УДК 519.17

ИЗУЧЕНИЕ ЗАВИСИМОСТИ ВРЕМЕНИ РАБОТЫ АЛГОРИТМА ДЕЙКСТРЫ ОТ РАЗМЕРНОСТИ И ПЛОТНОСТИ ГРАФА

Никуленко М.А.

студент,

Российский университет транспорта (МИИТ),

Москва, Россия

Иванова А.П.

к.ф.-м.н., доцент

Российский технологический университет – МИРЭА,

Российский университет транспорта (МИИТ),

Москва, Россия

Аннотация:

В статье проведено исследование алгоритма Дейкстры нахождения кратчайших расстояний для неориентированных графов. Рассмотрены графы разной размерности и разной плотности. Была написана программа на языке С. В результате исследования на случайных графах найдена зависимость времени выполнения алгоритма Дейкстры от количества вершин и коэффициента плотности неориентированного графа.

Ключевые слова: граф, матрица кратчайших расстояний, алгоритм Дейкстры.

STUDYING THE DEPENDENCE OF THE RUNNING TIME OF THE DIJKSTRA ALGORITHM ON THE DIMENSION AND DENSITY OF THE GRAPH

Nikulenko M.A.

student,

Russian University of Transport (MIIT),

Moscow, Russia

Ivanova A.P.
Ph.D., Associate Professor
MIREA – Russian Technological University,
Russian University of Transport (MIIT),
Moscow, Russia

Abstract: The article examines Dijkstra's algorithm for finding shortest distances for undirected graphs. Graphs of different dimensions and densities are considered. The program was written in C. As a result of the study on random graphs, the dependence of the execution time of Dijkstra's algorithm on the number of vertices and the density coefficient of an undirected graph was found.
Keywords: graph, shortest distance matrix, Dijkstra algorithm,

Изучение алгоритма Дейкстры и его применения к неориентированным графам является ключевым аспектом современной теории графов [1-3,6,9,10]. Этот алгоритм широко используется для решения задач оптимизации в различных областях, таких как логистика, телекоммуникации, маршрутизация и анализ социальных сетей. Наполненность (плотность) графа, определяемая количеством рёбер относительно количества вершин (обозначим ρ), существенно влияет на эффективность и сложность выполнения алгоритма. Анализ влияния этой характеристики актуален, поскольку в реальных приложениях графы варьируются от сильно разреженных (например, дорожные сети) до почти полных (социальные сети или сети взаимодействия).

Целью данной работы является нахождение зависимости времени выполнения алгоритма Дейкстры от количества вершин и коэффициента наполненности неориентированного графа.

Пусть задан граф $G = (V, E)$ где V – множество вершин, E – множество рёбер. Граф G – взвешенный, С = $\left(c_{ij}\right)_{n\times n}$ – матрица весов, $n = |V|$.

Будем предполагать, что:

1) все веса рёбер неотрицательные: $\forall c_{ij} \geq 0$,
2) в графе нет циклов с суммарным отрицательным весом (если это не так, то обойдя по такому циклу достаточно много раз, получим путь со сколь угодно малым весом, стремящемся к $-\infty$),
3) матрица весов, вообще говоря, не удовлетворяет неравенству треугольника ($c_{ij} \leq c_{ik} + c_{kj}$),
4) если в графе нет дуги (v_i, v_j), то её вес полагаем равным ∞ ($c_{ij} = \infty$).

Приведём краткое описание алгоритма Дейкстры [1,5,10]. Каждой вершине присваивается вес-метка, отражающая минимальное расстояние от исходной вершины до данной. Эта метка может быть временной или постоянной. Временная метка может быть обновлена, если будет найден более короткий путь. Если же обновление не происходит, метка становится постоянной, и её значение указывает на длину кратчайшего пути от начальной вершины до рассматриваемой. На каждом шаге алгоритма ровно одна из временных меток становится постоянной.

Будем искать кратчайшие пути из вершины s во все остальные.

Обозначим $l(v_i)$ метку вершины v_i.

Присвоение начальных значений

Шаг 1. Присвоить $l(s) = 0$ и считать эту метку постоянной. Присвоить $l(v_i) = \infty \ \ \forall v_i \neq s$ – временные метки. Присвоить $p = s$ – текущая вершина.

Обновление меток

Шаг 2. $\forall v_i \in \Gamma(p)$, метки которых временные, изменить метки:

$$l(v_i) = \min\{l(v_i), l(p) + c(p, v_i)\},$$

где $c(p, v_i)$ – вес ребра, соединяющего текущую вершину p с вершиной v_i.

Превращение меток в постоянные

Шаг 3. Среди всех вершин с временными метками найти вершину v_i^*:

$$l(v_i^*) = \min\left\{l(v_i)\right\}.$$

Шаг 4. Считать метку вершины v_i^* постоянной, обновить текущую вершину: $p = v_i^*$.

Шаг 5. Если все вершины имеют постоянные метки, то эти метки равны длинам кратчайших путей, иначе переход на шаг 2.

Будем генерировать графы с разным количеством вершин V (100, 200, …, 1000) и с различными коэффициентами наполненности ρ (0.1, 0.2, …, 1.0). При этом для генерации ребра генерируется случайное число от 0 до 10000, после это число делится на 10000. Если коэффициент наполненности ρ будет больше этого числа, то ребро будет добавлено в граф. После этого всем рёбрам графа приписывается вес, равный псевдослучайному значению из интервала [1, 50).

Далее для каждой комбинации V и ρ запускаем алгоритм Дейкстры для фиксированной начальной вершины (s=1). Замер времени выполнения производится с использованием стандартных средств языка С (функции clock() из <time.h>). Результаты записываем в файл.

На основе собранных данных строятся 10 графиков зависимости времени выполнения алгоритма Дейкстры от числа вершин V, при фиксированных значениях коэффициента наполненности ρ=0.1, 0.2, …, 1.0 (см. рис. 1-10). Экспериментальные данные для каждого графика аппроксимируются с использованием метода наименьших квадратов (МНК), встроенного в Excel. Аппроксимация выполняется полиномом второй степени $T(V) = aV^2 + bV + c$, что согласуется с теоретической сложностью алгоритма $O(|V|^2)$ [4].

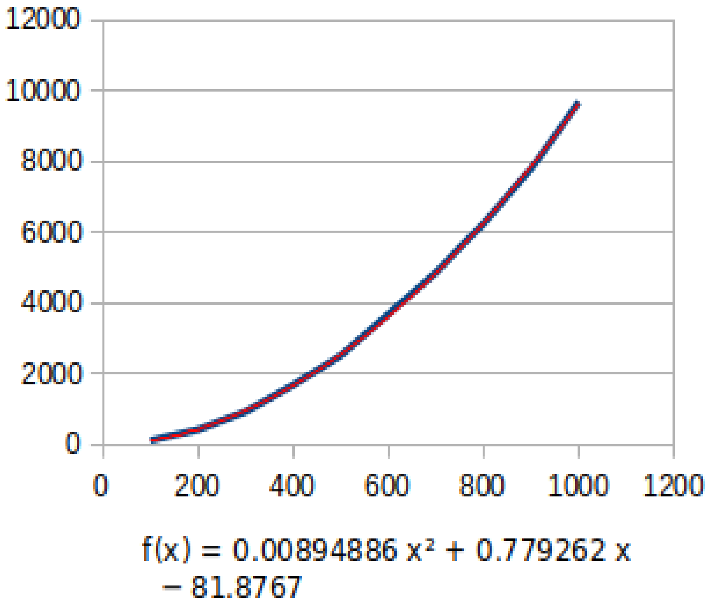

Рис. 1 – график зависимости времени работы алгоритма Дейкстры от количества вершин графа при ρ=0.1. Авторская разработка

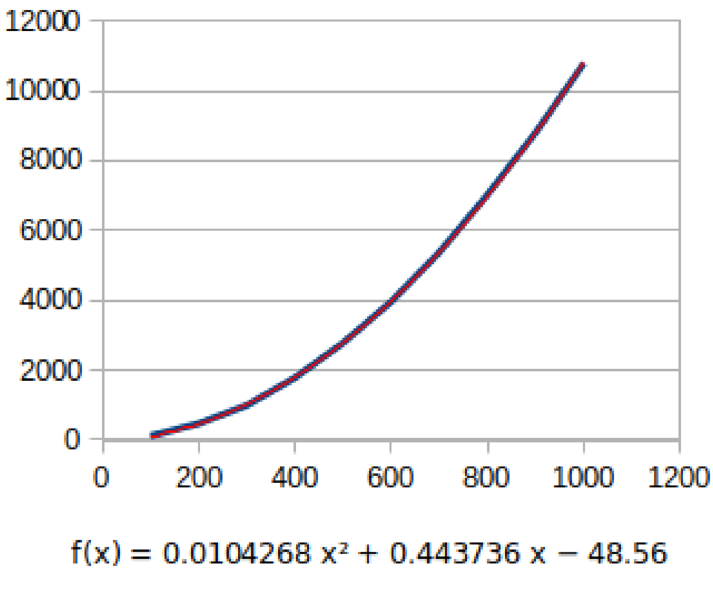

Рисунок 2 – график зависимости времени работы алгоритма Дейкстры от количества вершин графа при ρ=0.2. Авторская разработка

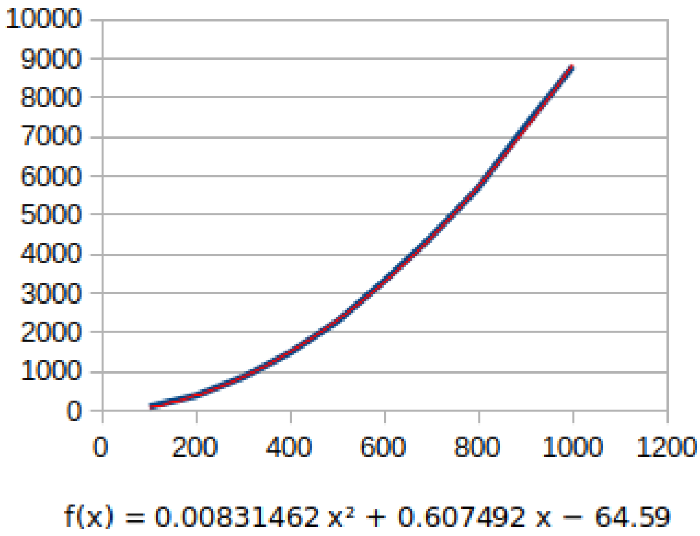

Рисунок 3 – график зависимости времени работы алгоритма Дейкстры от количества вершин графа при ρ=0.3. Авторская разработка

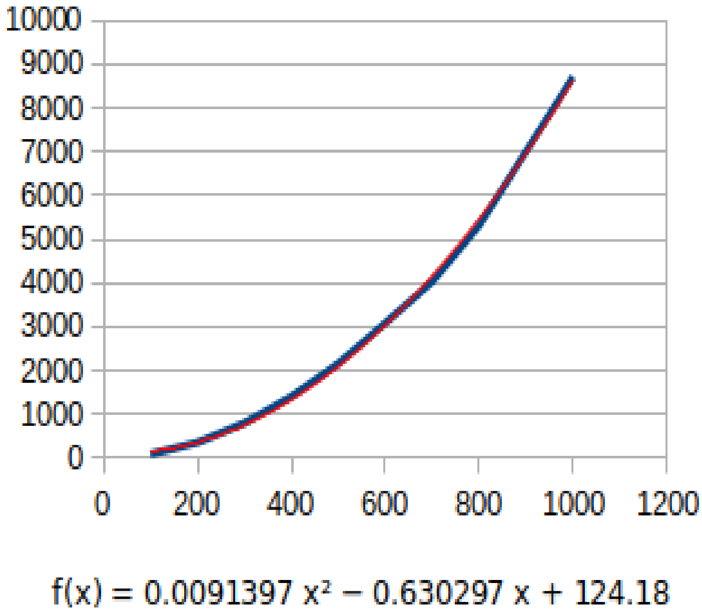

Рисунок 4 – график зависимости времени работы алгоритма Дейкстры от количества вершин графа при ρ=0.4. Авторская разработка

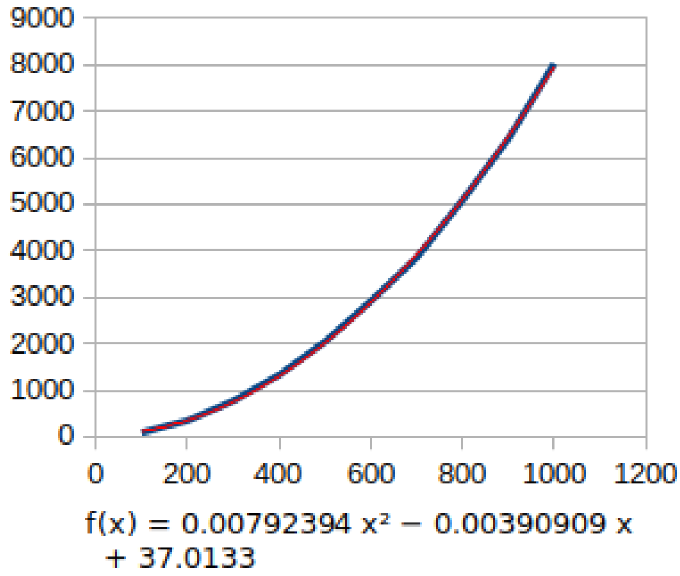

Рисунок 5 – график зависимости времени работы алгоритма Дейкстры от количества вершин графа при ρ=0.5. Авторская разработка

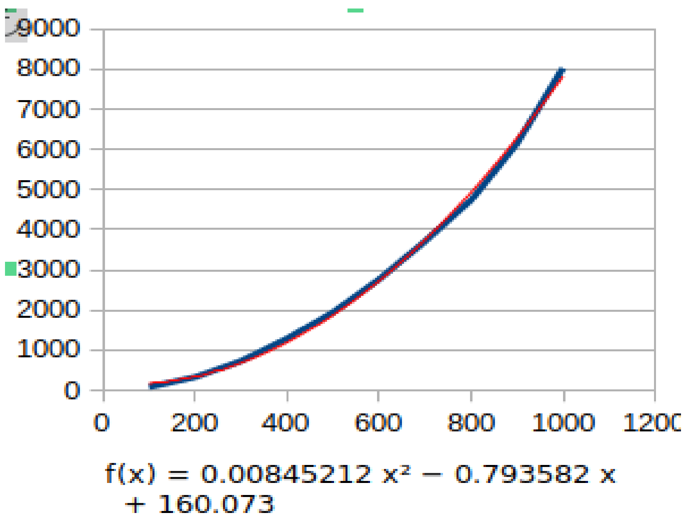

Рисунок 6 – график зависимости времени работы алгоритма Дейкстры от количества вершин графа при ρ=0.6. Авторская разработка

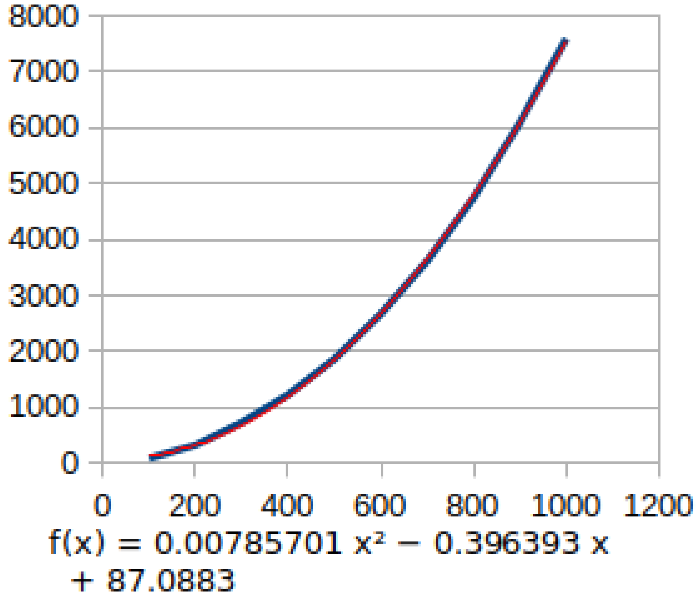

Рисунок 7 – график зависимости времени работы алгоритма Дейкстры от количества вершин графа при ρ=0.7. Авторская разработка

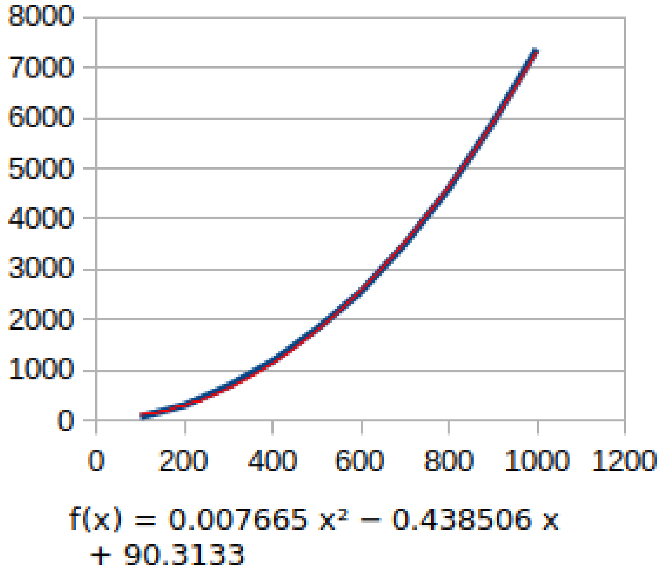

Рисунок 8 – график зависимости времени работы алгоритма Дейкстры от количества вершин графа при ρ=0.8. Авторская разработка

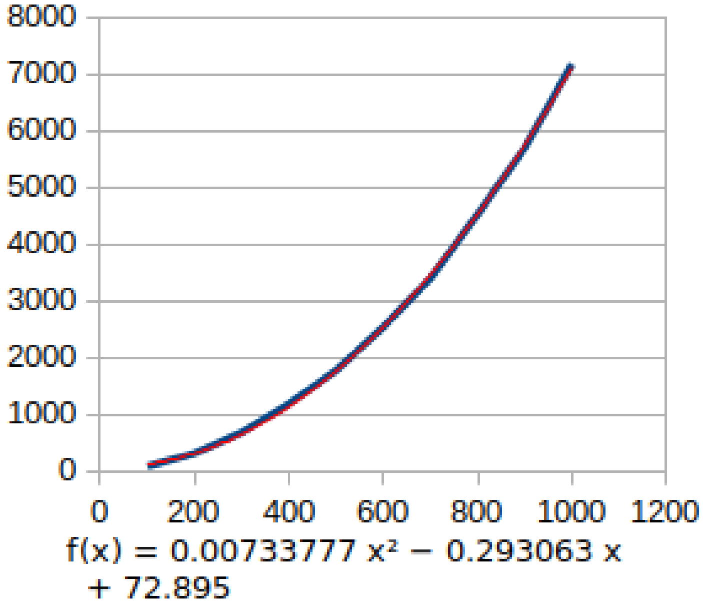

Рисунок 9 – график зависимости времени работы алгоритма Дейкстры от количества вершин графа при ρ=0.9. Авторская разработка

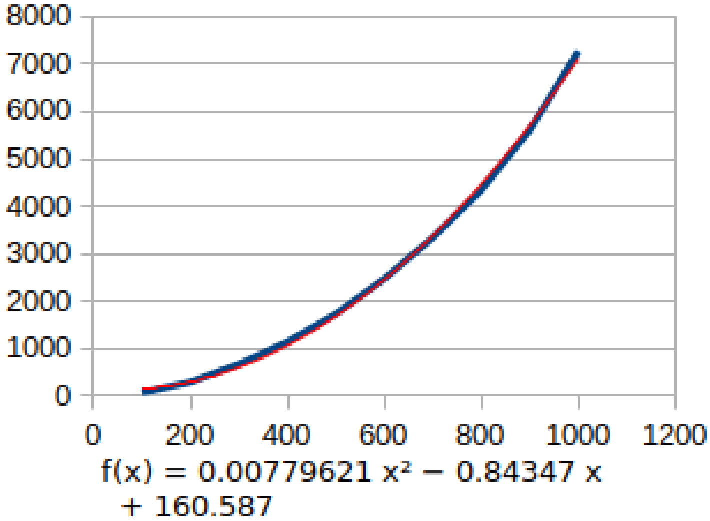

Рисунок 10 – график зависимости времени работы алгоритма Дейкстры от количества вершин графа при ρ=1.05. Авторская разработка

Далее каждый коэффициент a, b и c будем считать функцией, зависящей от ρ. Применяя МНК, подберём подходящую степень полинома для их аппроксимации, после чего получим окончательную функцию зависимости времени выполнения алгоритма Дейкстры $T(V, \rho)$:

$$T(V,\rho) = a(\rho)V^2 + b(\rho)V + c(\rho),$$

где

$$a(\rho) = -0.0025\rho + 0.0098,$$

$$b(\rho) = 1.7709\rho^2 - 3.4273\rho + 1.0464,$$

$$c(\rho) = -300.6357\rho^2 + 558.094\rho - 137.4950.$$

Заключение. В данной работе была изучена роль коэффициента наполненности графа в определении времени выполнения алгоритма Дейкстры для нахождения кратчайших путей в неориентированных графах. На основе экспериментальных данных были получены зависимости времени работы алгоритма от количества вершин графа и коэффициента наполненности. Была получена функция, которая описывает зависимость времени выполнения алгоритма Дейкстры от числа вершин графа и коэффициента его наполненности, что позволяет не только оценивать вычислительную сложность алгоритма в зависимости от параметров графа, но и эффективно планировать ресурсы для обработки данных в задачах, связанных с анализом графов.

Библиографический список:

1. Абрамова О.А., Ковалева В.Е., Ларина В.Е., Иванова А.П. Сравнительный анализ сложности алгоритмов Флойда и Беллмана-Форда для графов с различным количеством рёбер // Дневник науки. – 2025. – № 2 [Электронный ресурс]. (Дата обращения 07.03.2025).

2. Алексеев В.Е, Захарова Д.В. Теория графов: Учебное пособие. – Нижний Новгород: Нижегородский госуниверситет, 2017. – 119 с.

3. Дистель Р. Теория графов: Пер. с англ. – Новосибирск: Изд-во Ин-та математики, 2002. – 336 с.

4. Кормен, Томас Х., Лейзерсон, Чарльз И., Ривест, Рональд Л., Штайн, Клиффорд. Алгоритмы: построение и анализ, 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2022. – 1296 с.

5. Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир, 1978. – 432 с.

6. Лекции по теории графов / Емеличев В.А., Мельников О.И., Сарванов В.И., Тышкевич Р.И. – М.: Наука, 1990. – 384 с.

7. Новиков Ф.А. Дискретная математика для программистов. Учебник для вузов. 2-е изд. – СПб.: Питер, 2007. – 364 с.

8. Озерова, Г.П. Основы программирования на языке Python в примерах и задачах: учебное пособие для вузов / Политехнический институт ДВФУ. – Владивосток: Изд-во Дальневост. федерал. ун-та, 2022. – 128 с.

9. Оре О. Теория графов. – 2-е изд. - М.: Наука, 1980. – 336 с.

10. Теория графов. Часть 1: учебное пособие по дисциплине «Теория графов»: Иванова А.П. – Москва: Янус-К, 2024. – 96 с.

Оригинальность 80%