ISSN 2225-9309 (Print)
2026. № 1 (35). С. 85–88
Молодежный
Вестник УГАТУ
http://journal.uust.ru

УДК 519.6
doi 10.54708/22259309_2026_13513

Современные подходы к решению задачи поиска кратчайшего пути в графах

В. О. Юрьев[1], О. Э. Вафин[2]

[1]mainerdest@list.ru, [2]oskarvafin@gmail.com

[1, 2]ФГБОУ ВО «Уфимский университет науки и технологий» (УУНиТ)

Аннотация. Определение кратчайшего пути между двумя точками в графе является одной из ключевых проблем в теории графов. В работе сравниваются несколько алгоритмов, а именно – Форда–Беллмана, Флойда–Уоршелла и Дейкстры. Эффективность данных методов определяется в зависимости от поставленной задачи.

Ключевые слова: алгоритмы Дейкстры, Форда–Беллмана, Флойда–Уоршелла; задачи поиска кратчайших путей.

ВВЕДЕНИЕ

В данной работе сравниваются различные алгоритмы поиска кратчайших путей в графах, такие как: Дейкстра, Форд–Беллман и Флойд–Уоршелл.

Целью является поиск кратчайшего пути в графе между двумя парами точек. При этом веса у ребер могут быть отрицательными и положительными. С разной структурой или количеством вершин.

Каждый из представленных алгоритмов имеет недостатки и достоинства. Анализ позволит определить направления, в которых каждый из алгоритмов наиболее эффективен и универсален.

АЛГОРИТМ ФЛОЙДА–УОРШЕЛЛА

В алгоритме Флойда–Уоршелла создаётся квадратная матрица размером V на V, где V – это количество вершин в графе. В ячейках этой матрицы будут храниться значения расстояний между парами вершин. На диагонали, то есть в ячейках, где указана одинаковая вершина – записывается ноль. Если между двумя вершинами есть ребро – указывается его вес. Если нет, то бесконечность (∞).

Он рассматривает её как возможную промежуточную точку на пути между другими вершинами. Для каждой перебираем пары оставшихся сочетаний. При прохождении через промежуточную вершину путь должен быть меньше двух других вершин.

В матрице есть отрицательные циклы, если на главной диагонали существуют отрицательные значения. Для восстановления длины путей можно создать дополнительную таблицу родителей.

Алгоритм работает с ориентированными графами, в том числе с отрицательными весами. Но с ними он неэффективен.

АЛГОРИТМ ФОРДА–БЕЛЛМАНА

В алгоритме Форда–Беллмана на первом шаге создаётся массив $R[i]$ расстояниями от начальной вершины до остальных.

В ребрах вместо бесконечности (∞) указывается вес ребра между текущими вершинами. Если же расстояние до вершины уже определено (то есть $R[m]$ не бесконечность), и если сумма этого пути и веса ребра $w(m,n)$ меньше текущего значения $R[n]$, то это значит, что мы нашли более короткий путь до вершины n. Тогда мы обновляем значение $R[n]$, присваивая ему это меньшее значение – $R[m] + w(m,n)$.

Третий шаг – повторение предыдущего шага (проверка и обновление расстояний) столько раз, сколько вершин в графе минус один, то есть $|v|$ – 1 раз. Это нужно потому, что при каждом проходе обновляются расстояния, и за $|v|$ – 1 проход можно гарантированно найти все кратчайшие пути.

Если в какой-то момент ни одно значение $R[i]$ не изменилось, значит, все кратчайшие пути уже найдены либо, возможно, до некоторых вершин в данной конкретной ситуации нельзя добраться.

На четвёртом шаге алгоритм проверяет, есть ли в графе отрицательные циклы – это такие замкнутые маршруты, в которых значение пути является отрицательным. Для проверки мы ещё раз идём по рёбрам: если найдено ребро (u, v), для которого выполняется неравенство $R[u] + w(u,v) < R[v]$, следовательно, значение пути может измениться в меньшую сторону, что является осуществимым лишь при существовании отрицательного цикла. В таком случае найти кратчайший путь не представляется возможным, потому что цикл позволяет бесконечно уменьшать сумму. Если же таких рёбер не осталось, значит, кратчайшие пути найдены правильно, и в массиве $R[i]$ лежат верные значения.

Данный алгоритм, созданный Ричардом Беллманом и Лестером Фордом, обладает важным свойством – он способен выявлять отрицательные циклы.

АЛГОРИТМ ДЕЙКСТРЫ

Алгоритм Дейкстры не умеет работать с отрицательными весами, а также не способен находить отрицательные циклы.

Определяем стартовую вершину и создаем таблицу с расстояниями от нее до вершин. Расстояние до начальной вершины равно нулю, а для других – бесконечность.

Создаем очередь, в которую помещаем вершины и расстояние. Сравниваем полученное расстояние с тем, что уже записано в таблице. Если оно оказалось больше, переходим к следующей.

Если же расстояние совпадает с тем, что в таблице, или оказывается меньше, мы продолжаем. Рассматриваем все вершины, с которыми текущая вершина напрямую связана, – это её соседи в графе. Для каждого такого соседа вычисляем возможную длину пути от стартовой вершины, проходя через текущую. Иначе говоря, складываем расстояние до текущей и длину ребра, соединяющего её с соседом.

Если новый путь до соседней вершины короче уже известного, то обновляем значение в таблице – записываем новое, меньшее расстояние.

Соседа с новым расстоянием добавляем в очередь и анализируем его связи. Повторяем, пока очередь не опустеет.

СРАВНЕНИЕ И АНАЛИЗ АЛГОРИТМОВ

Учитывая структуру, наличие отрицательных ребер и их поиск, сложность, эффективность при больших значениях и универсальность, мы составили сравнительную таблицу данных алгоритмов (табл. 1).

Таблица 1

Сравнительная таблица алгоритмов поиска кратчайших путей

Критерий	*Название алгоритма*		
	Дейкстра	**Форда–Беллмана**	**Флойда–Уоршелл**
Тип задачи	От определенной вершины ко всем	От определенной вершины ко всем	Расстояния между каждой из вершин

Структура	Граф, не имеющий отрицательных ребер и являющийся разреженным	Граф, который может иметь или не иметь отрицательные ребра	Загруженный граф
Допуск отрицательных рёбер	Отсутствует	Допускаются	Может допускаться, но при условии отсутствия отрицательных циклов
Поиск отрицательных циклов	Нет	Да	Нет
Сложность (в худшем случае)	$O(v^2)$ или $O((v+E)\,log\,v)$ при использовании кучи	$O(V \cdot E)$	$O(v^3)$
Тип графа	С разреженными графами	С любыми графами	С плотным графами
Принцип работы	Жадный алгоритм	Динамическое программирование	Динамическое программирование
Промежуточные вершины	Неявно	Неявно	Явно
Эффективность при больших значениях	Если в графе нет отрицательных ребер, то большая	Универсален, но слабее Дейкстры, поэтому средняя	Может использоваться со средними и малыми графами, низкая
Универсальность	При положительных ребрах низкая	Позволяет взаимодействовать с любыми весами и проверять циклы – высокая	Нет защиты от циклов, в связи с этим средняя

ЗАКЛЮЧЕНИЕ

Алгоритм необходимо выбирать в зависимости от критериев. К ним относятся: структура, наличие отрицательных ребер, сложность и эффективность. В ходе исследования составлена сравнительная таблица, а также проанализированы все три алгоритма.

СПИСОК ЛИТЕРАТУРЫ

1. **Левитин, А. В.** Алгоритмы: введение в разработку и анализ. – М.: Вильямс, 2006. – С. 189-195.
2. **Шварц, Р**. Дискретная математика для программистов. – М.: «ДМК Пресс», 2009.
3. **Руднев, В. Ю.** Прикладная математика. – М., 2010.
4. **Томас, Х. К.** Алгоритмы: построение и анализ. – М.: МЦНМО, 2004. – 523 с.
5. **Листопад, Н. И. Воротницкий, Ю. И., Хайдер, А. А.** Оптимальная маршрутизация в мультисервисных сетях телекоммуникаций на основе модифицированного алгоритма Дейкстры // Вестник БГУ. Серия 1. – 2015. – № 1. – С. 70-76.
6. **Коган, Е. С.** Модификация алгоритма Дейкстры для нахождения кратчайшего пути в ориентированном графе. – М.: Забайкальский государственный университет, 2022. – С. 43-45.
7. **Коган, Е. С.** Графы. Алгоритм нахождения кратчайшего пути. – М.: МЦНС «Наука и Просвещение», 2018. – С. 49-51.
8. **Соколова, А. А**. Алгоритмы на графах. – М.: Сибирский институт бизнеса, управления и психологии, 2017. – С. 318-321.
9. **Перец, Р. В.** Анализ алгоритмов задачи поиска оптимальных путей. – М.: «Наука и Просвещение (ИП Гуляев Г.Ю.)», 2021. – С. 11-14.
10. **Маткурбанов, Т. А.** Анализ алгоритмов задачи поиска оптимальных путей. – СПб. .: Санкт-Петербургский государственный лесотехнический университет им. С.М. Кирова, 2024. – С. 106-112.
11. **Смелов, В. В.** Алгоритмы на графах и их реализации на С++ : учебно-методическое пособие / В. В. Смелов, Л. С. Мороз. – Минск : БГТУ, 2017. – 132 с.
12. **Иванова, Е. О.** Графы. Алгоритм нахождения кратчайшего пути. – М.: МЦНС «Наука и Просвещение», 2018. – С. 49-51.
13. **Воронин, В. В., Бахрушина, Г. И.** Поиск кратчайшего пути во взвешенном орграфе с помощью алгоритма Форда-Беллмана. – Х.: Тихоокеанский государственный университет, 2021. – С. 53-55.

14. **Фатыхов, И. Д., Кашапов, Т. И., Нуртдинов, Р. С.** Алгоритм Форда–Беллмана поиска кратчайшего пути в графе с отрицательной стоимостью. – М.: ЗАО «Университетская книга», 2024. – С. 340-343.

15. **Бойченко, П. В., Деменкова, Е. А.** Анализ существующих алгоритмов решения задачи нахождения кратчайшего пути. – Тамбов.: Тамбовский государственный технический университет, 2016. – С. 147-151.

16. **Балапанова, Г. Б.** Задача о кратчайшем пути на графе / Г. Б. Балапанова // Инновационные подходы в современной науке : сб. ст. по материалам XVIII Междунар. науч.-практ. конф. – Оренбург : ИП Шелистов Д. А. (Издательский центр «Quantum»), 2018. – С. 405–408.

17. **Пронникова, Т. Ю.** Алгоритмы теории графов / Т. Ю. Пронникова // Инновационное развитие: потенциал науки и современное образование : сб. ст. по материалам Междунар. науч.-практ. конф. – Уфа : ИП Зоркин В. А., 2021. – С. 1536–1542.

18. **Поречный, С. С., Житникова, Н. И., Шерыхалина, Н. М.**, **Ураков, А. Р.** Дискретная математика / С. С. Поречный, Н. И. Житникова, Н. М. Шерыхалина, А. Р. Ураков. – Уфа: РИК УГАТУ, 2019. – 400 с.

ОБ АВТОРАХ

ЮРЬЕВ Владислав Олегович, студент ПРО ИИМРТ УУНиТ;
ВАФИН Оскар Эдуардович, студент ПРО ИИМРТ УУНиТ.

METADATA

Title: Modern approaches to solving the shortest path problem in graphs.

Authors: V. O. Yurev[1], O. E. Vafin [2]

Affiliation:

[1, 2] Ufa University of Science and Technology (UUST), Russia.

Email: [1]mainerdest@list.ru, [2]oskarvafin@gmail.com

Language: Russian.

Source: Molodezhnyj Vestnik UGATU (scientific journal of Ufa University of Science and Technology), no. 1 (35), pp. 85-88, 2026. ISSN 2225-9309 (Print).

Abstract: Determining the shortest path between two points in a graph is one of the key problems in graph theory. The paper compares several algorithms, namely Bellman-Ford, Floyd-Warshall and Dijkstra. These approaches are widely used in various fields such as cartography and logistics. As a result, the choice of the best algorithm is determined by the specifics of the task and the conditions of its implementation.

Key words: shortest path search, Dijkstra algorithm, Bellman-Ford algorithm, Floyd-Warshall algorithm.

About authors:

YUREV Vladislav Olegovich, student PRO IIMRT UUST;

VAFIN Oskar Eduardovich, student PRO IIMRT UUST.