Санкт-Петербургский государственный университет

Васильев Василий Васильевич

Выпускная квалификационная работа бакалавра

Оценка трудоемкости алгоритма на основе эмпирического анализа

Направление 02.03.02
«Фундаментальная информатика и информационные технологии»
ООП СВ.5003.2016: «Программирование и информационные технологии»

Научный руководитель:
кандидат физ.-мат. наук, доцент
кафедры моделирования электромеханических
и компьютерных систем:
Никифоров Константин Аркадьевич

Санкт-Петербург
2020

Содержание

Введение

Оценка эффективности алгоритмов является важным этапом в создании качественных программных средств, причем один из критериев качества — временная эффективность, особенно актуальная для систем, работающих в режиме реального времени. Очевидно, что временная эффективность компьютерной программы связана с функцией трудоемкости алгоритма, т. е. с точным количеством операций, задаваемых алгоритмом в основе программной реализации. Однако, асимптотические оценки вычислительной сложности, получаемые в теоретическом исследовании алгоритмов, не всегда справедливы для конечного диапазона длин входов, что объясняется большими значениями коэффициентов у компонент функции трудоемкости. В работе предлагается практический подход на основе эмпирического анализа времени выполнения программной реализации, для чего создана автоматизированная система с критерием оценки по величине доверительной трудоемкости в выбранном диапазоне входных данных.

Вычисление доверительной трудоемкости связано с построением доверительных интервалов оцениваемой величины трудоемкости с заданной доверительной вероятностью в классическом подходе математической статистики [1]. Данный метод требует использования репрезентативных выборок достаточно большого объема и многократного запуска программных реализаций исследуемых алгоритмов в соответствующем многоэтапном процессе; поэтому, несомненно, актуальной является разработка системы для автоматизированного проведения анализа, значительно сокращающего время оценки качества алгоритмов.

В данной работе рассматривается построение такой системы и исследование применимости предложенного подхода на основе доверительной трудоемкости для более широкого класса алгоритмов.

Постановка задачи

Основными целями данной работы являются:

- исследование применимости существующего подхода для более широкого класса алгоритмов;
- создание программного обеспечения для автоматизации вычисления доверительной трудоемкости алгоритма и сравнение полученных результатов с классическим эмпирическим подходом.

Задача исследования применимости состоит в ознакомлении с предложенным подходом и его адаптация для поддержки более широкого класса алгоритмов. В первую очередь стоит цель рассмотреть возможность применения альтернативных методов, не затронутых в работе [1], на разных этапах анализа функции трудоемкости.

Задача создания автоматизированной системы состоит в разработке программного обеспечения, которое при некоторых заданных начальных данных, будет проводить многоэтапный анализ и вычислять доверительную трудоемкость для предоставленных программных реализаций алгоритмов.

Обзор литературы

При разработке алгоритма для решения поставленной задачи всегда стоит вопрос о его корректности, для чего необходимо провести анализ алгоритма. Данный анализ включает в себя доказательство корректности или правильности алгоритма и установление его емкостных и временных характеристик, т. е. определение того, какое количество ресурсов требуется алгоритму для решения задачи. В данной работы рассматривается именно вторая часть анализа при начальном условии, что все рассматриваемые алгоритмы корректно решают поставленные задачи. Алгоритм считается корректным (правильным), если при любых допустимых входных данных он заканчивает работу и выдает удовлетворяющий требованиям задачи результат.

В зависимости от того, когда производится анализ (до или после реализации алгоритма), его можно разделить на два этапа:

- Априорный (или теоретический) анализ — анализ алгоритма перед его запуском на определенной системе. При этом принимается предположение, что иные факторы, такие как производительность комплектующих на целевом компьютере, являются постоянными и не оказывают влияния на реализацию алгоритма.

- Апостериорный (или экспериментальный) анализ — анализ алгоритма выполняется только после его запуска на электронно-вычислительной машине с определенными комплектующими. К этому моменту для исследуемого алгоритма должна быть представлена программная реализация на одном из языков программирования. Данный вид анализа напрямую зависит от конфигураций системы и ее комплектующих.

При использовании апостериорного анализа стоит учитывать, что временная и емкостная сложность алгоритма может варьироваться, в зависимости от системы и программной реализации. При этом априорный анализ использует только асимптотические оценки сложности алгоритма,

зависящие от входных данных и их размеров, а не от системных или программных конфигураций.

Одними из первых методов оценки алгоритмов, ставших классическими, считается ряд подходов, рассматривающих различные аспекты сложности для алгоритмического формализма машины Тьюринга. Данная модель является одной из основной формальной моделью из тех, что применяются в теории алгоритмов. В работах [2, 3] были сформированы следующие два основных подхода к решению задачи оценки алгоритмов:

- оценка сложности программной реализации алгоритма;
- оценка сложности вычислительного процесса, задаваемого алгоритмом.

В первом подходе используется оценка количества содержащейся в записи алгоритма информации. Данную оценку сложности самого алгоритма можно связать с объемом программы, реализующей данный алгоритм. В качестве оценки сложности алгоритма рассматривается объем оперативной памяти в области кода, занимаемый его программной реализацией.

Во втором подходе вводится мера сложности вычислений, задаваемых алгоритмом для конкретных допустимых задач. В работе [4] используется некоторый функционал, соотносящий алгоритму и индивидуальной (конкретной) задаче определенное число.

В работе [5] представлен инструментарий для сравнения алгоритмов и оценки объективной трудности, присущей различным вычислимым функциям, путем введения некоторой меры сложности алгоритмов.

Однако подход, основанный на мерах сложности, имеет ряд недостатков. Один из них состоит в ограничении сложности вычисляемой функции f более слабой характеристикой [6]. Для функции f вводятся две функции ϕ_1 и ϕ_2, являющиеся нижней и верхней оценкой соответственно. В данном подходе близость оценок определяет точность, с которой связана сложность функции f. Несмотря на это, указанный подход используется в теории сложности арифметических вычислений [7] и при асимптотическом анализе сложности алгоритмов [8].

Одной из первых фундаментальных работ в области математического анализа сложности алгоритмов является [9]. Однако в ней, как и в других центральных работах по анализу алгоритмов [8, 10, 11] используются методы для вычисления трудоемкости алгоритма в среднем.

Также стоит отметить методы машинного обучения, которые активно совершенствуются в последние десятилетия и используются в различных областях, таких как медицина, экономика, компьютерное зрение, биоинформатика, поиск информации и др. Их приложения были использованы для улучшения алгоритмов, основанных на входных данных [12–14]. Можно предположить, что метод исследования алгоритмов с помощью машинного обучения будет активно развиваться в будущем, конкурируя с классическими подходами.

Совершенно новый подход был предложен в работе [1] для повышения точности результатов эмпирического анализа алгоритма. Для решения указанной задачи авторы рассматривают сложность алгоритма при заранее заданном и зафиксированном входном размере данных. Основная идея подхода состоит в построение доверительного интервала трудоемкости алгоритма. Для этого авторы используют бета-распределение в качестве непрерывного распределения для аппроксимации значений трудоемкости. При этом для исследуемых допустимых входов вводятся ограниченные дискретные случайные величины с неизвестным распределением. На практике данный подход показывает более реальные границы сложности алгоритма для рассматриваемых входных данных при заданном коэффициенте доверия.

Описанный метод включает в себя два этапа:

- предварительный этап — проверка гипотезы о законе распределения трудоемкости алгоритма как ограниченной дискретной случайной величины [15];
- основной этап — определение значения доверия трудоемкости $f_\gamma(n)$ в зависимости от длины n входного алгоритма [1].

Глава 1. Обзор существующих решений

В ранних работах системы автоматизированной оценки эффективности алгоритмов на основе анализа программных реализаций [16,17] ограничивались использованием специальных языков программирования (Nuprl и функциональный язык), что не подходит для реализаций алгоритмов, разработанных с помощью других систем или языков программирования. Более поздние системы на основе машинного обучения [13] работают как с параметризованными алгоритмами, так и без параметров, причем обладают большой гибкостью при построении регрессионных моделей на основе эмпирического анализа программных реализаций на различных языках программирования. В последнее время в области вычислительного интеллекта развиваются альтернативы статистическим подходам в выборе наилучшего или более подходящего алгоритма для решения конкретной задачи [14].

Однако, в упомянутых системах даются лишь точечные оценки трудоемкости (мода, медиана, математическое ожидание, коэффициент вариации), что не позволяет получить какие-либо сведения о поведении алгоритма на конкретном входе. Преимущество предлагаемой автоматизированной системы заключается в использовании доверительной трудоемкости, как гарантирующей оценки, повышающей точность результатов эмпирического анализа

Глава 2. Исследование трудоемкости

2.1 Общие положения исследования

Основной целью исследования является построение доверительной функции трудоемкости для рассматриваемого алгоритма. Стоит отметить, что объектом анализа выступает именно сам алгоритм, поскольку его программные реализации могут иметь совершенно разные показатели производительности на целевых компьютерах. В дальнейшем будем использовать следующее общепринятое определение трудоемкости:

Определение. Трудоемкость алгоритма A на входе D — количество базовых операций в заданной модели вычислений на рассматриваемом входе.

Также введем обозначение $f_A(D)$ для функции трудоемкости алгоритма A на входе D и D_n для совокупности всех входов D алгоритма A размерности n. Предполагается, что закон распределения значений $f_A(D)$ неизвестен.

Теоретическое исследование трудоемкости имеет некоторые трудности при построении вероятностной модели для рассматриваемого алгоритма, такие как определение полной группы событий и определение вероятностной меры [18]. Поэтому разумным и единственным путем на данный момент остается экспериментальное исследование. В рамках такого исследования для трудоемкости вводится ограниченная дискретная случайная величина с неизвестным распределением и строится гистограмма относительных частот. Далее полученная гистограмма аппроксимируется некоторой функцией. Обычно аппроксимирующую функцию выбирают из множества хорошо изученных функций плотности распределения вероятностей.

Для доказательства того факта, что аппроксимация выбранной функцией плотности распределения является корректной, необходимо сформировать и доказать гипотезу о соответствующем распределении относительных частот значений функции трудоемкости [15]. В случае, если гипотеза не будет доказана, необходимо выбрать другую функцию и повторить про-

цедуру.

Основная проблема теоретических значений трудоемкости состоит в том, что для полученные зависимости количества базовых операций от размеров входов могут быть не актуальны для подмножества входов алгоритма, которые используются в конкретных задачах на практике. Таким образом, актуальной становится проблема сокращения длины сегмента для оценки трудоемкости на рассматриваемом подмножестве входов.

Именно доверительная трудоемкость призвана решить описанную проблему [1]. Для выбранной аппроксимирующей функции распределения исследователями задается коэффициент доверия γ [19]. После решения интегрального уравнения для рассматриваемой функции распределения можно получить значение $f_\gamma(n)$ трудоемкости, где γ — заданный коэффициент доверия. Данное значение называется доверительной трудоемкостью.

Одним из его важных свойств является то, что оно не будет превышено никаким другим значением функции трудоемкости для рассматриваемого единичного входа с коэффициентом доверия γ. В данном случае длина сегмента может быть существенно сокращена, поскольку для единичного входа алгоритма трудоемкость будет заключена в сегменте $[f_A^\vee, f_\gamma]$: между лучшим случаем и значением $f_\gamma(n)$ с вероятностью γ.

2.2 Построение гистограммы частот

Для построения гистограммы относительных частот по экспериментальным данным необходимо сперва нормировать данные. Введем случайную нормированную величину T. Ее реализации t_i получаются на основе теоретических и эмпирических значений трудоемкости [1]:

$$t_i = \frac{f_i - f^\vee}{f^\wedge - f^\vee},$$

где f_i — значение трудоемкости для сгенерированных случайных допустимых входов D_i: $f_i = f_A(D_i), i = \overline{1,m}$, а $f^\wedge, f^\vee$ — теоретический максимум и минимум функции трудоемкости соответственно. При этом отметим, что нормированные величины t_i принимают значения из сегмента $[0,1]$.

Далее необходимо определить оптимальное количество полусегмен-

тов для построения гистограммы. Оптимальность числа полусегментов заключается в достаточном представлении функции плотности распределения вероятностей.

Стоит учесть тот факт, что частоты, вычисляемые для построения гистограммы, будут использоваться при рассмотрении гипотезы о законе распределения. Поэтому не подходит группа методов, которые вычисляют сперва длину интервала, а только потом количество полусегментов. К таким относятся, например, формулы Скотта [20] и Фридмана – Диакониса [21]. Указанное множество способов группировки данных по ширине интервала не подходит, поскольку для многих критериев согласия длины интервалов могут быть различными. Также некоторые критерии согласия требуют определенной группировки данных (например, критерий χ^2).

Рассмотрим несколько способов вычисления количества полусегментов k.

2.2.1 Формула Стерджесса

Одним из самых популярных методов определения количества полусегментов является использование формулы Стерджесса [22]:

$$k = 1 + \lfloor \log_2 n \rfloor,$$

где n — общее число наблюдений. Также встречается вариант записи через десятичный логарифм:

$$k = 1 + \lfloor 3.322 \cdot \lg n \rfloor.$$

Данная эмпирическая формула основывается на оценке количества событий с разными вероятностями в схеме Бернулли длительностью в $k-1$ этап. Однако в последнее время формула подвергается критике и все реже используется специалистами [23]. Главная претензия заключается в том, что формула аппроксимирует нормальное распределение биномиальным, что применимо далеко не во всех ситуациях. Формула Стерджесса применима только для выборок небольшого размера ($m < 200$). В иных случаях

настоятельно рекомендуется использовать другие методы определения количества полусегментов для гистограммы.

2.2.2 Эмпирический метод

Другим методом определения количества полусегментов является эмпирическое правило:

$$k = \left[\sqrt{n}\right], \tag{1}$$

где n — общее число наблюдений. Вместо формулы (1) может быть использована любая другая формула, позволяющая исследователям получить достаточное количество полусегментов.

2.2.3 Принцип наихудших измерения

Одна из новейших и практически значимых оценок для вычисления количества полусегментов основывается на устойчивости относительной частоты выборки [18]:

$$k = \left\lfloor \frac{n}{n_i} \right\rfloor < \frac{n}{1 + [t(\gamma, n_i)]^2}, \tag{2}$$

где n — общее число наблюдений, n_i — число наблюдений в i-ом полусегменте, $t(\gamma, n_i)$ — значения критерия Стьюдента при заданном коэффициенте доверия γ и объеме группы n_i.

В основе оценки (2) находится предположение, что относительная частота становится устойчивой одновременно со средней групповой по полусегменту, вследствие чего рассматривается интервальная оценка для выборки исходя из принципа наихудших измерений.

Существуют и другие методы определения количества полусегментов гистограммы. Однако выбор соответствующего метода зависит от задачи исследователей и вида распределения данных.

2.3 Определение объема выборки

При построении гистограммы остался нерешенным вопрос о требуемом размере выборки. Стоит отметить, что в некоторых случаях исследователи имеют возможность работать только с ранее полученными выборками. Повторное воспроизведение эксперимента для извлечения новых выборок иного объема невозможно, трудоемко или дорого. Однако более распространены ситуации, когда возможность извлекать выборки есть. Даже в этом случае, очевидно, приоритет направлен в сторону минимизации объема выборок для эксперимента, что позволит уменьшить временные, ресурсные и финансовые затраты исследователей. Поэтому возникает проблема определения минимального числа экспериментов m для вычисления трудоемкости алгоритма при заданной доверительной вероятности γ. Заметим, что длина входа n при этом остается постоянной.

2.3.1 Метод с использованием схемы Бернулли

При известной вероятности p значений трудоемкости с наименьшей частотной встречаемостью можно применить метод на основе схемы Бернулли [18]. Его суть заключается в том, что величина p трактуется как вероятность успеха и рассматривается событие A — наблюдение как минимум одного успеха в m испытаниях с заданной вероятностью γ. Сама задача сводится к определению числа испытаний n в схеме Бернулли:

$$P(A) = 1 - (1 - p)^m \geq \gamma,$$

откуда можно получить требуемый объем выборки:

$$m \geq \left\lceil \frac{\ln(1 - p)}{\ln(1 - \gamma)} \right\rceil.$$

2.3.2 Метод на основе закона распределения

Более общий метод основан на рассмотрении гипотезы о законе распределения функции трудоемкости алгоритма [15]. Поскольку закон распределения значений $f_A(D)$ неизвестен, вводится гипотеза, что значения

функции трудоемкости являются ограниченной дискретной случайной величиной, которая распределена по одному из рассмотренных ранее законов распределения. Суть данного метода сводится к тому, что для определения минимального объема выборки m проводятся ряд последовательных экспериментов с постоянной длиной входа n. Значение n задается исследователями, как и начальный объем выборки m.

На каждой итерации извлекаются выборки размера m и вычисляются значения трудоемкости. Далее рассчитывается ряд статистических величин: выборочное среднее $\overline{f_э}(m)$ и выборочная исправленная дисперсия S^2, которые являются оценками теоретической трудоемкости $\overline{f_A}$ и теоретической дисперсии трудоемкости σ_A^2 соответственно. Требуемый объем выборки рассчитывается по формуле:

$$m^* = m^*(\delta, \gamma) = \min m : P(\left|\overline{f_э}(m) - \overline{f_A}\right| \leq \delta) \leq \gamma,$$

т. е. минимальный объем выборки нужно выбирать таким образом, чтобы средние значения в выборке $\overline{f_э}$ позволяли построить доверительный интервал длиной 2δ, который покрывал бы неизвестное значение $\overline{f_A}$ с надежностью γ.

При этом условием останова для описанной последовательности итераций является выполнение неравенства:

$$m^*_{(\mathrm{i}\,+\,1)} < m^*_{(\mathrm{i})},$$

где $m^*_{(\mathrm{i})}$ — рассчитанный минимальный объем выборки на итерации i.

2.4 Используемые законы распределения

При проведении анализа функции трудоемкости одной из основных целей является получение функциональной зависимости трудоемкости в введенной абстрактной модели вычислений, что позволит прогнозировать временные оценки алгоритма на конкретных входах. Для достижения этой цели необходимо аппроксимировать гистограмму относительных частот некоторой функцией, которую обычно выбирают из множества хорошо изучен-

ных функций плотности распределения вероятностей.

Отметим, что функция трудоемкости алгоритма ограничена теоретическими значениями в худшем и лучшем случаях, т. е. ее значения находятся в сегменте $[f_A^\vee, f_A^\wedge]$ при постоянной длине входа n. Таким образом, для аппроксимации следует использовать только функции плотности распределения вероятностей с ограниченной вариацией [18]. Данные функции заданны на конечном множестве сегментов действительных чисел. Это множество и ограничивает область возможных значений случайной величины.

Рассмотрим некоторые функции плотности $f(x)$ ограниченной случайной величины X на сегменте $[a, b]$ $(a < b)$. Не умаляя общности, рассматриваться будут функции плотности, ограниченные только одним сегментом. При этом функция $f(x)$ должна удовлетворять следующим условиям:

1. $f(x) \geq 0, \quad \forall x \in [a, b]$;
2. $f(x) = 0, \quad \forall x \notin [a, b]$;
3. $\int_a^b f(x)dx = 1$.

2.4.1 Равномерное распределение

Равномерное распределение задается функцией плотности:

$$f(x) = \begin{cases} \dfrac{1}{b-a}, & x \in [a, b], \\ 0, & x \notin [a, b]. \end{cases}$$

Равномерное распределение можно свести к стандартному равномерному распределению на промежутке $[0, 1]$ с помощью линейного преобразования $u = (x - a)/(b - a)$ [24]:

$$f(u) = \begin{cases} 1, & u \in [0, 1], \\ 0, & u \notin [0, 1]. \end{cases}$$

Равномерное распределение, несмотря на свою простоту, находит широкое практическое применение. Отметим, что стандартное равномерное распределение является частным случаем бета-распределения с параметрами $\alpha = 1, \beta = 1$.

2.4.2 Треугольное распределение (распределение Симпсона)

Треугольное распределение (распределение Симпсона) задается функцией плотности:

$$f(x) = \begin{cases} \dfrac{2}{b-a} - \dfrac{2}{(b-a)^2} \cdot |a + b - 2x|, & x \in [a, b], \\ 0, & x \notin [a, b]. \end{cases}$$

Треугольное распределение можно свести к стандартному треугольному распределению на промежутке $[0, 1]$ с помощью линейного преобразования $u = (x - a)/(b - a)$ [24]:

$$f(u) = \begin{cases} 2 - 2 \cdot |1 - 2u|, & u \in [0, 1], \\ 0, & u \notin [0, 1]. \end{cases}$$

Распределение Симпсона применяется при малых размерах выборок и недостаточном количестве данных. Отметим, что стандартное треугольное распределение представимо в виде комбинации двух бета-распределений:

$$f(u) = \frac{4 \cdot \left\lfloor \frac{3}{2} - u \right\rfloor \cdot f_1(u) + 4 \cdot \left\lfloor \frac{1}{2} + u \right\rfloor \cdot f_2(u)}{\left\lfloor \frac{5}{2} - u \right\rfloor \cdot \left\lfloor \frac{5}{2} - u \right\rfloor},$$

где $f_1(u) = 2 \cdot u$ — бета-распределение с параметрами $\alpha = 2, \beta = 1$,
$f_2(u) = 2 \cdot (1 - u)$ — бета-распределение с параметрами $\alpha = 1, \beta = 2$.

2.4.3 Бета-распределение

Бета-распределение задается функцией плотности [25]:

$$f(x) = \begin{cases} \dfrac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\cdot\Gamma(\beta)\cdot(b-a)^{\alpha+\beta+2}} \cdot (x-a)^{\alpha-1} \cdot (b-x)^{\beta-1}, & x \in [a,b], \\ 0, & x \notin [a,b], \end{cases}$$

где α, β — параметры бета-распределения ($\alpha > 0, \beta > 0$), Γ — гамма-функция Эйлера [26].

Бета-распределение можно свести к стандартному бета-распределению на промежутке $[0,1]$ с помощью линейного преобразования $u = (x-a)/(b-a)$ [24]:

$$f(u) = \begin{cases} \dfrac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\cdot\Gamma(\beta)} \cdot u^{\alpha-1} \cdot (1-u)^{\beta-1}, & x \in [0,1], \\ 0, & x \notin [0,1]. \end{cases}$$

Бета-распределение отличается большой универсальностью поведения. Плотность вероятности является очень гибкой за счет своих параметров α и β, что позволяет можно получать другие распределения (например, равномерное распределение). Аналогичного результата можно добиться, комбинируя несколько бета-распределений с разными значениями параметров (например, треугольное распределение).

Указанные особенности позволяют утверждать, что исследователям стоит в первую очередь рассматривать бета-распределение. Однако выбор распределения должен быть подтвержден проверкой гипотезы о законе распределения.

2.5 Восстановление параметров функции плотности

Перед рассмотрением гипотезы о законе распределения для выбранной функции плотности необходимо проделать еще один шаг — восстановить ее параметры на основе данных выборки. Для этого существует несколько активно используемых методов:

- метод моментов [24, 25];
- метод максимального правдоподобия [24];
- метод минимума χ^2 [24].

Стоит отметить, что не существует универсального метода для определения параметров функции плотности. Применение соответствующего метода зависит от задачи исследователей и вида распределения данных, как и в случае с выбором метода для определения полусегментов гистограммы относительных частот.

2.6 Рассмотрение гипотезы о законе распределения

При проведении анализа трудоемкости необходимо убедиться, что выбранный закон распределения соответствует наблюдаемым экспериментальным данным. В таком случае необходимо сформировать, рассмотреть и принять или отвергнуть гипотезу о соответствующем законе распределения.

Рассмотрим несколько широко применяемых критериев проверки статистических гипотез.

2.6.1 Критерий согласия Пирсона (χ^2)

Критерий Пирсона проверяет значимость расхождения теоретической плотностью и эмпирической гистограммы относительных частот. В качестве нулевой гипотезы H_0 выступает предположение о соответствии выбранного теоретического закона распределения реальным результатам экспериментов. Заметим, что гипотеза H_0 является сложной, поскольку рассматривается закон распределения с восстановленными по выборке параметрами. Тогда критерий χ^2 вычисляется по формуле [24]:

$$\chi^2_m = m \cdot \sum_{i=1}^{k} \frac{O_i - E_i}{E_i}, \tag{3}$$

где O_i, E_i — относительные эмпирические и теоретические частоты в i-ом сегменте гистограммы соответственно, m — общий размер выборки, k —

количество полусегментов гистограммы. Величина χ^2_m имеет закон распределения χ^2 с s степенями свободы ($s = k - 1 - r$, где r — число параметров выбранного распределения).

В некоторых источниках в формуле (3) множитель m отсутствует ввиду того, что рассматриваются абсолютные частоты, а не относительные. Использование относительных частот обусловлено нормированием рассматриваемого сегмента трудоемкости $[f_A^\vee, f_A^\wedge]$ на сегмент $[0, 1]$.

Рассчитанное по формуле (3) значение сравнивается со значением правосторонней критической области:

$$\chi^2_m < \chi^2_{\text{кр}}(\alpha', s), \tag{4}$$

где α' — заданный уровень значимости, s — количество степеней свободы, а значение $\chi^2_{\text{кр}}(\alpha', s)$ определяется по теоретическому распределению χ^2_m. Гипотеза H_0 принимается, если выполняется неравенство (4). В противном случае H_0 отвергается, необходимо выбрать другой закон распределения, построить соответствующую гистограмму относительных частот и снова рассмотреть гипотезу о законе распределения.

Отметим, что критерий Пирсона является состоятельным, т. е. почти наверняка отвергает ошибочную гипотезу при достаточно большом количестве испытаний. Также данный критерий обеспечивает минимальную возможную ошибку второго рода по сравнению с другими критериями [27]. Рекомендуется использовать критерий χ^2 при достаточно большом выборки ($m > 100$) и количестве наблюдений в отдельном сегменте гистограммы относительных частот более 10 [24]. Если в какой-либо сегмент попадает менее 10 значений выборки, необходимо объединить его с ближайшим. Иначе следует использовать другие критерии проверки статистических гипотез.

2.6.2 Критерий согласия Колмогорова

Критерий Колмогорова проверяет значимость расхождения эмпирической и теоретической функций распределения. Нулевая гипотеза H_0 выбирается так же, как и при рассмотрении критерии Пирсона. Тогда крите-

рий Колмогорова вычисляется по формуле [24]:

$$D_m = \sup_{|x|<\infty} |F_m(x) - F(x,\theta)|,$$

где $F_m(x)$ — эмпирическая функция распределения, $F(x,\theta)$ — теоретическая функция распределения.

По теореме Колмогорова [24] при справедливости проверяемой гипотезы:

$$\forall z > 0 : \lim_{n\to\infty} P\{\sqrt{m} \cdot D_m < z\} = K(z) = \sum_{j=-\infty}^{+\infty} (-1)^j \cdot e^{-2\cdot j^2 \cdot z^2}.$$

Гипотеза H_0 принимается, если значение $\sqrt{m} \cdot D_m$ не превышает квантиль распределения $K_{\alpha'}$, где α' — заданный уровень значимости.

Классический критерий Колмогорова стоит применять только для простых гипотез. При применении данного критерия для сложных гипотез появляются определенные различия, которые необходимо учитывать отдельно [28].

2.7 Прогнозирование функции трудоемкости

Получение значений доверительной трудоемкости $f_\gamma(n)$ на интересующем исследователей сегменте требует значительных вычислительных затрат. Это обусловлено применением одного из методов восстановления параметров аппроксимирующего распределения на основе экспериментальных данных. Для решения проблемы сокращения временных затрат можно попытаться спрогнозировать выборочную дисперсию и выборочную среднюю с помощью функций регрессии. Уравнения регрессии могут быть построены на основе анализа имеющихся данных из меньшего сегмента входных данных. Таким образом, это позволит экстраполировать значения доверительной трудоемкости $f_\gamma(n)$ на весь интересующий сегмент, не проводя дополнительных экспериментов.

Обычно строят несколько функций регрессии, затем выбирают наи-

лучшую из них. Набор функций для рассмотрения зависит от конкретной задачи и определяется оценкой имеющихся данных. Для задачи прогнозирования трудоемкости выбор функции регрессии можно осуществить одним из методов регрессионного анализа. Далее будет использоваться метод наименьших квадратов [29]. Сравнение функций разного вида будет производиться по коэффициенту детерминации R^2.

2.7.1 Коэффициент детерминации (R^2)

Коэффициент детерминации R^2 определяет долю дисперсии зависимой переменной, объясняемой рассматриваемой моделью. Истинный коэффициент детерминации модели зависимости случайной величины y от факторов x задается формулой [29]:

$$R^2 = 1 - \frac{D(y|x)}{D(y)} = 1 - \frac{\sigma^2}{\sigma_y^2},$$

где $D(y|x)$ — дисперсия случайной ошибки модели, $D(y)$ — дисперсия случайной величины y. Однако интерес представляет выборочный коэффициент детерминации (который обычно и подразумевается под коэффициентом детерминации):

$$R^2 = 1 - \frac{\widehat{\sigma^2}}{\widehat{\sigma_y^2}} = 1 - \frac{SS_{res}/m}{SS_{tot}/m} = 1 - \frac{SS_{res}}{SS_{tot}},$$

где SS_{res} — сумма квадратов остатков регрессии, SS_{tot} — общая сумма квадратов.

Коэффициент детерминации является одним из наиболее используемых критериев оценки качества линейных и нелинейных моделей. Данный коэффициент следует применять с осторожностью, поскольку его значение увеличивается от добавления в модель новых переменных, даже в случаях, когда новые переменные не оказывают никакого влияния на объясняемую переменную.

Глава 3. Проведение экспериментального исследования трудоемкости

3.1 Описание выбранного алгоритма

Нахождение кратчайшего пути от одной вершины до всех остальных является одной из основных задач оптимизации сети. Существует несколько распространенных и хорошо известных алгоритмов, которые решают эту проблему. Основные категории состоят из алгоритма Дейкстры [30] и его модификаций, а также из алгоритма Беллмана – Форда – Мура [31–33] и его модификаций. Есть также ряд других алгоритмов, которые не входят в эти категории.

Алгоритм Паллоттино — алгоритм, который находит кратчайшее расстояние от одной из вершин до всех остальных на графах без петель, является модификацией алгоритма Беллмана – Форда – Мура. Этот алгоритм также работает для графов с ребрами отрицательного веса при отсутствии отрицательных циклов (циклов с отрицательной суммой длин ребер). Он широко используется для решения задач оптимального распределения грузопотоков по транспортной сети и выбора наиболее выгодных путей ее развития.

В литературе есть неоднозначность с наименованием этого алгоритма. Пейп развил идею Д'Эсопо [34] и предложил улучшенный алгоритм [35]. В то же время Левит и Лившиц разработали свою версию с той же идеей [36]. Таким образом, имеем алгоритм Д'Эсопо – Пейпа – Левита, который использует двухстороннюю очередь для поддержания помеченных вершин в графе. Позже Паллоттино предложил использовать две очереди вместо двухсторонней очереди, чтобы избежать экспоненциальной сложности алгоритма в некоторых случаях [37]. Именно последний алгоритм и является объектом исследований в данной работе.

Входными данными для алгоритма является граф G, представленный в виде упорядоченной совокупности множеств вершин V и ребер E: $G = (V, E)$. Оценка размера входных данных производится по размеру множеств вершин и ребер. Пусть n — количество вершин в множестве V,

l — количество ребер в множестве E: $n = |V|, l = |E|$. В качестве базовой операции алгоритма выберем операцию релаксации ребра.

Для анализа алгоритма Паллоттино был использован генератор полных графов без мультиребер. Стоит отметить, что генератор входных данных должен обеспечивать репрезентативность выборки, т. е. генерировать такие входные данные, которые по вероятности соответствуют особенностям применения данного алгоритма. Поскольку алгоритм Паллоттино используется для решения транспортных задач, в реализацию генератора было добавлено условие: для любых трех узлов графа должно выполняться неравенство треугольника. Пусть $G = (V, E)$ — полный неориентированный граф с n вершинами, $W : E \to R_+$ — функция весов ребер. Тогда неравенство треугольника имеет вид:

$$W((u, w)) \leq W((v, u)) + W((u, w)), \forall u, v, w \in V.$$

Отметим, что алгоритм Паллотино имеет оценки

$$O(n, l) = n,$$

$$O(n, l) = n \cdot l,$$

$$O(n, l) = n^2 \cdot l \tag{5}$$

в лучшем, среднем и худшем случаях соответственно [37].

3.2 Этап предварительного исследования

3.2.1 Основные этапы

Поскольку организация экспериментального исследования сложности алгоритма требует подсчета выполняемых основных операций на полученном входе, в исходный код реализации алгоритма Паллоттино были добавлены счетчики. По результатам каждого эксперимента программная реализация алгоритма сохраняет число основных операций в файл для

дальнейшей обработки.

Рассмотрим более подробно этап планирования предварительного исследования [15]. Основной задачей этого этапа является определение рационального размера выборки при фиксированной длине входа для проведения экспериментальных исследований трудоемкости алгоритма в среднем.

Выдвигается гипотеза, что функция трудоемкости алгоритма Паллоттино имеет бета-распределение. Основные этапы предварительного исследования:

1. Фиксация некоторого значения длины входа n из реального сегмента длин в области применения алгоритма Паллоттино. В рассматриваемом случае $n = 80$.

2. Определение необходимого числа экспериментов m с программной реализацией для получения гистограммы относительных частот значений трудоемкости одним из описанных ранее способов.

3. Проведение экспериментального исследования и получение значений трудоемкости f_i для сгенерированных случайных допустимых входов D_i: $f_i = f_A(D_i), i = \overline{1, m}$.

4. Получение теоретических функций трудоемкости алгоритма для лучшего и худшего случаев, как функций длины входа. Для алгоритма Паллоттино эти функции имеют вид:

$$f_A^{\vee}(n) = n,$$

$$f_A^{\wedge}(n) = \frac{n^3 \cdot (n-1)}{2}. \tag{6}$$

   Последняя формула следует из того, что алгоритм Паллоттино в худшем случае имеет оценку (5), а поскольку для полных графов $l = \frac{n \cdot (n-1)}{2}$, получаем (6).

5. Выбор числа k полусегментов для гистограммы частот значений трудоемкости.

6. Нормирование значений экспериментальной трудоемкости и построение на основе полученных данных гистограммы относительных частот в полусегментах.

7. Вычисление нормированного выборочного среднего и нормированной исправленной выборочной дисперсии по формулам [15]:

$$\bar{t} = \frac{\overline{f_t}(n) - f^{\wedge}}{f^{\wedge} - f^{\vee}},$$

$$s^2 = \frac{1}{m-1} \sum_{i=1}^{m} \frac{(f_i - \overline{f_t}(n))^2}{(f^{\wedge} - f^{\vee})^2},$$

где $f^{\wedge}$ и $f^{\vee}$ — соответственно максимальное и минимальное значение теоретических функций трудоемкости, $\overline{f_t}(n)$ — выборочное среднее.

8. Формулировка гипотезы и расчет параметров аппроксимирующего закона распределения. В данном случае выдвигается гипотеза о бета-распределении. Параметры бета-распределения рассчитываются по формулам [15]:

$$\alpha = \frac{\bar{t}}{s^2}(\bar{t} - (\bar{t})^2 - s^2), \tag{7}$$

$$\beta = \frac{(1 - \bar{t})}{s^2}(\bar{t} - (\bar{t})^2 - s^2). \tag{8}$$

9. Расчет теоретических частот по функции плотности по формуле [15]:

$$p_i = \int_{x_i}^{x_i + \Delta x_i} b(x, \alpha, \beta) dx,$$

где b — функция бета-распределения.

10. Расчет наблюдаемого значения критерия согласия Пирсона по фор-

муле [24]:

$$\chi^2_{\text{набл}} = m \sum_{i=1}^{s} \frac{(w_i - p_i)^2}{p_i},$$

где w_i — относительные частоты.

11. Проверка гипотезы о законе распределения. Если нет оснований отвергнуть нулевую гипотезу, то переход к основному этапу исследования. В противном случае — выбор другого закона распределения и повторная проверка гипотезы.

3.2.2 Результаты предварительного исследования

Для оценки необходимого числа экспериментов с программной реализацией алгоритма Паллоттино для фиксированной длины входа ($n = 80$) в соответствие с описанной методикой был проведен этап предварительного исследования с коэффициентом доверия $\gamma = 0.95$ и относительной ошибкой $\varepsilon = 0.001$. Для определения необходимого числа экспериментов использовался метод на основе бета-распределения [15]. Суть метода была изложена ранее.

Сначала была извлечена выборка объемом 200, вычислено значение $m^*_{(1)}$, результаты расчетов приведены в таблице 1.

Таблица 1.

Предварительный объем выборки	200
Выборочное среднее	9372.87
Выборочная дисперсия	2774677.39005
Нормированное среднее	0.000459499
Нормированная дисперсия	0.000000006784
Альфа	31.10865322
Бета	67670.14123
Дельта для $\varepsilon = 0.001$	9.401506539
Нижний предел интегрирования	0.000459034
Верхний предел интегрирования	0.000459964
Рассчитанный объем выборки	84934

Далее была извлечена выборка объемом 84934, результаты ее обработки приведены в таблице 2. Поскольку рассчитанный объем выборки оказался больше, чем объем выборки текущего эксперимента, то $m^*_{(1)} = 103725$.

Таблица 2.

Предварительный объем выборки	84934
Выборочное среднее	9524.105
Выборочная дисперсия	3498806.02004
Нормированное среднее	0.000466977
Нормированная дисперсия	0.000000008554
Альфа	25.47950063
Бета	54537.17062
Дельта для $\varepsilon = 0.001$	8.832817303
Нижний предел интегрирования	0.00046654
Верхний предел интегрирования	0.000467414
Рассчитанный объем выборки	103725

Далее была извлечена выборка объемом 103725, результаты ее обработки приведены в таблице 3. Поскольку рассчитанный объем выборки

оказался больше, чем объем выборки текущего эксперимента, то $m^*_{(1)} = 104740$.

Таблица 3.

Предварительный объем выборки	148178
Выборочное среднее	9531.38
Выборочная дисперсия	3538446.02776
Нормированное среднее	0.000467337
Нормированная дисперсия	0.000000008651
Альфа	25.23289925
Бета	53967.72266
Дельта для $\varepsilon = 0.001$	8.037933252
Нижний предел интегрирования	0.000466939
Верхний предел интегрирования	0.000467734
Рассчитанный объем выборки	104740

Далее была извлечена выборка объемом 104740, результаты ее обработки приведены в таблице 4. Поскольку рассчитанный объем выборки оказался меньше, чем объем выборки текущего эксперимента, то $m = 104435$.

Таблица 4.

Предварительный объем выборки	104435
Выборочное среднее	9526.48
Выборочная дисперсия	3524490.40991
Нормированное среднее	0.000467095
Нормированная дисперсия	0.000000008617
Альфа	25.30655136
Бета	54153.34542
Дельта для $\varepsilon = 0.001$	7.983101972
Нижний предел интегрирования	0.0004667
Верхний предел интегрирования	0.000467489
Рассчитанный объем выборки	104435

Далее была извлечена итоговая выборка объемом 104435 и построена гистограмма относительных частот на 387 полусегментах. Данное значение для количества полусегментов было получено с помощью эмпирического метода, чтобы разбить гистограмму на достаточное количество интервалов. Нормирование значений экспериментальной трудоемкости и построение на основе полученных данных гистограммы в полусегментах представлены на рис. 1. Значения параметров бета-распределения для итоговой выборки: $\alpha = 25.24821, \beta = 53988.69414$.

Результаты расчетов теоретических частот по функции плотности приведены на рис. 1. Наблюдаемое значения критерия Пирсона в данном случае $\chi^2_{\text{набл}} = 282.24386$. Поскольку $\chi^2_{\text{кр}}(0.05, 384) = 430.69192$ (вычислено стандартной функцией пакета Microsoft Excel), получаем $\chi^2_{\text{набл}} < \chi^2_{\text{кр}}(0.05, 384)$. Следовательно, нет оснований отвергать нулевую гипотезу и можно перейти к основному этапу исследования.

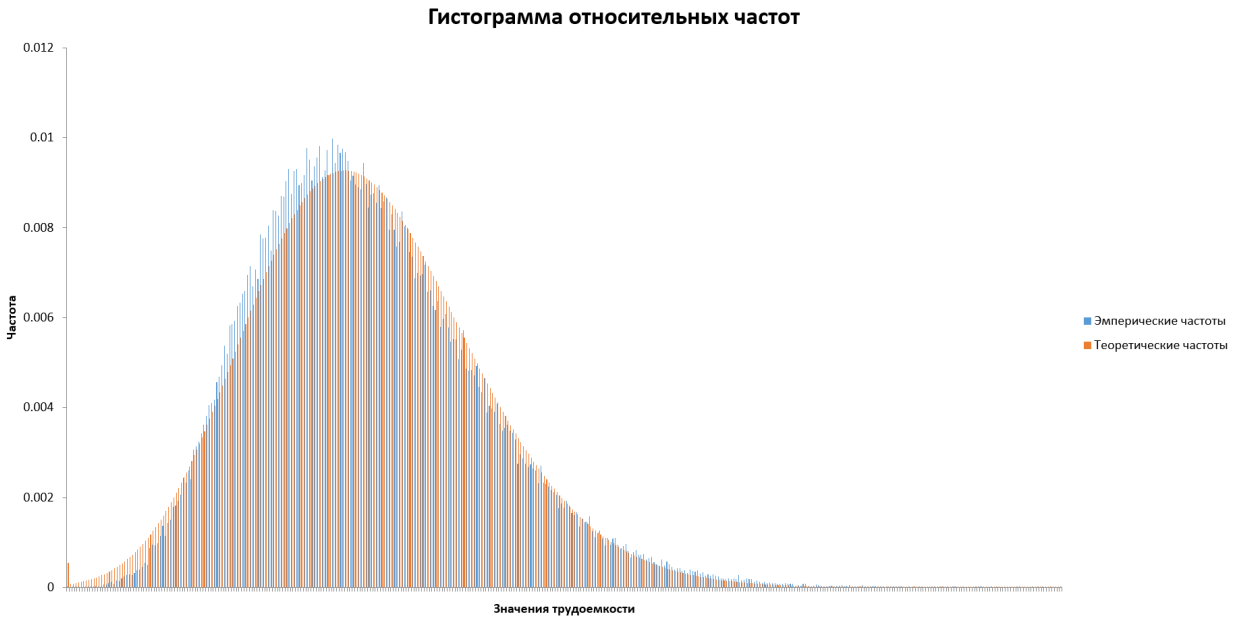

Рис. 1. Теоретические и эмпирические частоты для алгоритма Паллоттино при $n = 80$ с разбиением нормированного сегмента [0, 1] на 387 полусегментов

3.3 Этап основного исследования

3.3.1 Основные этапы

1. Определение сегмента значений длин входа, соответствующего особенностям применения данного алгоритма в разрабатываемой программной системе. В данном случае алгоритма Паллоттино будет применяться для массивов длиной от 80 до 2560;

2. Определение сегмента значений длин входа, для которого будут проводиться экспериментальные исследования. В данном примере таким сегментом является сегмент от 80 до 320;

3. Выбор шага изменения длины входа в экспериментальном исследовании. В данном случае значение шага равно 10;

4. Выбор необходимого числа m экспериментов с программной реализацией алгоритма для фиксированной длины входа для определения выборочной средней и дисперсии. В данном случае по результатам предварительного исследования $m = 104435$;

5. Расчет на основе экспериментальных данных значений выборочной средней и дисперсии для каждого значения n. В данном случае n изменяется от 80 до 320 с шагом 10;

6. Анализ экспериментальных данных — построение уравнения регрессии для выборочной средней и выборочной дисперсии;

7. Расчет на основе полученных результатов параметров аппроксимирующего бета-распределения по формулам (7), (8) как функций длины входа $\alpha(n), \beta(n)$;

8. Выбор значения доверительной вероятности и вычисление значений левого γ-квантиля бета-распределения [18]: $x_\gamma(n) = B^{-1}(\gamma, \alpha(n), \beta(n))$;

9. Вычисление значений функции доверительной трудоемкости для ис-

следуемого сегмента длин входа по формуле [1]:

$$f_\gamma(n) = f^\vee(n) + x_\gamma(n)(f^\wedge(n) - f^\vee(n)).$$

3.3.2 Результаты основного исследования

Было проведено основное исследование с выборками объемом 104435 для входных данных в сегменте $n = [80, 2560]$. Результаты представлены на рисунках 2–7.

Результаты построения уравнения регрессии для нормированной выборочной средней и выборочной дисперсии показаны на рис. 2 и рис. 3 соответственно. В первом случае уравнение регрессии имеет вид $\bar{t} = 0.9449 \cdot n^{-1.1735}$, во втором — $s^2 = y = 0.1165 \cdot n^{-3.741}$. В данном случае $y = ax^{-b}$ — наилучший в смысле максимума значения коэффициента детерминации R^2 вид функции, подбор коэффициентов осуществлен с помощью метода наименьших квадратов (расчеты выполнены в Microsoft Excel).

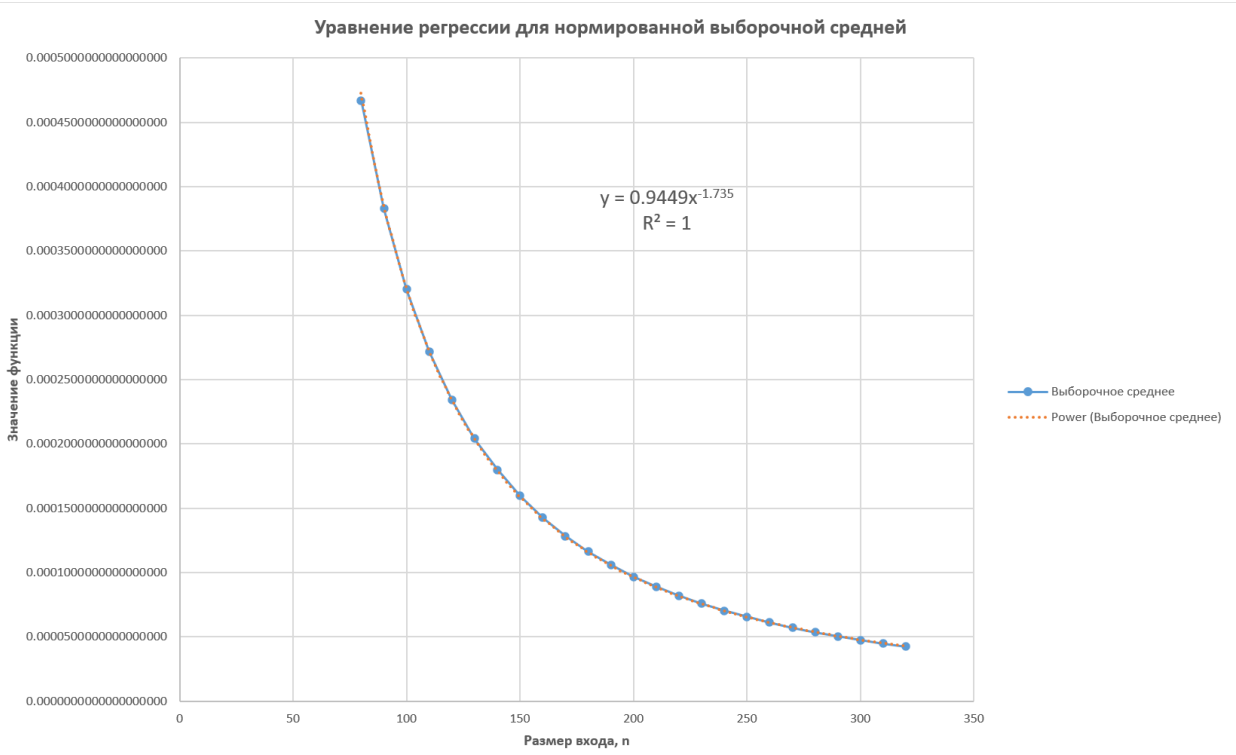

Рис. 2. Экспериментальные данные и уравнение регрессии для нормированной выборочной средней значений трудоемкости алгоритма Паллоттино

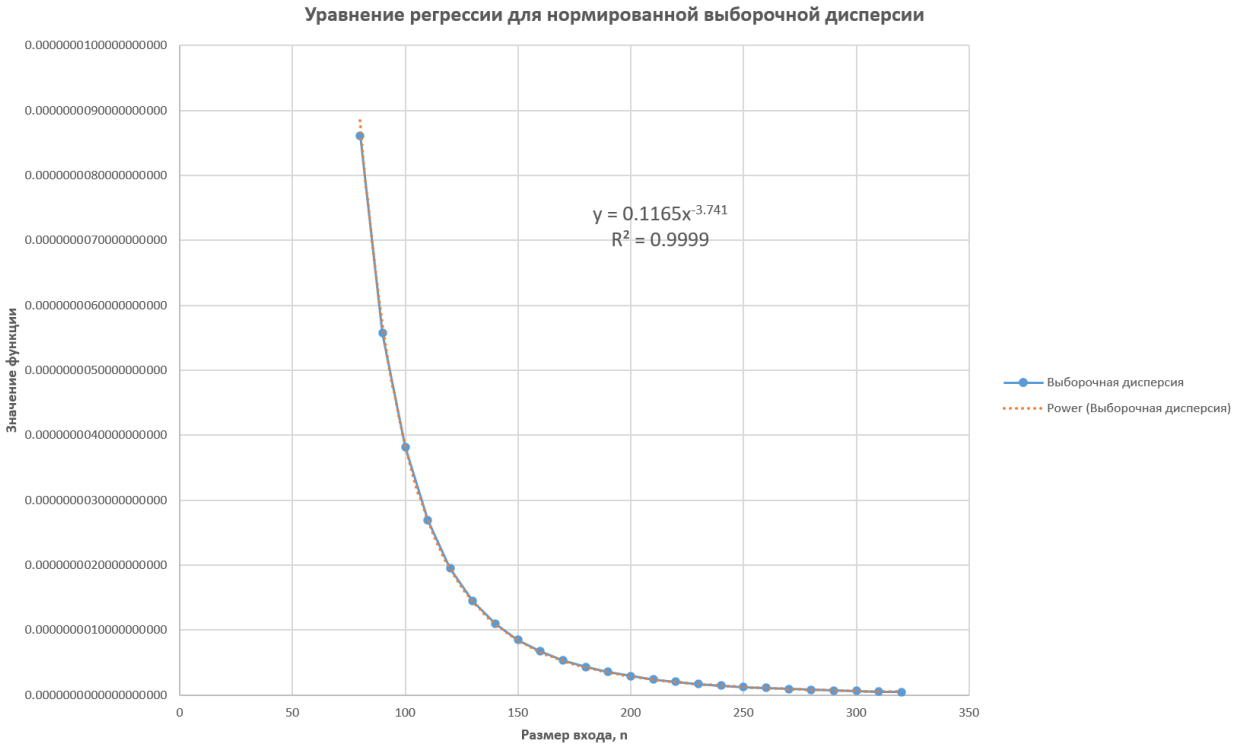

Рис. 3. Экспериментальные данные и уравнение регрессии для нормированной выборочной дисперсии значений трудоемкости алгоритма Паллоттино

График функции $\alpha(n)$ показан на рис. 4, график функции $\beta(n)$ показан на рис. 5.

В рассматриваемом примере $\gamma = 0.95, \alpha < \beta$, график значений $x_\gamma(n)$ показан на рис. 6. На рис. 7 показан график значений доверительной трудоемкости и трудоемкости в худшем случае для алгоритма Паллоттино на сегменте $n = [80, 2560]$. Отдельно отметим, что доверительная трудоемкость получена для значения доверительной вероятности $\gamma = 0.95$, т. е. в 95% случаев наблюдаемая в единичном эксперименте трудоемкость алгоритма не будет превышать значение доверительной трудоемкости. Для рассматриваемого примера эти значения в разы меньше трудоемкости в худшем случае на исследуемом сегменте длин входа.

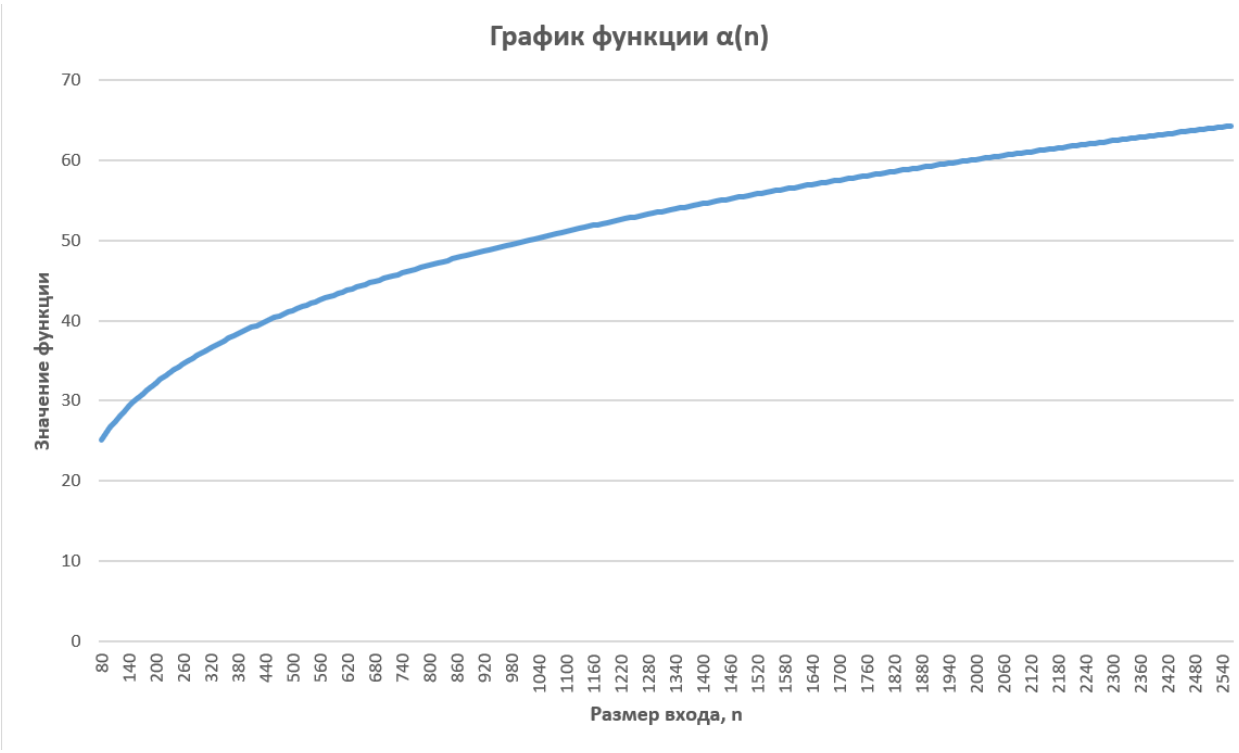

Рис. 4. График функции $\alpha(n)$ — параметра α аппроксимирующего бета-распределения для алгоритма Паллоттино

Рис. 5. График функции $\beta(n)$ — параметра β аппроксимирующего бета-распределения для алгоритма Паллоттино

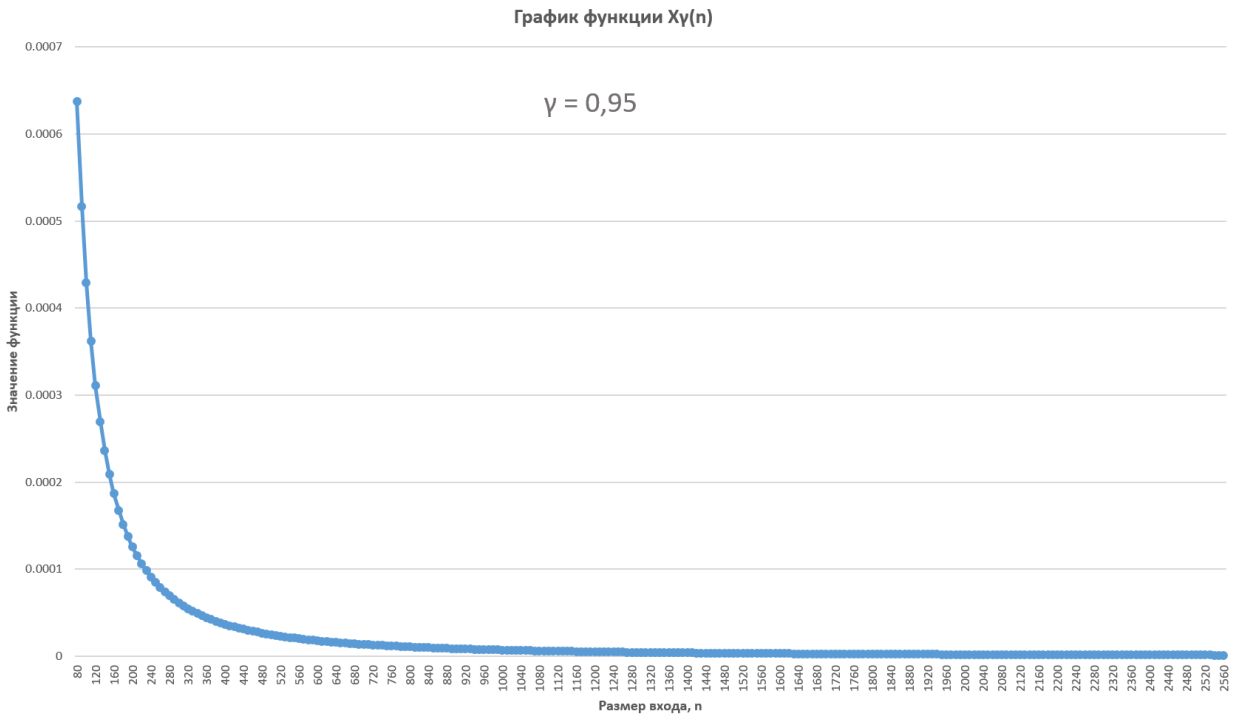

Рис. 6. График зависимости левого γ-квантиля бета-распределения $x_\gamma(n)$ от длины входа для алгоритма Паллоттино

Рис. 7. График доверительной трудоемкости и трудоемкости в худшем случае для алгоритма Паллоттино

Глава 4. Создание инструментария

Реализация инструментария включает разработку desktop-приложения на языке программирования C# с функцией формирования отчетов в виде документа для программного пакета Microsoft Excel.

4.1 Постановка задачи

В соответствии с описанным ранее критерием оценки по доверительной трудоемкости необходимо создать автоматизированную систему для проведения следующих этапов исследования программной реализации алгоритмов:

- предварительный этап, целью которого является проверка гипотезы о виде закона распределения значений трудоемкости алгоритма как дискретной ограниченной случайной величины [15];
- основной этап, в ходе которого значения доверительной трудоемкости вычисляются в зависимости от длины входных данных алгоритма [1];
- этап обработки результатов анализа;
- оформление результатов анализа алгоритма в виде отчета, включающего итоговые значения требуемых вычислительных ресурсов для входных данных определенной размерности.

Необходимо отметить, что предполагается наличие реализации компьютерного алгоритма, которая используется для проведения анализа. Также модуль с реализацией должен содержать компонент для генерации входных данных.

4.2 Архитектура системы

Одной из главных задач является проектирование системы с учетом поддержки наибольшего количества программных реализаций алгоритмов для проведения анализа.

Для выполнения этого требования программный модуль должен иметь определенный интерфейс, т. е. принимать аргументы из командной строки. Они требуются для проведения экспериментальных исследований трудоемкости в различных конфигурациях алгоритма.

В процессе разработки использованы следующие технологии: Windows Presentation Foundation (WPF) для реализации пользовательского интерфейса, EPPlus для формирования отчета в виде документа пакета Microsoft Excel [38]. Такой стек технологий позволяет вести разработку и поддержку специалисту, владеющему только одним языком программирования. Схема архитектуры прототипа представлена на рис. 8.

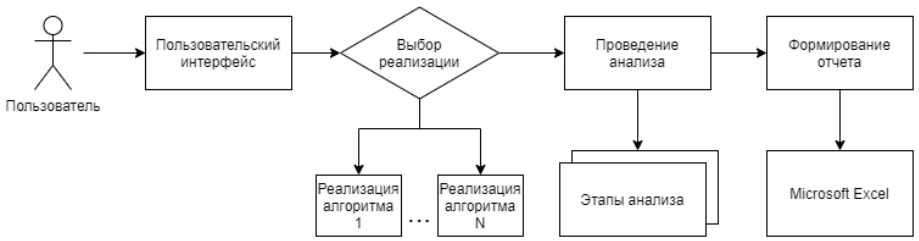

Рис. 8. Схема архитектуры системы

4.3 Описание реализации

На основе приведенной архитектуры разработано приложение на языке C# [39], позволяющее проводить анализ алгоритмов и получать результаты в виде отчета для изучения их применимости на заданных входных данных.

В первую очередь, нужно получить программную реализацию алгоритма. Язык программирования или технологии, используемые для получения реализации, не имеют значения. Единственное требование — возможность принимать входные параметры для проведения экспериментальных исследований. В случае отсутствия возможности принимать параметры система не сможет провести анализ для предоставленной программной реализации.

Система имеет предопределенный набор программных реализаций алгоритмов, которые можно использовать для анализа. Для добавления

новой реализации достаточно поместить разработанный модуль в каталог с установленной системой. При запуске система автоматически обнаружит новый модуль и добавит его в качестве одного из вариантов для проведения анализа.

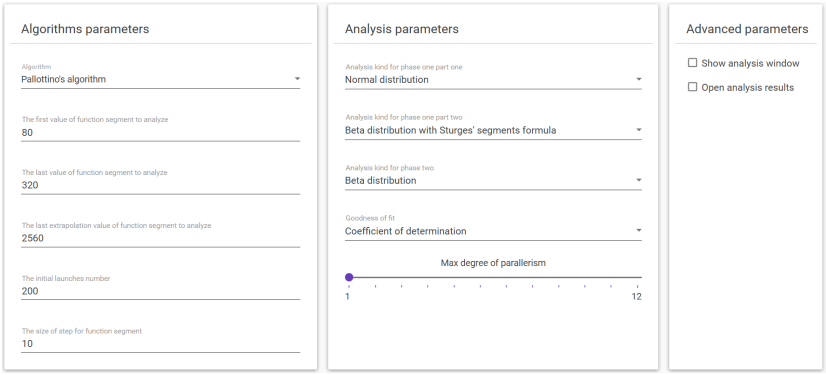

Рис. 9. Интерфейс настройки параметров анализа

Предусмотрена возможность настройки системы и выбора желаемых методов расчета количества полусегментов гистограммы относительных частот, размера выборки и т. д. Практически все описанные в работе методы присутствуют в системе и доступны для использования. Это позволяет исследователям проверять различные гипотезы при анализе функции трудоемкости для программных реализаций алгоритмов.

Параметры для экспериментальных исследований формирует система на основе пользовательского ввода и передает их программе в качестве аргументов командной строки при запуске. Пример набора параметров показан на рис. 9. Результаты многократного запуска исследуемой программной реализации сохраняются во временный файл, из которого потом загружаются в систему для проведения основного этапа исследования. По окончании анализа формируется отчет с подробным описанием полученных оценок эффективности (рис. 10).

	D	E	F	G	H	I	J
1	Карман	Частота	Эмперические частоты	Теоретические частоты	Значение для χ2	Дополнительные параметры	Значения
2	0.000240656	0	0	0.0018983	0.0018983	Размер входа	80
3	0.000274094	3	0.003	0.007083503	0.002354061	min f(n)	80
4	0.000307532	13	0.013	0.020979473	0.003034966	average f(n)	252800
5	0.00034097	47	0.047	0.046176928	1.46707E-05	max f(n)	20224000
6	0.000374408	80	0.08	0.079873185	2.01344E-07	Кол-во экспериментов	1000
7	0.000407846	150	0.15	0.113187104	0.011973001	Коэффициент доверия	0.95
8	0.000441285	146	0.146	0.135651011	0.000789538	Уровень значимости	0.05
9	0.000474723	134	0.134	0.140947837	0.000342484	Eps	0.001
10	0.000508161	105	0.105	0.129500432	0.004635283	Минимальное значение выборки	0.000240656
11	0.000541599	110	0.11	0.106900661	8.98582E-05	Максимальное значение выборки	0.00086902
12	0.000575037	77	0.077	0.080325006	0.000137637	Длина интервала	3.34382E-05
13	0.000608475	59	0.059	0.055536322	0.000216022	Число полусегментов для гистограммы	19
14	0.000641914	27	0.027	0.035652707	0.002099962	χ2 наблюдаемое	37.8084349
15	0.000675352	17	0.017	0.021415285	0.000910319	Кол-во степеней свободы	16
16	0.00070879	13	0.013	0.012114622	6.47065E-05	Критическое значение χ2	26.2962276
17	0.000742228	10	0.01	0.006490701	0.001897357	Проверка гитотезы функцией ТЕСТ	1
18	0.000775666	4	0.004	0.00330966	0.000143994		
19	0.000809104	1	0.001	0.001612977	0.000232949		
20	0.000842542	3	0.003	0.000754123	0.00668852		
21	0.000875981	1	0.001	0.000590165	0.000284607		

Рис. 10. Пример результатов анализа

Выводы

Поставленные задачи выполнены, созданный инструментарий позволяет проводить все этапы анализа функции трудоемкости и вычислять доверительную трудоемкость.

Поставленный эксперимент с алгоритмом Паллоттино показал, что оценка функции сложности в худшем случае может приводить к существенному завышению временного прогноза из-за малой вероятности входных данных, обеспечивающих максимум функции трудоемкости для рассматриваемой задачи. Полученные результаты подтверждают возможность повышения достоверности прогнозирования временной эффективности компьютерных алгоритмов и более эффективного решения задачи выбора рациональных алгоритмов на основе сравнительного анализа функций доверительной трудоемкости вместо традиционного сравнения трудоемкости в среднем случае.

Созданная автоматизированная система принимает на вход любые программные реализации алгоритмов, удовлетворяющих требуемому программному интерфейсу вне зависимости от выбранного языка программирования или используемых технологий разработки.

Отдельный интерес для дальнейшего исследования представляет рассмотрение алгоритмов класса NPR с функцией трудоемкости, зависящей от двух и более параметров входа. В данной работе в качестве объекта исследований был взят алгоритм Паллоттино с функцией трудоемкости от двух параметров входа (количество вершин и количество ребер). Чтобы использовать методологию, изложенную в [1], потребовалось зафиксировать второй параметр. Именно поэтому в качестве входных данных рассмотрены только полные графы, для которых можно легко рассчитать количество ребер при заданном количестве вершин. Однако не всегда можно выразить один параметр функции трудоемкости через другой. Таким образом, одним из потенциальных улучшений работы является исследование функции трудоемкости при вариации нескольких параметров, что позволит применять описанную методологию к более широкому классу алгоритмов.

Заключение

Исследованы различные дополнительные методы для применения методологии [1] на более широкий класс алгоритмов.

Разработана автоматизированная система для оценки качества алгоритмов по доверительной трудоемкости, вычисленной в процессе эмпирического анализа программной реализации. Преимущество системы состоит в получении гарантирующей оценки на основе статистических заключений с заданным уровнем значимости. Исходный код и документация системы находятся в публичном репозитории в GitHub [39].

Список литературы

[1] Петрушин В. Н., Ульянов М. В., Кривенцов А. С. Доверительная трудоемкость — новая оценка качества алгоритмов // Информационные технологии и вычислительные системы. 2009. № 2. С. 23–37.

[2] Трахтенброт Б. А. Сложность алгоритмов и вычислений. Новосибирск: Изд-во Новосибирского ун-та, 1967.

[3] Офман Ю. П. Об алгоритмической сложности дискретных функций // ДАН СССР. 1962. Т. 45, вып. 1. С. 48–51.

[4] Цейтин Г. С. Оценка числа шагов при применении нормального алгоритма // Математика в СССР за 40 лет. Т. 1. М., 1959. С. 44–45.

[5] Трахтенброт Б. А. Сигнализирующие функции и табличные операторы // Записки Пензенского ГПИ. 1956. Вып. 4. С. 75–87.

[6] Алферова З. В. Теория алгоритмов. М.: Статистика, 1973.

[7] Гашков С. Б., Чубариков В. Н. Арифметика. Алгоритмы. Сложность вычислений: Учеб. пособие для вузов / Под ред. В. А. Садовничего. 2-е изд., перераб. М.: Высш. шк., 2000.

[8] Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. Introduction to Algorithms. Chapter 1: Foundations (Second ed.) // Cambridge, MA: MIT Press and McGraw-Hill. 2001. P. 3–122.

[9] Knuth D. The Art of Computer Programming / Addison – Wesley, 1968.

[10] Wegener I. Complexity theory: exploring the limits of efficient algorithms // Berlin, New York: Springer – Verlag. P. 20.

[11] Juraj H. Theoretical computer science: introduction to Automata, computability, complexity, algorithmics, randomization, communication, and cryptography // Springer. 2004. P. 177–178.

[12] Berube P., Amaral J. N. Combined profiling: A methodology to capture varied program behavior across multiple inputs // ISPASS 2012 — IEEE International Symposium on Performance Analysis of Systems and Software. 2012. № 6189227. P. 210–220.

[13] Hutter F., Xu L. Hoos H., Leyton – Brown K. Algorithm runtime prediction: Methods & evaluation // Artificial Intelligence. 2014. Vol. 206. № 1. P. 79–111.

[14] Oprea M. A general framework and guidelines for benchmarking computational intelligence algorithms applied to forecasting problems derived from an application domain-oriented survey // Applied Soft Computing. 2020. Vol. 89. № 4. P. 106–103.

[15] Петрушин В. Н., Ульянов М. В. Планирование экспериментального исследования трудоемкости алгоритмов на основе бета-распределения // Информационные технологии и вычислительные системы. 2008. № 2. С. 81–91.

[16] Benzinger R. Automated complexity analysis of Nuprl extracted programs // Journal of Functional Programming. 2001. Vol. 11. № 1. P. 3–31.

[17] Hickey T., Cohen J. Automating program analysis // Journal of the ACM. 1988. Vol. 35. № 1. P. 185–220.

[18] Петрушин В. Н., Ульянов М. В. Информационная чувствительность компьютерных алгоритмов. М.: Физматлит, 2010.

[19] Гмурман В. Е. Теория вероятностей и математическая статистика: Учеб. пособие для вузов, 9-е изд., стер. М.: Высш. шк., 2003.

[20] Scott D. W. On optimal and data-based histograms // Biometrika. 1979. Vol. 66. P. 605–610.

[21] Freedman D., Diaconis P. On the histogram as a density estimator: 12 theory // Z. Wahrscheinlichkeit. 1981. Vol. 57. P. 453–476.

[22] Sturges H. The choice of a class-interval // J. Amer. Statist. Assoc. 1926. Vol. 21. P. 65–66.

[23] Hyndman R. J. The problem with Sturges' rule for constructing histograms // The Pennsylvania State University. 1995.

[24] Королюк В. С., Портенко Н. И., Скороход А. В., Турбин А. Ф. Справочник по теории вероятностей и математической статистике. М.: Наука, 1985.

[25] Прохоров Ю. В., Розанов Ю. А. Теория вероятностей (Основные понятия. Предельные теоремы. Случайные процессы). М.: Наука. Физматлит, 1987.

[26] Арсенин В. Я. Математическая физика: основные уравнения и специальные функции. М.: Наука, 1966.

[27] Смирнов Н. В., Дунин – Барковский И. В. Краткий курс математической статистики для технических приложений. М.: Физматгиз, 1959.

[28] Kac M., Kiefer J., Wolfowitz J. On Tests of Normality and Other Tests of Goodness of Fit Based on Distance Methods // Ann. Math. Stat., 1955. Vol. 26. P. 189–211.

[29] Hughes A., Grawoig D. Statistics: A Foundation for Analysis / Addison – Wesley, 1971.

[30] Dijkstra E. W. A note on two problems in connexion with graphs // Numer. Math. 1959. Vol. 1. P. 269–271.

[31] Bellman R. On a routing problem // Quart. Appl. Math. 1958. Vol. 16. P. 87–90.

[32] Ford L. R. and Fulkerson D. R. Flows in Networks // Princeton: Princeton University Press. 1962.

[33] Moore E. F. The shortest path through a maze // Bell Telephone System. Technical publications. 1959. Vol. 3523.

[34] Pollack M. and Wiebenson W. Solutions of the shortest-route problem – A review // Oper. Res. 8. 1960. P. 224–230.

[35] Pape U. Implementation and efficiency of Moore-algorithms for the shortest route problem // Math. Program. 1974. Vol. 7. P. 212–222.

[36] Левит Б. Ю., Лившиц В. Н. Нелинейные сетевые транспортные задачи // Институт комплексных транспортных проблем. М., Изд-во «Транспорт». 1972. С. 1–144.

[37] Pallottino S. Shortest-path methods: Complexity, interrelations and new propositions // Networks. 1984. Vol. 14. P. 257–267.

[38] Приложение для работы с электронными таблицами Microsoft Excel [Электронный ресурс]: `URL:https://products.office.com/ru-ru/excel` (дата обращения: 30.04.2020).

[39] Репозиторий проекта в системе контроля версий GitHub [Электронный ресурс]: `URL:https://github.com/Vasar007/algorithm_analysis` (дата обращения: 30.04.2020).