Применимость модифицированного алгоритма Дейкстры	14
5	Практические результаты	16
	Литература	20

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.

В наше время задачи на транспортную тематику приобретают особую актуальность, 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.

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. Также стоит отметить, что разработчики Яндекс.Пробок используют статистические методы и машинное обучение в качестве инструмента для решения своих задач.

Мы рассматриваем более прозрачную и простую модель, когда все маршруты изначально проложены и известны, и они не меняются с течением времени. Такой уровень информированности о движении участников вряд ли достижим в реальной жизни, но 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.

Важную роль в задачах маршрутизации играет представление правил движения АТС с помощью формальной математической модели. Она должна быть подобрана на основе имеющихся данных. 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 сре-

ди 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 задачи оптимизации является моделирование движения.

В первом разделе введем определение модели движения как некоторого автомата и в этих терминах опишем поставленную задачу. 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 завершение получим результаты тестирования алгоритмов и сделаем выводы.

1 Постановка задачи

Пусть задан ориентированный *граф дорожной сети* $G(V, E, l)$ таким образом, что вершины V осуществляют роль перекрестков, а ребра 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 попробовать 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 $M = (n, G, S, \{s_i^0\}_{i=1}^n, \{F_i\}_{i=1}^n, \{t_i\}_{i=1}^n, \{\varphi_i\}_{i=1}^n)$, где n — количество участников движения, G — граф дорожной сети, S — множество состояний, которые могут принимать участники, $s_i^0 \in S$ — начальное состояние участника i , $F_i \subset S$ — множество заключительных состояний участника i , $t_i : S^n \rightarrow \mathbb{R}_{\geq 0}$ — 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 .

Выбор множества S зависит от правил движения, которые мы хотим описать. Например, правила движения могут учитывать положение на графе $(e, x) \in E \times [0; 1]$, скорость $v \in \mathbb{R}$ участника и время в пути $t \in \mathbb{R}$. В таком случае считаем $S = E \times [0; 1] \times \mathbb{R}^2$. 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. Считаем, что, попав в заключительное состояние, мы не можем его покинуть:

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

Модель движения АТС можно изобразить 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, могут быть заданы следующей диаграммой (см. рис. 1):

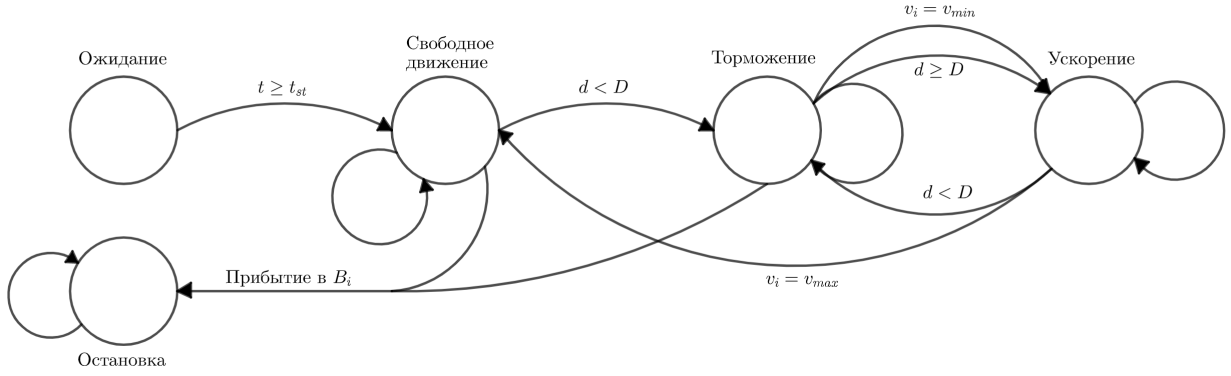


Рис. 1: Диаграмма для i -ого участника в модели движения с множеством состояний $S = \{(e, x, v, t) : e \in E, x \in [0, 1], v \in [v_{min}, v_{max}], t \in \mathbb{R}\}$, где d – расстояние до впереди идущего участника, D – максимальное расстояние взаимодействия с впереди идущим участником, v_{max} – максимально возможная скорость, v_{min} – минимально возможная скорость, t_{st} – время старта.

Заметим, что в данном примере все зависимости с характеристикой состояния «время в пути» можно перенести в функции t_i и φ_i . Поэтому можно рассматривать множество состояний $S = \{(e, x, v) : e \in E, x \in [0, 1], v \in [v_{min}, v_{max}]\}$.

Далее ограничимся только теми моделями движения, в которых все участники достигают своих заключительных состояний за конечное время. Такие модели назовем *вычислимыми*.

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 $A \in V$ в пункт $B \in V$. Определим $P(A, B)$ – множество всех простых путей из A в B . Пусть $(n+1)$ -ый участник двигается по некоторому пути $p \in P(A, B)$. Далее будем считать, что начальные состояния в модели движения M соответствуют началам путей p_1, \dots, p_n, p , а заключительные – концам этих путей. Вычислимая модель движения позволяет определить момент времени, когда $(n+1)$ -ый участник достиг заключительного состояния, двигаясь по маршруту p . Обозначим его $T(p)$.

Требуется найти такой путь $p^* \in P(A, B)$, что $T(p^*)$ – минимален. Другими словами, для заданной модели движения M 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

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

Предполагаем ничтожно малое влияние $n+1$ -ого участника на движение остальных, хотя такое предположение и заведомо неверно для крупногабаритных транспортных средств, для легковых автомобилей оно представляется достаточно разумным, исключая случаи аварийных ситуаций или неопытных водителей за рулем.

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 . Такие модели называются *макроскопическими*, а все остальные — *микроскопическими*. В этом разделе будет показано, что как макроскопические, так и микроскопические модели могут быть выражены в терминах нашей автоматной модели движения.

2.1 Макроскопические модели

Множество состояний в макроскопических моделях движения определяется только положением на графе, т.е. $S = E \times [0, 1]$. Эти состояния 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 $n(n - 1) + 2n$.

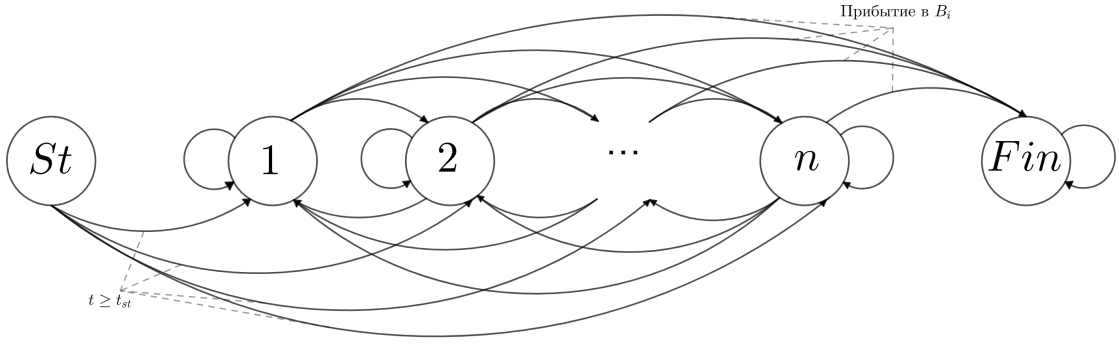


Рис. 2: Диаграмма для i -ого участника в макроскопических моделях движения

Пусть $s_i = (e_i, x_i)$ — положение участника i на графе дорожной сети. Пусть $n_i(s_1, \dots, s_n)$ — количество участников на ребре e_i . Переход из класса состояний k в класс состояний m осуществляется 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:

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

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 $|p_i|$ в пути p_i , из чего можно сделать вывод, что все заключительные состояния достижимы, а следовательно, модель движения вычислима.

Примером такой модели может послужить набор

$$v_k = (v_{max} - v_{min}) \left(1 - \frac{k}{n}\right) + v_{min}, \quad k \in \{1, \dots, n\}, \quad (1)$$

где v_{max}, v_{min} – заданные максимальная и минимальная скорости.

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 $0 < c \leq v_{min} < v_{max}$ скорости участников ($c = const$), 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.

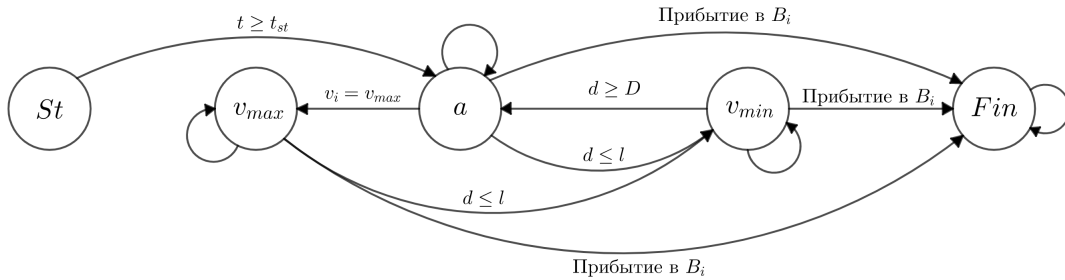


Рис. 3: Диаграмма для i -ого участника в модели следования за лидером.

В данной модели множество состояний есть $S = \{(e, x, v) : e \in E, x \in [0, 1], v \in [v_{min}, v_{max}]\}$.

Для каждого перехода между классами состояний время ограничено снизу положительной константой, т.к. разгон до максимальной скорости и отдаление от лидера происходят не мгновенно. Поскольку минимальная скорость также ограничена снизу, каждый участник достигнет конца своего пути. Следовательно, описанная модель движения является вычислимой.

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:  $T(p) \leftarrow t$ 

```

Процесс моделирования представляет собой последовательную смену состояний участников. Поэтому для сложности алгоритма $S_M(p_1, \dots, p_n, p)$ верно

$$S_M(p_1, \dots, p_n, p) = S_{\phi, t}(p_1, \dots, p_n, p) N_t(p_1, \dots, p_n, p),$$

где $S_{\phi, t}$ — сложность обновления на шаге 6 - 13, а N_t — количество смен состояний, то есть количество циклов шага 5.

Опишем сложность моделирования макроскопической модели. Количество смен состояний в такой модели равно количеству смен ребер. Поэтому $N_t(p_1, \dots, p_n, p) = |p| + \sum_{i=1}^n |p_i|$. Количество операций при обновлении состояний зависит от n линейно. Таким образом, сложность моделирования составляет $S_M(p_1, \dots, p_n, p) = n \left(|p| + \sum_{i=1}^n |p_i| \right)$.

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 $S_M(p_1, \dots, p_n, p)$ – сложность моделирования, т.е. нахождения времени $T(p)$, потраченного на путь p . Тогда сложность перебора

$$S = \sum_{p \in P(A, B)} S_M(p_1, \dots, p_n, p) = |P(A, B)| \bar{S}_M, \text{ где } \bar{S}_M - \text{средняя сложность моделирования.}$$

Заметим, что S растет при увеличении количества возможных путей. Так, в полном графе на $|V|$ вершинах получим $S = \left(\sum_{k=1}^{|V|-2} k! \right)^n \bar{S}_M$.

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

$$C_{2m}^m = \frac{\sqrt{2\pi(2m)} \left(\frac{2m}{e}\right)^{2m} + \tilde{o}(1)}{\left(\sqrt{2\pi m} \left(\frac{m}{e}\right)^m + \tilde{o}(1)\right) \left(\sqrt{2\pi m} \left(\frac{m}{e}\right)^m + \tilde{o}(1)\right)} = \frac{2\sqrt{\pi m} \left(\frac{2m}{e}\right)^{2m}}{2\pi m \left(\frac{m}{e}\right)^{2m}} + \tilde{o}(1) = \frac{2^{2m}}{\sqrt{\pi m}} + \tilde{o}(1).$$

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 Поиск кратчайшего пути в динамическом графе

Граф $G(V, E)$ будем называть *динамическим*, если на его ребрах определены динамические затраты $\phi_e(t) : \mathbb{R}_+ \rightarrow \mathbb{R}$, $e \in E$. Если мы оказались в начальной вершине ребра e в момент времени t , то время преодоления ребра будет равняться $\phi_e(t)$.

Рассмотрим путь $p = \langle V_0, e_1, V_1, e_2, V_2, \dots, V_{k-1}, e_k, V_k \rangle$, начало движения происходит в вершине V_0 в момент времени t_0 , а вершину V_i мы посещаем в некоторый момент времени t_i . Тогда

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

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

$$\hat{p} = \underset{p \in P(A, B)}{\operatorname{argmin}} t_{|p|}. \quad (2)$$

Множество всех решений задачи (2) обозначим $\hat{P}(A, B)$.

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

$$\phi_e(t) < \Delta t + \phi_e(t + \Delta t), \quad \Delta t > 0 \quad (3)$$

Назовем условие (3) *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.

Лемма 4.1. Пусть для функции $\phi_e(t)$ выполняется условие (3). Множество решений задачи (2) есть $\hat{P}(A, B) \subset P(A, B)$. Рассмотрим вершину C – вершина из некоторого пути $\hat{p} \in \hat{P}(A, B)$. Ограничим путь \hat{p} вершиной C и обозначим такой путь как $\pi_C(\hat{p})$. Тогда верно $\pi_C(\hat{p}) \in \hat{P}(A, C)$.

Доказательство. От противного. Предположим $\pi_C(\hat{p}) \notin \hat{P}(A, C)$. Без ограничения общности считаем, что в G нет кратных ребер, и C – предпоследняя точка пути p . Тогда можно считать, что $\hat{p} = p_1 DCB$, где p_1 – некоторый некоторый путь из A . Значит, $\pi_C(\hat{p}) = p_1 DC$.

Рассмотрим некоторый $\hat{q} = p_2 FC \in \hat{P}(A, C)$. Тогда верно

$$t_F^{p_2} + \phi_{FC}(t_F^{p_2}) < t_D^{p_1} + \phi_{DC}(t_D^{p_1}), \quad (4)$$

где $t_F^{p_2}$ – время прибытия в вершину F через путь p_2 , $t_D^{p_1}$ – время прибытия в вершину D через путь p_1 . Обозначим за $t_C^{p_2 F}$ и $t_C^{p_1 D}$ левую и правую часть неравенства (4) соответственно.

В новых обозначениях $t_C^{p_2^F} < t_C^{p_1^D}$. Рассмотрим путь $q = \hat{q}B = p_2FCB \in P(A, B)$. Поскольку \hat{p} - решение задачи имеем

$$t_C^{p_2^F} + \phi_{CB}(t_C^{p_2^F}) \geq t_C^{p_1^D} + \phi_{CB}(t_C^{p_1^D}).$$

или,

$$\phi_{CB}(t_C^{p_2^F}) \geq (t_C^{p_1^D} - t_C^{p_2^F}) + \phi_{CB}(t_C^{p_2^F} + (t_C^{p_1^D} - t_C^{p_2^F})). \quad (5)$$

Противоречие с условием (3), где $\Delta t = t_C^{p_1^D} - t_C^{p_2^F} > 0$. \square

Таким образом, лемма 4.1 утверждает, что любой подпуть оптимального пути задачи (2) является оптимальным, а условие (3) является достаточным признаком выполнения такого свойства.

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.

Утверждение 4.1. Пусть для функции $\phi_e(t)$ выполняется условие (3). Тогда алгоритм Дейкстры для каждого $B \in V$ позволяет найти хотя бы одно решение задачи (2).

Доказательство. Обозначим $t(v)$ — минимальные затраты в задаче (2) с $v \in V$, $d(v)$ — затраты на вершине $v \in V$, обновляемые в ходе работе алгоритмом. Докажем по индукции, что в момент посещения любой вершины B выполняется равенство $d(v) = t(v)$.

База. Первая посещаемая вершина есть A . В этот момент $d(A) = t(A) = t_0$.

Шаг. Пусть мы выбрали для посещения вершину $B \neq A$. Докажем, что в этот момент $d(B) = t(B)$. Для начала отметим, что для любой вершины $v \in V$ всегда выполняется $d(v) \geq t(v)$, поскольку алгоритм не может найти путь с затратами меньше минимальных. Пусть \hat{p} — путь из A в B с минимальными затратами. Вершина B находится на \hat{p} и не посещена. Поэтому множество непосещённых вершин на \hat{p} непусто. Пусть C — первая непосещённая вершина на \hat{p} , D — предшествующая ей, и, следовательно, посещённая. Поскольку путь \hat{p} оптимальный, то по лемме 4.1 его часть, ведущая из A через D в C , тоже оптимальна, следовательно $t(C) = t(D) + \phi_{DC}(t(D))$.

По предположению индукции, в момент посещения вершины D выполнялось $d(D) = t(D)$, следовательно, вершина C тогда получила затраты не больше чем $t(D) + \phi_{DC}(t(D)) = t(C)$. Следовательно, $d(C) = t(C)$. Если $C = B$, то шаг индукции доказан. Иначе, поскольку сейчас выбрана вершина B , а не C , затраты $d(B)$ минимальны среди непосещённых, то есть $d(B) \leq d(Y) = t(Y) \leq t(B)$. Учитывая, что $d(B) \geq t(B)$, имеем $d(B) = t(B)$, что и требовалось доказать.

Поскольку алгоритм заканчивает работу, когда все вершины посещены, в этот момент $d(v) = t(v), v \in V$. \square

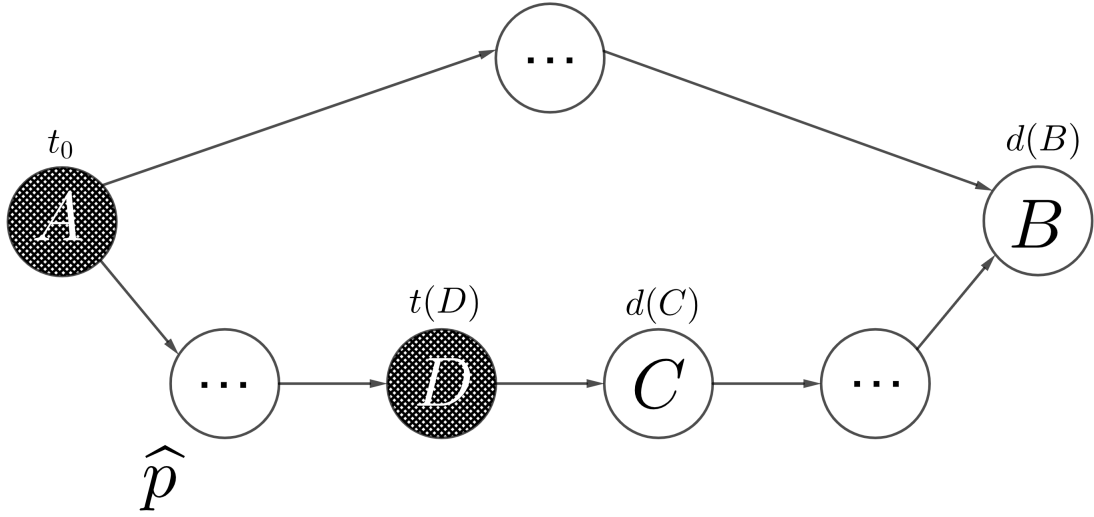


Рис. 4: Шаг индукции в доказательстве утверждения 4.1.

Если \bar{S}_ϕ – средняя сложность вычисления функций ϕ_e , $e \in E$, то средняя сложность модифицированного алгоритма Дейкстры есть $S = |E|\bar{S}_\phi$.

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. 5). Пусть $\phi_e(t)$ — время, которое требуется на прохождение ребра e . Пусть из A в C ведут два ребра с временными затратами 1 и 2, а время, необходимое для преодоления ребра CB , зависит от момента начала движения по этому ребру. Другими словами, если в вершине C мы оказались позже, а именно, в момент времени $t > 1.5$, то времени на участок CB понадобится меньше (например, мы пропустили час-пик, и загруженность дорог стала меньше). Тогда временные затраты возможных путей в таком графе:

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

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 .

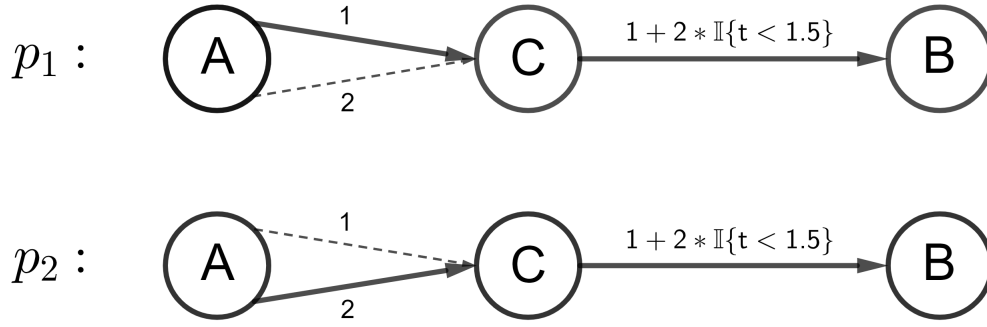


Рис. 5: Пример графа с невыполненным условием неравенства прохождения ребер.

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 $\bar{S}_\phi = \bar{S}_M$ и $S = |E|\bar{S}_M$.

Неравенство прохождения ребер можно переформулировать так: дорожная сеть обладает условием FIFO — если участник i въехал на дорогу позже участника j , то i -ый участник покинет ее не раньше j -ого. 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. Пусть в модели движения M известно, что выбор пути участника $n + 1$ не влияет на движение других участников. Предположим, что состояния в модели движения описываются только положением участника на графе, то есть $S = E \times [0, 1]$. Тогда в M выполняется неравенство прохождения ребер (3) для участника $n + 1$.

Доказательство. Докажем от противного. Рассмотрим одного участника и его копию, которая могла двигаться в другое время по другому ребру. Пусть в какой-то момент копия участника $n + 1$ приехала в ребро раньше, чем другая копия, однако выехала из ребра позже. Тогда существует момент времени и состояние, для которых $s_{n+1}^1(t) = s_{n+1}^2(t)$. Тогда $s_i^1(t) = s_i^2(t)$, $i = 1, \dots, n + 1$. Поскольку процесс моделирования можно обратить, получим, что обе копии приехали в это ребро одновременно. Противоречие. \square

В микроскопических моделях условия выполнимости неравенства не найдены, а как мы уже выяснили на примере (см. рис. 5), 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

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

Такая разница в расстоянии между возможными положениями участника в разные моменты времени достигается при преодолении одного перекрестка-вершины. При дальнейшем движении по графу может случиться ситуация, когда обогнавшая копия участника пройдет следующий перекресток без торможений, а участник позади встанет в очередь АТС, прибывших с других ребер. В худшем случае участник и его копия будут преодолевать оставшийся путь p с минимальной и максимальной скоростями соответственно:

$$t_{\text{погр}} = \frac{\Delta s}{v_{max}} + \sum_{e \in p} \frac{l(e)}{v_{min}} - \frac{l(e)}{v_{max}} = \frac{\Delta s}{v_{max}} + \sum_{e \in p} l(e) \frac{v_{max} - v_{min}}{v_{min} v_{max}}.$$

5 Практические результаты

Рассмотрим работу алгоритма на следующем графе дорожной сети G (см. рис. 6).

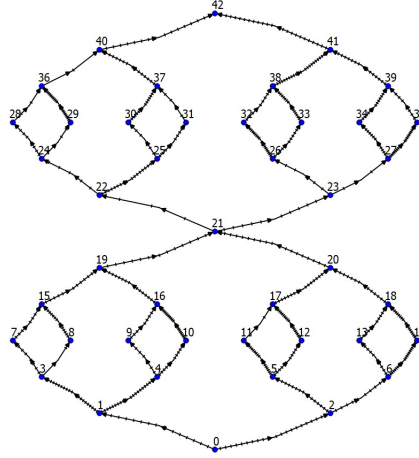


Рис. 6: Граф дорожной сети G , на котором производилось моделирование движения. $|V| = 43$, $|E| = 52$, ребра произвольной длины. Одно деление на ребре соответствует расстоянию $L = 500$).

Случайным образом запустим $n = 50$ участников из вершины с номером 0 в вершину с номером 42. Выбор такого графа обусловлен тем, что в нем имеются участки сужения и расширения, которые позволят в полной мере исследовать результат взаимодействия участников.

Для того, чтобы обеспечить между участниками некоторое расстояние в начальный момент времени, запустим их с промежутками в $\Delta t = 0.2$.

Предположим, что в некоторый момент времени t_{st} из вершины с номером 0 выезжает участник в вершину с номером 42. Рассмотрим две модели движения АТС: модель следования за лидером (см. рис. 3) и макроскопическую модель (1). Исследуем возможность применения модифицированного алгоритма Дейкстры к задаче оптимизации маршрута путем сравнения с обычным перебором (см. таблицы 1, 2, 3, 4.).

Таблица 1: Результаты запуска программы поиска оптимального пути в микроскопической модели следования за лидером при условии влияния добавленного участника на движение остальных в зависимости от параметров v_{max} , v_{min} , a , где $a = \frac{v_{max}^2 - v_{min}^2}{2(D-l)}$.

	t_{st}	v_{max}	v_{min}	a	Перебор			Мод. Дейкстра			$\min_p T(p)$	$\max_p T(p)$
					$T(p^*)$	Кол-во итераций моделирования	Время работы, с	$T(p^*)$	Кол-во итераций моделирования	Время работы, с		
1	5	60	10	4.375	1129.16	64	16.7008	1129.16	52	12.3305	1129.16	2018.37
2	5	60	20	4	1114.88	64	27.197	1114.88	52	30.4414	1114.88	1967.52
3	5	80	10	7.875	850.046	64	19.2823	850.046	52	12.4218	850.046	1471.65
4	5	80	20	7.5	841.158	64	14.2945	841.158	52	15.3071	841.158	1470.66
5	40	60	10	4.375	1273.21	64	23.071	1273.21	52	12.226	1273.21	2151.37
6	40	60	20	4	1231.54	64	27.1132	1231.54	52	34.6609	1231.54	2101.94
7	40	80	10	7.875	971.713	64	19.0107	971.713	51	13.8246	971.713	1623.74
8	40	80	20	7.5	945.739	64	28.5323	945.739	52	34.8503	945.739	1595.29

Таблица 2: Результаты запуска программы поиска оптимального пути в микроскопической модели следования за лидером при условии отсутствия влияния добавленного участника на движение остальных в зависимости от параметров v_{max} , v_{min} , a , где $a = \frac{v_{max}^2 - v_{min}^2}{2(D-l)}$.

	t_{st}	v_{max}	v_{min}	a	Перебор			Мод. Дейкстра			$\min_p T(p)$	$\max_p T(p)$
					$T(p^*)$	Кол-во итераций моделирования	Время работы, с	$T(p^*)$	Кол-во итераций моделирования	Время работы, с		
1	5	60	10	4.375	1142.26	64	27.0787	1142.26	52	39.7554	1142.26	2018.37
2	5	60	20	4	1114.88	64	33.0677	1114.88	52	23.0935	1114.88	2014.19
3	5	80	10	7.875	850.046	64	30.5048	850.046	52	17.3063	850.046	1471.65
4	5	80	20	7.5	841.158	64	29.033	841.158	52	18.1372	841.158	1479.91
5	40	60	10	4.375	1273.21	64	28.2584	1273.21	52	14.2799	1273.21	2151.37
6	40	60	20	4	1231.54	64	36.577	1231.54	52	20.4323	1231.54	2101.94
7	40	80	10	7.875	971.713	64	33.1075	971.713	51	14.571	971.713	1623.74
8	40	80	20	7.5	945.739	64	25.1396	945.739	52	20.1858	945.739	1595.29

Таблица 3: Результаты запуска программы поиска оптимального пути в макроскопической модели при условии влияния добавленного участника на движение остальных в зависимости от параметров v_{max} , v_{min} , a , где $a = \frac{v_{max}^2 - v_{min}^2}{2(D-l)}$.

	t_{st}	v_{max}	v_{min}	a	Перебор			Мод. Дейкстра			$\min_p T(p)$	$\max_p T(p)$
					$T(p^*)$	Кол-во итераций моделирования	Время работы, с	$T(p^*)$	Кол-во итераций моделирования	Время работы, с		
1	5	60	10	4.375	1208.42	64	0.38878	1208.42	51	0.308759	1208.42	2229.88
2	5	60	20	4	1164.72	64	0.390943	1164.72	51	0.308681	1164.72	2093.01
3	5	80	10	7.875	914.833	64	0.389688	914.833	51	0.30841	914.833	1702.06
4	5	80	20	7.5	891.223	64	0.390798	891.223	51	0.307565	891.223	1617.92
5	40	60	10	4.375	1220.1	64	0.389539	1220.1	51	0.308927	1220.1	2230.95
6	40	60	20	4	1177.82	64	0.390696	1177.82	51	0.307389	1177.82	2096.77
7	40	80	10	7.875	926.374	64	0.394311	926.374	51	0.309656	926.374	1702.62
8	40	80	20	7.5	902.47	64	0.392885	902.47	51	0.308543	902.47	1620.23

Таблица 4: Результаты запуска программы поиска оптимального пути в макроскопической модели при условии отсутствия влияния добавленного участника на движение остальных в зависимости от параметров v_{max} , v_{min} , a , где $a = \frac{v_{max}^2 - v_{min}^2}{2(D-l)}$.

	t_{st}	v_{max}	v_{min}	a	Перебор			Мод. Дейкстра			$\min_p T(p)$	$\max_p T(p)$
					$T(p^*)$	Кол-во итераций моделирования	Время работы, с	$T(p^*)$	Кол-во итераций моделирования	Время работы, с		
1	5	60	10	4.375	1202.6	64	0.409948	1202.6	51	0.309044	1202.6	2210.82
2	5	60	20	4	1161.61	64	0.417604	1161.61	51	0.309546	1161.61	2082.44
3	5	80	10	7.875	910.634	64	0.389438	910.634	51	0.311105	910.634	1685.62
4	5	80	20	7.5	888.15	64	0.390584	888.15	51	0.310151	888.15	1607.24
5	40	60	10	4.375	1214.21	64	0.39104	1214.21	51	0.30969	1214.21	2213.23
6	40	60	20	4	1172.99	64	0.412499	1172.99	51	0.310533	1172.99	2086.97
7	40	80	10	7.875	921.952	64	0.418656	921.952	51	0.312077	921.952	1687.66
8	40	80	20	7.5	898.244	64	0.412369	898.244	51	0.312483	898.244	1610.64

На основе полученных результатов можно сделать следующие выводы:

- Для обеих моделей модифицированный алгоритм Дейкстры позволяет получить оптимальный путь независимо от влияния нового участника на движения остальных, даже при условии, что неравенство прохождения ребер (3) может не выполняться.
- Макроскопическая модель требует меньших вычислительных ресурсов для моделирования по сравнению с микроскопической моделью.
- На практике оказалось, что алгоритм Дейкстры до момента посещения точки назначения успел промоделировать передвижение по всем ребрам, быть может кроме одного.

- Микроскопическая модель более приближена к естественному движению автомобилей, и на практике, как мы и предполагали при постановке задачи, влияние нового участника на другие АТС оказалось ничтожно мало.

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.