МЕТОДИКА РЕШЕНИЯ ЗАДАЧИ ОПТИМИЗАЦИИ МАРШРУТА В ДОРОЖНЫХ СЕТЯХ

ДЕРЕВЯНЧУК Екатерина Дмитриевна
кандидат физико-математических наук, доцент
АРТЕМОВ Максим Викторович
студент
Пензенский государственный университет
г. Пенза, Россия

Данная статья посвящена разработке методологии поиска оптимальных путей в дорожной сети города с учётом пробок с использованием теории графов. В работе предложены две модификации алгоритма Дейкстры, которые позволяют учитывать информацию о дорожных пробках и тупиковых ветвях. В результате применения предлагаемого подхода ожидается улучшение ситуации на дорогах, сокращение времени в пути и повышение эффективности городского транспортного потока, а также снижение негативного воздействия на окружающую среду.

Ключевые слова: алгоритм Дейкстры, модификация, дорожные пробки, тупиковая ветвь, кратчайший маршрут.

В современном мире управление дорожным движением становится одной из ключевых задач в городском планировании. Проблемы перегруженности дорожных сетей и постоянные заторы оказывают серьезное воздействие на жизнь городских жителей и экономическое развитие регионов. Одним из важных инструментов в решении подобных проблем является применение теории графов для оптимизации путей движения [1-3].

В рамках данной статьи рассматривается проблема организации дорожного движения в городе. Например, город Тольятти сталкивается с серьезной проблемой перегрузки дорожной сети, особенно при въезде в город со стороны Жигулевска, где формируются длительные пробки, простирающиеся на несколько километров вплоть до конца дамбы на Куйбышевском водохранилище на Волге. Эта ситуация обусловлена значительным потоком грузового транспорта, который возникает на железнодорожном переезде в данном районе. Стоит отметить, что ситуация с перегруженностью дорожной инфраструктуры не только приводит к временным и финансовым потерям для водителей, но и сказывается на окружающей среде в виде повышенных выбросов от транспортных средств и ухудшения экологической обстановки в городе.

Существует потенциал для оптимизации движения путем нахождения альтернативных маршрутов, позволяющих водителям избежать стояния в заторе. Допустим, у нас есть граф G, представляющий собой набор вершин, некоторые из которых соединены ребрами, а некоторые не соединены. Например, эти вершины могут быть перекрестками, а рёбра – улицами, каждая из которых имеет свою длину. Возникает вопрос – каким образом можно выбрать кратчайший маршрут из одной точки дорожной сети в другую, учитывая при этом возможные пробки на дорогах, которые могут появляться или исчезать со временем.

<u>Постановка задачи А.</u> **Требуется найти оптимальный маршрут из вершины x_i в вершину x_j, учитывая информацию о пробках на дороге.**

Первоначально наиболее эффективный метод решения задачи о кратчайшем пути был предложен Эдсгером Вибе Дейкстрой, который относится к численно-аналитическим методам.

---

[1]**Эдсгер Вибе Дейкстра** – выдающийся голландский учёный в области информатики, который руководил созданием операционной системы THE «Technische Hogeschool Eindhoven» с поддержкой многозадачности, создатель алгоритма Дейкстры и семафоров, один из основателей структурного программирования.

*Рисунок 1.***Эдсгер Вибе Дейкстра (11.05.1930- 06.08.2002) и Роттердам – город, в котором он родился в 1930 году в семье учёных: отце был химиком, мать – математиком.**

Суть его метода заключается в присваивании временных пометок вершинам. Каждая пометка вершины определяет верхнюю границу длины пути от некоторой начальной вершины x_i до рассматриваемой x_k. После каждой итерации значения пометок уменьшаются, пока одна из них не станет постоянной, указывая на точную длину кратчайшего пути от вершины x_i до вершины x_k. Однако у алгоритма Дейкстры, как и у других методов, есть свои недостатки.

Во-первых, его сложность. Сложность алгоритма оценивается как $O(n^2)$, где n – количество вершин графа. Для произвольного графа с n вершинами это означает, что количество операций пропорционально квадрату количества вершин. Например, при n=1000, количество операций составит 1000^2, что равно миллиону операций. Решение таких задач может потребовать использования суперкомпьютеров.

Во-вторых, алгоритм Дейкстры применим только для поиска пути в статических графах, где вес каждого ребра не меняется со временем. Если граф динамический и веса ребер изменяются со временем, например, при движении по городским улицам, то необходимо пересчитывать путь при каждом изменении весов. В связи с этим в данной работе рассматриваются модификации алгоритма Дейкстры для решения поставленных задач.

Первая модификация алгоритма Дейкстры. Численно-аналитический метод решения задачи основывается на модификации алгоритма Дейкстры.

Пусть дан граф G (X, A) с конечным числом вершин и дуг, где X = $\{x_i\}$, i=1, 2, 3, ..., n – множество вершин, A = $\{a_i\}$, i=1, 2, 3, ..., m – множество дуг (рисунок 2).

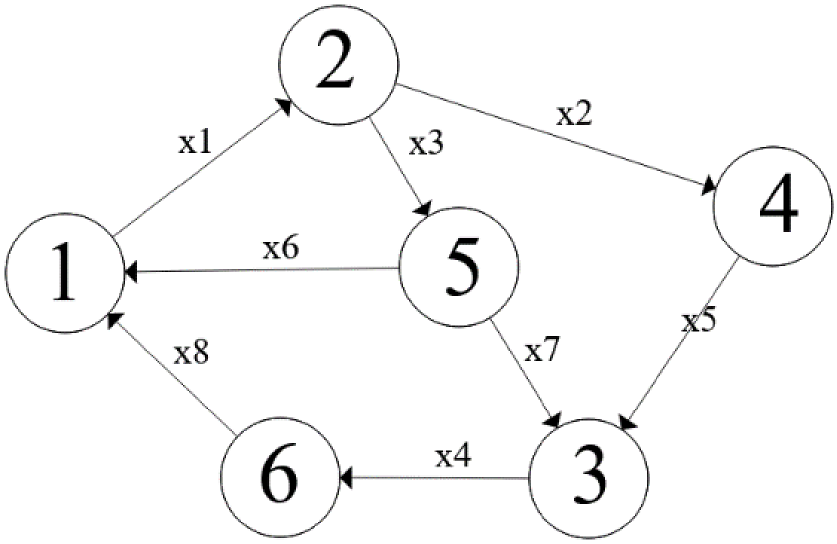

Рисунок 2. **Граф со взвешенными ребрами**

Известна матрица весов графа

$$C = \begin{pmatrix} c_{11} & \cdots & c_{11} \\ \cdots & \cdots & \cdots \\ c_{m1} & \cdots & c_{11} \end{pmatrix},$$

Где C_{ij} – это длина от вершины x_i до x_j

Также известна матрица коэффициентов пробок:

$$P = \begin{pmatrix} p_{11} & \cdots & p_{11} \\ \cdots & \cdots & \cdots \\ p_{m1} & \cdots & p_{11} \end{pmatrix},$$

где p_{ij}– это бал пробки на участке пути от вершины x_i до x_j

Для расчета времени прохождения маршрута можно использовать следующую формулу:

$$\text{Время(в ч)} = \frac{\text{Расстояние(в км)}}{\text{Скорость движения(в км/ч)}} \times \text{Коэффициент пробки}$$

Предположим, что автомобилю нужно проехать путь равный трём километрам в условиях пробок от 1 до 10 баллов, также установим, что средняя скорость движения в регионах 40 км/ч, а в городах-миллиониках 30 км/ч. Используя эти данные и формулу, получим третью матрицу весов, где учитывается балл пробок:

$$F = \begin{pmatrix} f_{11} & \cdots & f_{11} \\ \cdots & \cdots & \cdots \\ f_{m1} & \cdots & f_{11} \end{pmatrix}.$$

где f_{ij}– это время на участке пути от вершины x_i до x_j с учётом балла пробки p_{ij}.

В результате получим две таблицы соот ветствия баллов пробки расстоянию и времени: для регионов (таблица 1) и для крупных городов (таблица 2).

Таблица 1

СООТВЕТСВИЕ БАЛЛОВ ПРОБКИ РАССТОЯНИЮ И ВРЕМЕНИ ДЛЯ РЕГИОНОВ

Балл пробки	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Расстояние, км	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
Время, мин	**4.5**	**9**	**13.5**	**18**	**22.5**	**27**	**31.5**	**36**	**40.5**	**45**

Таблица 2

СООТВЕТСВИЕ БАЛЛОВ ПРОБКИ РАССТОЯНИЮ И ВРЕМЕНИ ДЛЯ КРУПНЫХ ГОРОДОВ

Балл пробки	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Расстояние, км	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
Время, мин	**6**	**12**	**18**	**24**	**30**	**36**	**42**	**48**	**54**	**60**

Выбранная формула основана на простом математическом принципе: время в пробке пропорционально длине маршрута и баллу пробки, который отражает степень затруднения движения.

При этом предполагается, что скорость движения автомобилей в пробке значительно снижается по сравнению со свободным движением. Поэтому мы используем обратно пропорциональную зависимость между скоростью движения и временем в пробке: чем меньше скорость, тем больше времени требуется для прохождения того же расстояния.

Вторая модификация алгоритма Дейкстры. Пусть требуется найти кратчайший путь между парой вершин для графа G на рисунке 3. В качестве численного метода выберем алгоритм Дейкстры.

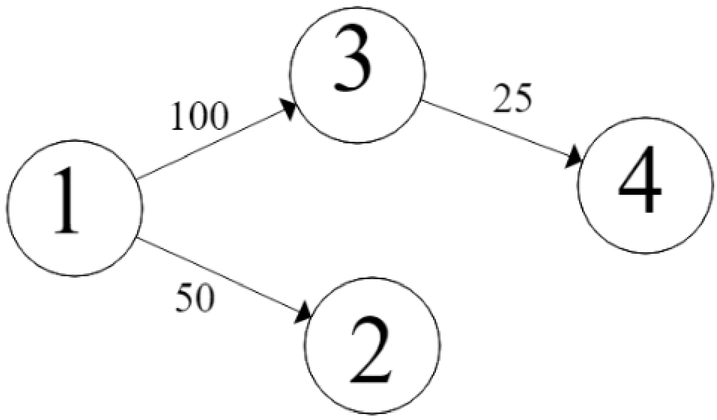

Рисунок 3. **Исходные данные задачи**

1. Устанавливаем расстояние D[i] от первой вершины до i-ой вершины. Считается, что изначально расстояние равно ∞, а D[1] – расстояние от первой вершины до самой себя равно 0 (таблица 3).

Таблица 3

ИСХОДНЫЕ ДАННЫЕ

i	**1**	**2**	**3**	**4**
D[i]	**0**	∞	∞	∞

Запускаем цикл из *n* итераций (по числу вершин). Рассматриваем отображение первого порядка из вершины 1 в другие вершины. Это будут вершины 2 и 3. Отмечаем расстояние от вершины 1 до вершин 1-3 (таблица 4). Выбираем наименьшее расстояние от вершины 1. Это будет отображение во вторую вершину. И снова повторяем итерацию, но уже для вершины 2.

Таблица 4

НУЛЕВАЯ ИТЕРАЦИИ

i	**1**	**2**	**3**	**4**
D[i]	**0**	**0+50=50**	**0+100=100**	∞

Итерация № 1. Рассматриваем отображение первого порядка вершины 2. Его нет. Тогда рассматриваем отображения вершины 3 на оставшиеся вершины (таблица 5 и рисунок 4).

Таблица 5

ПЕРВАЯ ИТЕРАЦИИ

i	**1**	**2**	**3**	**4**
D[i]	**0**	**50**	**100**	∞

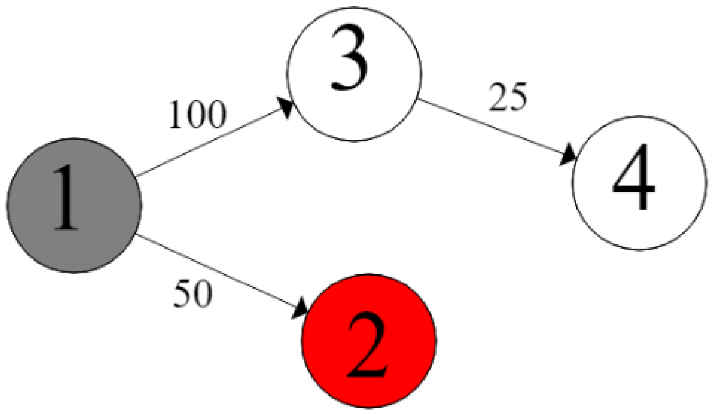

Рисунок 4. **Результат первой итерации**

Итерация № 2. На второй итерации определяем, что отображением первого порядка вершины 3 будет вершина 4. Рассчитываем расстояние от вершины 1 до вершины 4 через вершину 3, оно будет равно 125 (таблица 5 и рисунок 5).

Таблица 5

ВТОРАЯ ИТЕРАЦИИ

n	1	2	3	4
D	0	50	100+25=125	∞

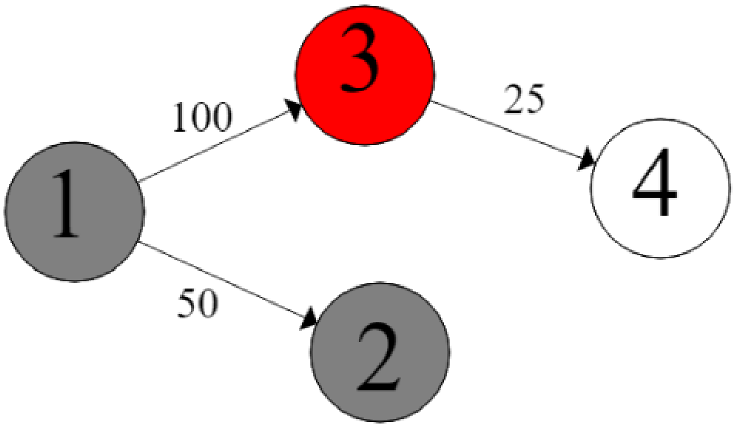

Рисунок 5. **Результат второй итерации**

Из рисунка 4 видно, что кратчайший путь из вершины 1 в вершину 4: 1→3→4; длина пути составит 125 (рисунок 6).

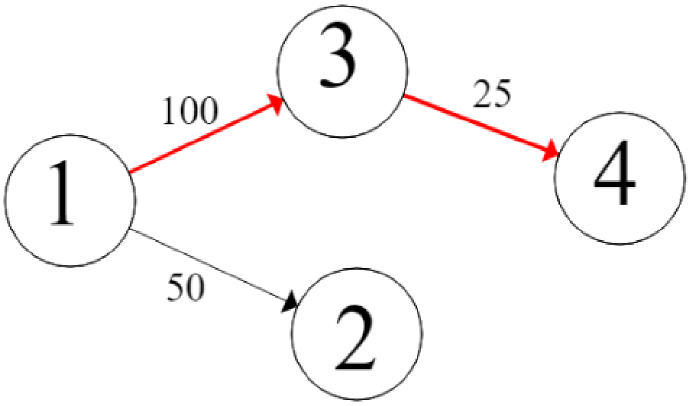

Рисунок 6. **Кратчайший путь из вершины 1 в вершину 4**

Следует отметить, что данный простой пример был подобран специально, чтобы проиллюстрировать понятие «тупиковой» ветви и её исключение из рассмотрения при выборе кратчайшего пути.

Вывод. Анализируя каждый шаг решения данной задачи, приходим к выводу, что время расчёта задачи с помощью алгоритма Дейкстры зависит от нумерации вершин. Так, например, на первой итерации рассчитывается расстояние до второй вершины, но для расчета расстояний от первой вершины до четвертой – данный расчет не играет роли, так как нет пути из вершины 1 в вершину 4 через вершину 2. Поэтому для увеличения скорости расчета можно было эту итерацию исключить, и решение было бы получено на 1 итерацию быстрее. Это можно было бы сделать, например, с помощью редактирования нумерации вершин. Поэтому, если граф содержит «тупиковую ветвь», как например, в рассмотренной задаче – ветвь 1-2 (т.е. путь из вершины 1 в вершину 2), то следует сначала исключить ее из графа, а именно с помощью удаления вершины, где находится тупик. С точки зрения теории графов, это будет означать, что строка в матрице смежности исходного графа с номером вершины, в которой «тупик», будет нулевой. Тогда методом исключения тупиковых ветвей из задачи (т.е. всех нулевых строк из матрицы смежности и с таким же индексом всех соответствующих столбцов) достигаем оптимизации решения задачи.

ЛИТЕРАТУРА

1. *Деревянчук Е.Д.* Нахождение кратчайшего пути с помощью уточнения весовой матрицы в алгоритме Дейкстры // Общество. – 2024. – № 1(32). Часть 2. – С. 17-19.
2. *Изотова Т.Ю.* Обзор алгоритмов поиска кратчайшего пути в графе // Новые информационные технологии в автоматизированных системах. – 2016. – № 19. – С. 341-344.
3. *Деревянчук Е.Д.* Влияние планировки города и карты дорожной сети на выбор оптимального маршрута // Общество. – 2024. – № 1(32). Часть 2. – С. 10-17.

THE METHODOLOGY OF SOLVING THE ROUTE OPTIMIZATION PROBLEM IN ROAD NETWORKS

DEREVYANCHUK Ekaterina Dmitrievna
Candidate of Science in Physics and Mathematics, Associate Professor
ARTYOMOV Maxim Viktorovich
Student, Penza State University
Penza, Russia

This article is devoted to the development of a methodology for finding optimal paths in the city's road network, taking into account traffic jams using graph theory. It is proposed two Dijkstra's algorithm modifications of, which allows taking into account information about traffic jams and dead-end branches. As a result of the proposed approach, it is expected to improve the situation on the roads, reduce travel time and increase the efficiency of urban traffic flow, as well as reduce the negative impact on the environment.

Keywords: Dijkstra's algorithm, modification, traffic jams, dead-end branch, shortest route.