САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Яковлев Владимир Сергеевич

Выпускная квалификационная работа бакалавра

Эмпирический анализ трудоемкости алгоритма и средств программной реализации

Направление 01.03.02

Прикладная математика и информатика

ООП СВ.5005.2016 Прикладная математика, фундаментальная информатика и программирование

Научный руководитель:

кандидат физ.-мат. наук, доцент

кафедры моделирования электромеханических

и компьютерных систем

Никифоров Константин Аркадьевич

Рецензент:

кандидат физ.-мат. наук, доцент

кафедры радиофизики и электронных систем

ФГАОУ ВО «Северо-Восточный

федеральный университет имени М.К.Аммосова»

Антонов Степан Романович

Санкт-Петербург

2020

Содержание

Введение

Актуальность темы исследования обусловлена тем, что существует огромное количество различных компьютерных алгоритмов, в том числе и для решения одной и той же задачи, для которой необходимо проводить анализ их эффективности с последующим выбором наиболее подходящего алгоритма в той или иной области применения. На вход алгоритм может принимать различные наборы данных и разные реализации могут хорошо и быстро работать с одними наборами, но очень плохо с другими. Проведение оценки эффективности позволяет определить, что именно будет работать лучше в конкретных условиях.

Целью работы является построение доверительных интервалов оцениваемой величины трудоемкости компьютерного алгоритма с заданной доверительной вероятностью.

Постановка задачи

Для достижения указанной цели работы необходимо выполнить следующие задачи:

1) Провести литературный обзор, определить понятие доверительной трудоемкости алгоритма и установить ее связь со временем выполнения программной реализации или с количеством базовых операций, выполняемых программой.
2) Выбрать алгоритм и соответствующую программную реализацию для выполнения эмпирического анализа по указанной методике.
3) Выполнить предварительный этап исследования с проверкой гипотезы о законе распределения значений трудоемкости алгоритма как дискретной ограниченной случайной величины.
4) Выполнить основной этап исследования с определением значений доверительной трудоемкости как функции длины входа алгоритма

В данной работе будет описан процесс проведения оценки качества алгоритма с помощью доверительной трудоёмкости. Сначала будет описана методология, которая будет применена для анализа алгоритма, а также будет представлено описание самого алгоритма.

В качестве образца для исследования будет использована реализация метода триангуляции Делоне для выпуклых и невыпуклых многоугольников.

Программная реализация метода и сбор данных выполнен на языке программирования C#.

Необходимо обеспечить генерацию множества случайно построенных многоугольников с достаточно большим количеством вершин и собрать данные о том, сколько времени заняла триангуляция каждого из них.

Затем, по полученным данным будет проведена оценка качества выбранного алгоритма.

Обзор литературы

Время работы (или количество операций), выполняемых при запуске программной реализации, связано со сложностью алгоритма, положенного в ее основу, поэтому оценка времени работы (или количества операций) программы позволяет судить о временной сложности и ресурсной эффективности соответствующего алгоритма. Повысить точность результатов эмпирического анализа, который в обычном случае позволяет получить лишь точечные оценки среднего времени выполнения программы и лежащего в ее основе алгоритма [1] можно с помощью классического подхода математической статистики, связанного с построением доверительных интервалов оцениваемой величины трудоемкости с заданной доверительной вероятностью. Такой подход приводит к понятию доверительной трудоемкости [2].

Анализ алгоритма в рамках выбранного подхода подразумевает выполнение исследования в два этапа ─ предварительный и основной [2].

Задача предварительного этапа исследования состоит в проверке гипотезы о законе распределения значений трудоемкости алгоритма как дискретной ограниченной случайной величины.

Задача основного этапа исследования состоит в определении значений доверительной трудоемкости как функции длины входа алгоритма.

Алгоритм построения триангуляция Делоне широко используется в различных приложениях. Например, получаемые с помощью алгоритма сеточные элементы применяются в численных методах (метод конечных элементов, метод граничных элементов, реализованные в COMSOL, MATLAB и т.д.) основанных на дискретизации исходной вычислительной области на небольшие компоненты. В книге [1] есть ссылка на тестовую задачу ─ Алгоритм оптимальной триангуляции многоугольника. Этот алгоритм и соответствующие программные библиотеки описаны в книге [3] с различными подходами динамического программирования и необходимо

выбрать подходящую программную реализацию для проведения эмпирического анализа.

Для рассмотрения более точного понятия триангуляции Делоне использована книга [4]. В статье [5] были представлены примеры обыкновенной триангуляции и триангуляции Делоне, которые были задействованы в работе.

Для построения триангуляции Делоне применяется реализация, предложенная в [6], которая представляет из себя перенос на C# пакета [7].

Генерация произвольных многоугольников с заданным количеством вершин выполнена на основе алгоритма, предложенного в статье [8].

Для исследования трудоемкости алгоритма на основе бета-распределения были использованы методы, предложенные в работе [9].

Глава 1. Методика исследования

1.1 Функция трудоемкости алгоритма

Важной составляющей для комплексной оценки алгоритма являются его ресурсные характеристики – временная и емкостная эффективности. Наибольший интерес для нас представляет сам алгоритм, а не его программная реализация, поэтому наиболее грамотно было бы определить базовые операции, которые и показывали бы операционные затраты. Это позволило бы избежать строгой привязки скорости работы алгоритма к конкретному окружению, например, к вычислительным возможностям компьютера.

Под трудоёмкостью алгоритма A на входе D в дальнейшем будем понимать количество действий, которое будет затрачено на совершение операции. При рассмотрении ряда алгоритмов можно заметить, что не всегда трудоемкость алгоритма на одном входе D длины n будет совпадать с трудоемкостью на другом входе такой же длины. Рассмотрим допустимые входы алгоритма длины n – существует подмножество множества D_A, которое включает все входы длины n, обозначим его через D_n [2].

Для разных входов трудоемкость может быть разной, поэтому будем использовать следующие обозначения для числа операций, задаваемых алгоритмом A на входах длины n как функций длины входа:

$f_A^{\wedge}(n)$ — худший случай – было затрачено наибольшее время.

$f_A^{\vee}(n)$ — лучший случай – было затрачено наименьшее время.

$\overline{f_A}(n)$ — средний случай – среднее время работы алгоритма.

1.2 Трудоемкость – дискретная ограниченная случайная величина

Часто используемая в анализе алгоритмов оценка трудоемкости в среднем (математическое ожидание) некорректна для оценки единичных входов, т. К. возможно наблюдение любого значения трудоемкости в

теоретическом диапазоне с определенными, не равными нулю, вероятностями.

Вариант оценки по моде также некорректен – знание, что именно это значение трудоемкости встречается наиболее часто, не есть гарантия того, что это значение мы будем наблюдать в конкретном эксперименте. Более того, для несимметричных распределений мода, медиана и математическое ожидание в общем случае не совпадают и могут значительно различаться.

Таким образом, точечные оценки трудоемкости как дискретной ограниченной случайной величины – мода, медиана и математическое ожидание не могут быть использованы как гарантирующие оценки, а очевидно гарантирующая оценка по максимуму – теоретическая трудоемкость в худшем случае – дает слишком завышенные временные прогнозы.

В силу вышесказанного актуальной является задача построения оценки трудоемкости, опирающейся на рассмотрение функции трудоемкости как дискретной ограниченной случайной величины, которая дает практически значимую интервальную оценку для конкретных входов алгоритма [2].

Для экспериментального исследования алгоритма с целью определить частотную встречаемость значений трудоемкости, необходимо провести ряд экспериментов с программной реализацией при фиксированной длине входа. Интервал между худшим и лучшим случаем разбивается на выбранное число полусегментов. Далее определяется, как часто каждая трудоемкость попала в тот или иной полусегмент и строится гистограмма относительных частот.

1.3 Аппроксимация гистограммы относительных частот трудоемкости бета-распределением

Проблема вероятностного подхода заключается в сложности теоретического доказательства того факта, что значения трудоемкости имеют определенный вид распределения.

Необходимо рассматривать функции распределения, ограниченные на

сегменте, т.к. значения функции трудоемкости ограничены при фиксированной длине входа: $f_A^{\vee} \le f_A \le f_A^{\wedge}$. Причем из-за достаточно большого числа различных значений трудоемкости можно перейти к непрерывным распределениям. Будем использовать аппарат бета-распределения, который описывает непрерывную случайную величину, имеющую ограниченный размах варьирования.

Плотность распределения вероятностей для бета-распределения задается функцией:

$$b(x,\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)*\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}, x \in [0,1], \quad (1)$$

где Γ – гамма функция Эйлера, а α и β – параметры функции плотности бета-распределения [2].

1.4 Понятие доверительной трудоемкости

Необходимо построить такую интервальную оценку трудоемкости алгоритма, которая будет более практична, чем теоретически определенный сегмент варьирования трудоемкости между лучшим и худшим случаями при фиксированной длине входа. Заметим, что ближе к худшему случаю встречается гораздо меньше результатов, а ведь наиболее важно ограничить значения трудоемкости сверху.

Решение этой проблемы основано на подходе математической статистики, и связано с построением доверительных интервалов оцениваемых величин с заданной доверительной вероятностью.

1.5 Методика исследования на доверительную трудоемкость

Все этапы анализа, указанные ниже, были представлены в статье [2].

Этап предварительного исследования:

1. Фиксация некоторой длины входа *n*.
2. Определение количества экспериментов *m* для адекватного построения гистограммы относительных частот.

3. Проведение вычислительного эксперимента и получение эмпирических значений трудоемкости.
4. Получение теоретических оценок функции трудоемкости алгоритма для лучшего и худшего случаев.
5. Выбор числа полусегментов для построения гистограммы частот эмпирических значений трудоемкости.
6. Нормирование эмпирических значений трудоемкости и построение гистограммы.
7. Вычисление выборочной средней и выборочной дисперсии.
8. Формулировка гипотезы о виде аппроксимирующего закона распределения и расчет его параметров.
9. Расчет теоретических частот по функции плотности вероятности.
10. Проверка гипотезы о законе аппроксимирующего распределения.

Этап основного исследования:

1. Определение диапазона длин входа с учетом применения данного алгоритма.
2. Определение диапазона длин входа для проведения вычислительного эксперимента.
3. Выбор шага по длине входа в вычислительном эксперименте.
4. Выбор необходимого числа экспериментов.
5. Расчет выборочной средней и дисперсии для каждого значения длины входа.
6. Построение уравнения регрессии для выборочной дисперсии.
7. Расчет параметров бета-распределения.
8. Выбор уровня доверительной вероятности и вычисление значений левого γ-квантиля аппроксимирующего бета-распределения.
9. Вычисление значений доверительной трудоемкости.

Глава 2. Алгоритм триангуляции

2.1 Триангуляция Делоне

На множестве точек на плоскости задана триангуляция, если некоторые пары точек соединены ребром, любая конечная грань в получившемся графе образует треугольник, ребра не пересекаются, и граф максимален по количеству ребер [5].

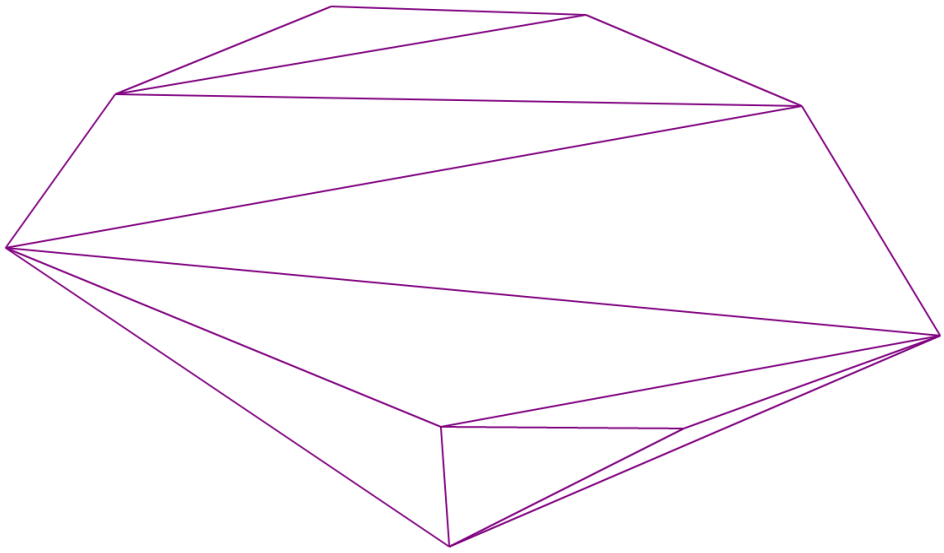

Рис. 1. Триангуляция выпуклого многоугольника

Триангуляцией Делоне называют такую триангуляцию, в которой для любого треугольника верно, что внутри описанной около него окружности не находится точек из исходного множества [5].

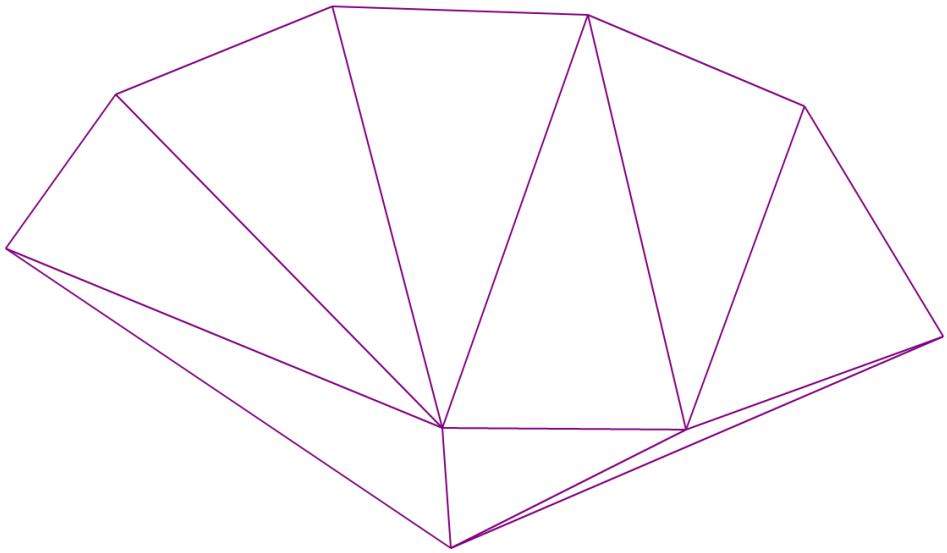

Рис. 2. Триангуляция Делоне выпуклого многоугольника

Данная триангуляция была впервые описана Борисом Делоне в 1934 году.

Трудоёмкость этой задачи составляет $O(N\ logN)$. Существуют алгоритмы, позволяющие достичь данной оценки в среднем и худшем случаях [4].

В данной работе применятся библиотека Triangle.NET [6], которая является портом программного обеспечения Triangle Джонатана Шевчука [7]. Применяется алгоритм “Разделяй и властвуй”, который позволяет достичь трудоемкости $O(N\ logN)$ в худшем и среднем случаях. В данном алгоритме множество точек разбивается на две как можно более равные части с помощью горизонтальных и вертикальных линий [4]. Он рекурсивно применяется к подчастям, а затем производится слияние полученных подтриангуляций [4].

Данный алгоритм работает и для невыпуклых многоугольников.

2.2 Входные данные, единицы измерения

На вход триангуляция будет принимать многоугольник, для которого сначала определяются вершины, а потом последовательно соединяются.

Выбор единиц измерения трудоемкости является важной задачей при

анализе алгоритма. Неверный выбор может привести к неправильным результатам и изменить итоговый вид функции трудоемкости.

В приведенном ниже иллюстративном примере в роли единицы измерения выступает число проходов по сортируемому массиву.

Для данного алгоритма в качестве единиц измерения трудоемкости используется количество образованных треугольников при триангуляции исходного набора данных.

2.3 Генерация входных данных

Для генерации входных данных применятся алгоритм предложенный в статье [8]. Многоугольник генерируется внутри некоторого заданного прямоугольника и возвращает массив вершин. Сгенерирован может быть как выпуклый, так и невыпуклый многоугольник, длина входа соответствует количеству вершин.

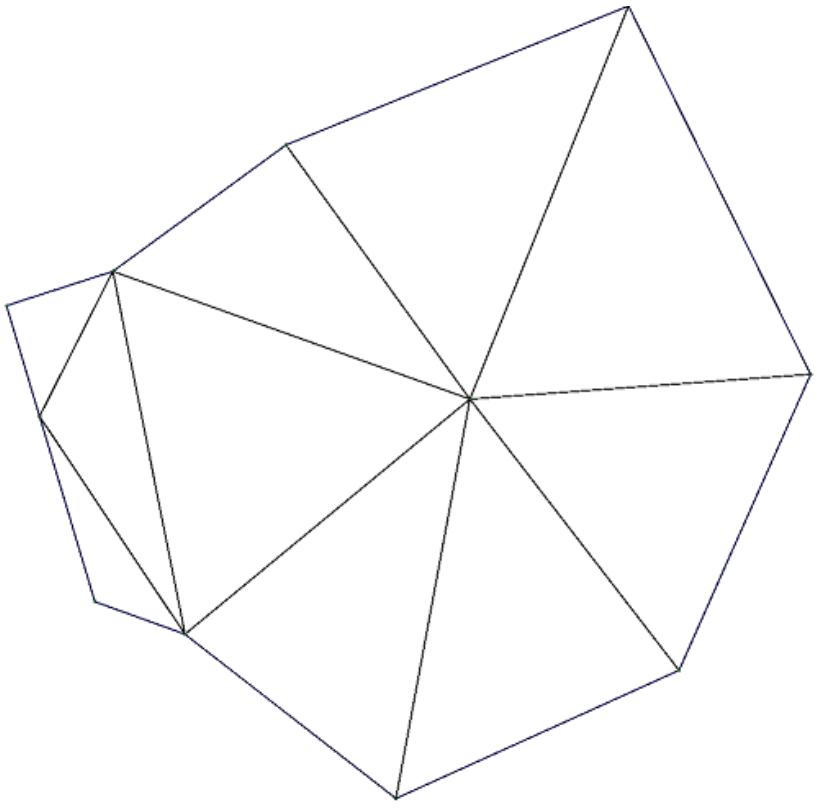

Рис. 3. Пример триангуляции сгенерированного невыпуклого многоугольника

Глава 3. Эмпирический анализ алгоритма пузырьковой сортировки

В качестве иллюстративного примера исследования доверительной трудоёмкости проведём эмпирический анализ алгоритма пузырьковой сортировки.

3.1 Расчёт объёма выборки

В данном случае объём выборки был посчитан на основе нормального распределения.

1. Длина массива составляла 100, а количество экспериментов 200. На первом шаге, новый рассчитанный объём выборки составил 17922.

2. На втором шаге были получены данные для выборки размером 17922. Действуя аналогично, получаем, что рассчитанный объём выборки теперь составляет 17805. Так как он меньше предыдущего объёма, то расчёт объёма выборки следует завершить.

3.2 Предварительный этап

1. Величина входных данных массива равна n=100.

2. Количество испытаний m примем равным 17805.

3. Теоретические функции трудоемкости для худшего и лучшего случаев: $f_A^{\wedge}(n) = n^2$; $f_A^{\vee}(n) = n$.

4. Выбираем 53 полусегмента для гистограммы.

5. Строим гистограмму относительных частот в полусегментах.

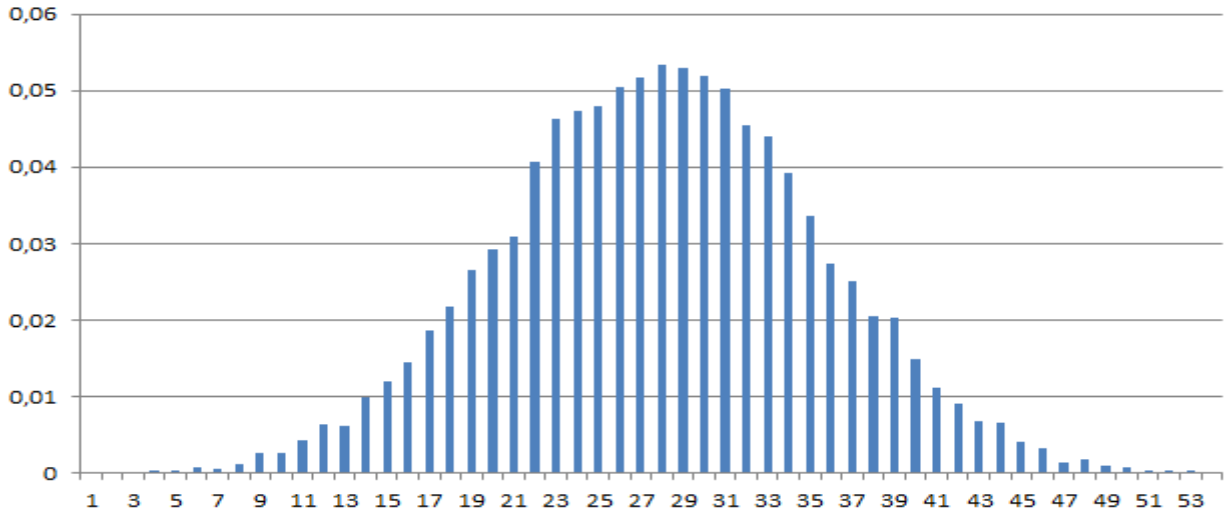

Рис. 4. Гистограмма относительных частот для алгоритма сортировки

6. Вычисляем выборочное среднее и выборочную дисперсию:

$$\bar{t} = \frac{1}{n}\sum_{i=0}^{n}\frac{f_i - f^{\vee}}{f^{\wedge} - f^{\vee}} \;;\; s^2 = \frac{1}{n-1}\sum_{i=0}^{n}\frac{(f_i - \bar{f}_э(n))^2}{(f^{\wedge} - f^{\vee})^2} \quad (2)$$

$$\bar{t} = 0{,}239847;\ s^2 = 0{,}000291$$

7. Вычисляем параметры бета-распределения:

$$\alpha = \frac{\bar{t}}{s^2}(\bar{t} - \bar{t}^2 - s^2) \;;\; \beta = \frac{1-\bar{t}}{s^2}(\bar{t} - \bar{t}^2 - s^2) \quad (3)$$

$$\alpha = 149{,}9948;\ \beta = 475{,}3816$$

8. Находим теоретические частоты.

9. Проверяем гипотезу с помощью встроенной в Excel функцией ХИ2ТЕСТ. Получаем положительный результат, а значит нет оснований опровергнуть гипотезу и переходим к основному этапу.

3.3 Основной этап

1. Величина массива входных данных будет варьироваться от 100 до 630.

2. Вычисления проводятся для сегмента от 100 до 320.

3. Шаг изменения длины входа равен 10.

4. Количество испытаний m примем равным 17805.

5. Для каждого n найдем свои значения выборочной средней и

дисперсии.

6. Строим уравнение регрессии.

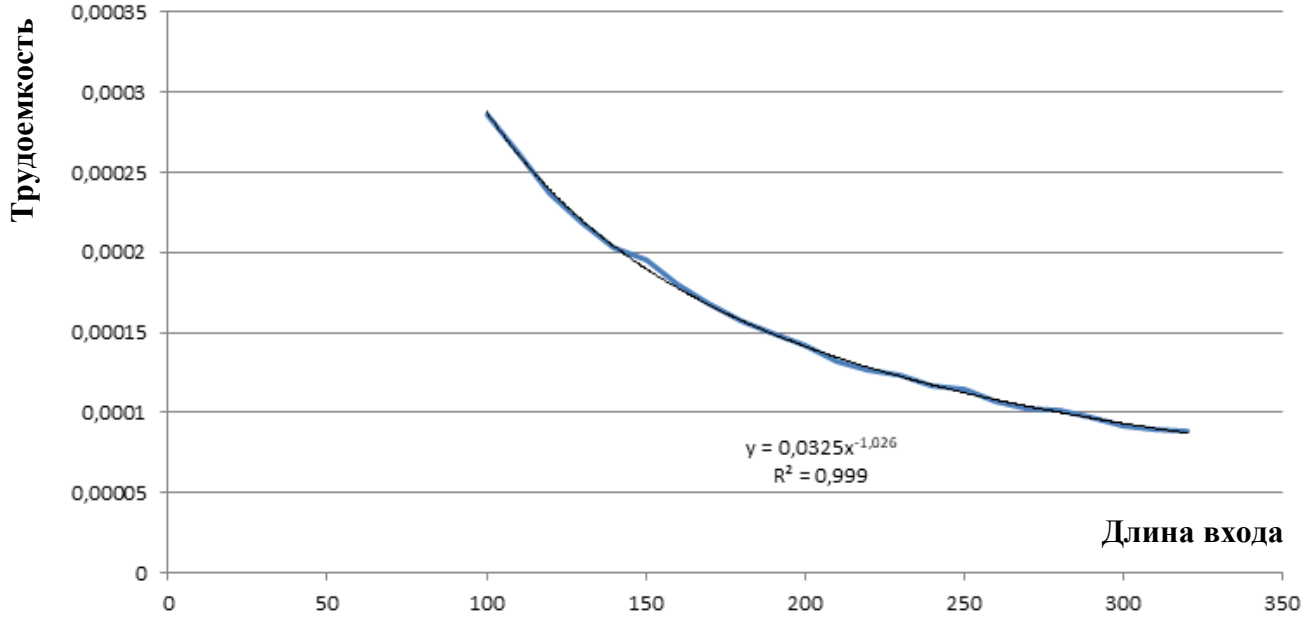

Рис. 5. Эмпирические значения и уравнение регрессии для выборочной дисперсии трудоёмкости алгоритма пузырьковой сортировки

7. Расчёт параметров аппроксимирующего бета-распределения как функций длины входа – $\alpha(n)$, $\beta(n)$.

8. Выбираем значения доверительной вероятности γ=0,95 и вычисляем левый квантиль бета-распределения

$$x_\gamma(n) = B^{-1}(\gamma, \alpha(n), \beta(n)) \quad (4)$$

9. Вычисляем значения функций доверительной трудоемкости по формуле

$$f_\gamma(n) = f^\vee(n) + x_\gamma(n) * (f^\wedge(n) - f^\vee(n)) \quad (5)$$

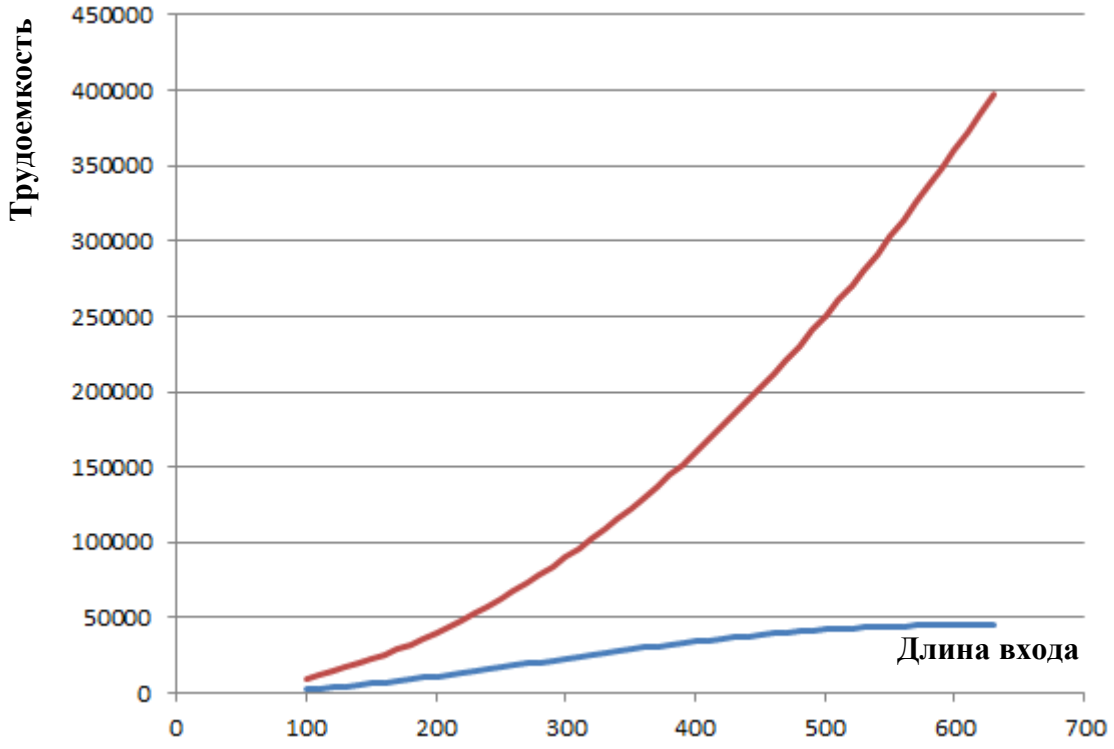

Рис. 6. График доверительной трудоёмкости (синий) и трудоёмкости в худшем случае (красный) для алгоритма пузырьковой сортировки

Таким образом, на Рис. 5 видно, как сильно будет отличаться трудоёмкость в худшем случае и доверительная трудоёмкость. В 95% случаев трудоёмкость алгоритма пузырьковой сортировки не будет превышать значение доверительной трудоёмкости.

Глава 4. Эмпирический анализ алгоритма триангуляции

4.1 Расчёт объёма выборки

1. Количество вершин многоугольника примем равным n=200.

2. Количество испытаний m примем равным 500.

3. Получив выборку, нормируем значения трудоёмкости.

4. Вычисляем выборочное среднее и выборочную дисперсию по формуле (2).

$$\overline{t} = 0{,}520615385;\ s^2 = 0{,}028021582$$

5. Вычисляем параметры бета-распределения по формуле (3):

$$\alpha = 5293{,}519;\ \beta = 20315{,}29$$

6. Вычисление пределов интегрирования:

$$\bar{t} \pm \delta_t = \frac{\bar{f}_э - f^\vee \pm \delta}{f^\wedge - f^\vee} \qquad (6)$$

$$\bar{t} + \delta_t = 0{,}206267712;\ \bar{t} - \delta_t = 0{,}207146242;$$

7. Решение уравнения $P(n) = \int_{\bar{t}-\delta_t}^{\bar{t}+\delta_t} B(x, u(n), v(n))dx$ до выполнения условия $P(n) \geq \gamma$.

8. Получаем новый объём выборки m=127.

9. Повторяем эксперимент с новым m.

10. Получаем новый объём выборки m=113. Так как он меньше, чем предыдущий, то дополнительные вычисления не требуются, и мы нашли оптимальный объём выборки.

4.2 Предварительный этап

1. Количество вершин многоугольника примем равным n=200.

2. Количество испытаний m примем равным 113, как было вычислено на предыдущем шаге.

3. Теоретические функции трудоемкости для худшего и лучшего

случаев: $f_A^{\wedge}(n) = n*log(n)$; $f_A^{\vee}(n) = n$

4. Выбираем 43 полусегментов для гистограммы.

5. Строим гистограмму относительных частот в полусегментах.

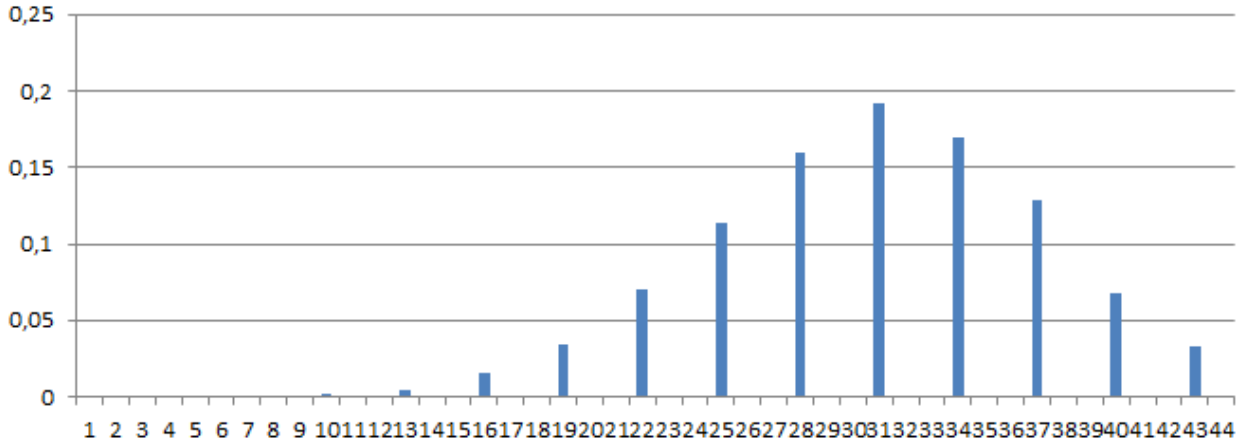

Рис. 7. Гистограмма относительных частот для алгоритма триангуляции

6. Вычисляем выборочное среднее и выборочную дисперсию по формуле (2).

$$\overline{t} = 0{,}206894423;\ s^2 = 0{,}000007067$$

7. Вычисляем параметры бета-распределения по формуле (3):

$$\alpha = 4803{,}381984;\ \beta = 18413{,}20317$$

8. Находим теоретические частоты.

9. Проверяем гипотезу с помощью встроенной в Excel функцией ХИ2ТЕСТ. Получаем положительный результат, а значит нет оснований опровергнуть гипотезу и переходим к основному этапу.

4.3 Основной этап

1. Величина массива входных данных будет варьироваться от 200 до 1000.

2. Вычисления проводятся для сегмента от 200 до 550.

3. Шаг изменения длины входа равен 50.

4. Количество испытаний m примем равным 113.

5. Для каждого n найдем свои значения выборочной средней и дисперсии.

6. Строим уравнение регрессии

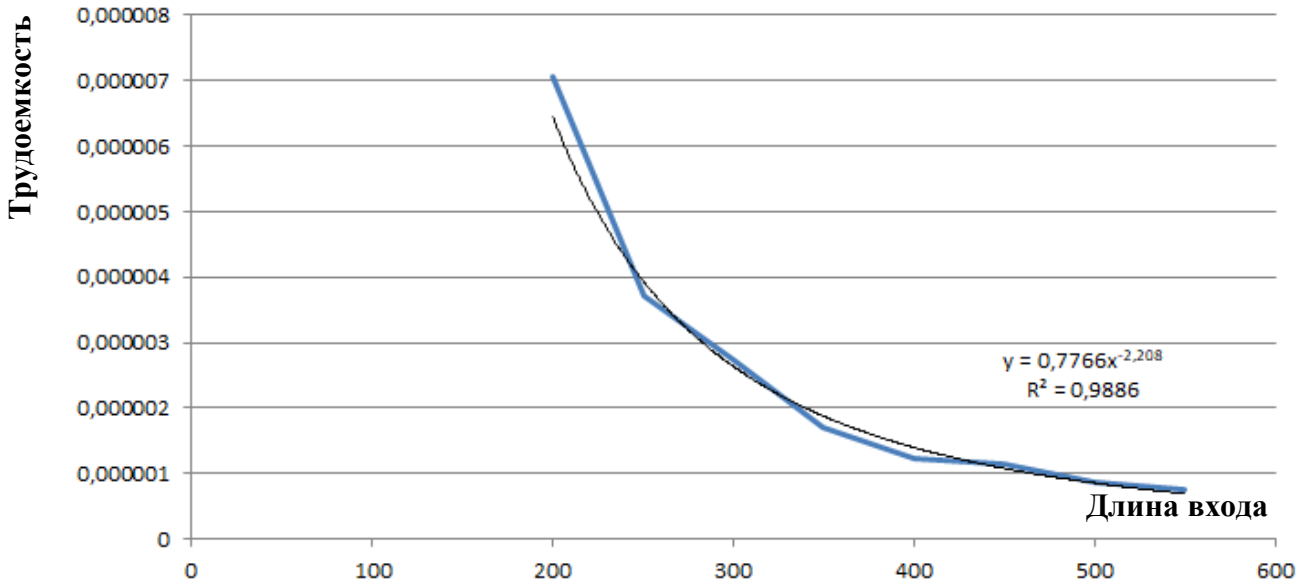

Рис. 8. Эмпирические значения и уравнение регрессии для выборочной дисперсии трудоёмкости алгоритма триангуляции

7. Расчёт параметров аппроксимирующего бета-распределения как функций длины входа – $\alpha(n)$, $\beta(n)$.

8. Выбираем значения доверительной вероятности γ=0,95 и вычисляем левый квантиль бета-распределения по формуле(4).

9. Вычисляем значения функций доверительной трудоемкости по формуле (5).

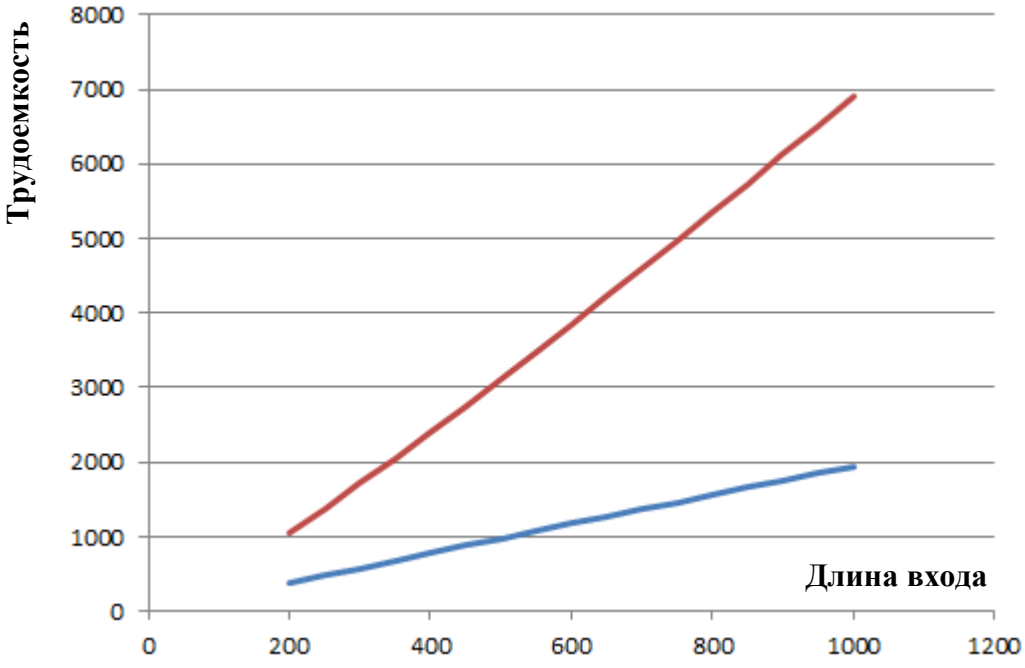

Рис. 9. График доверительной трудоёмкости(синий) и трудоёмкости в худшем

случае(красный) для алгоритма триангуляции

Таким образом, на Рис. 8 видно, как сильно будет отличаться трудоёмкость в худшем случае и доверительная трудоёмкость. В 95% случаев трудоёмкость алгоритма триангуляции не будет превышать значение доверительной трудоёмкости.

Заключение

Таким образом, в работе проведен эмпирический анализ алгоритма триангуляции Делоне, разработанного по методу “Разделяй и властвуй” с программной реализацией в библиотеке Triangle.NET [6], которая является портом программного обеспечения Triangle [7]. При этом эмпирический анализ проводился по методике, нацеленной на получение доверительной трудоемкости, которая позволяет значительно сузить оцениваемый сегмент варьирования трудоемкости. В качестве иллюстративного примера рассмотрено также применение данной методики к алгоритму пузырьковой сортировки.

В процессе выполнения предварительного этапа исследования с проверкой гипотезы о законе распределения значений трудоемкости алгоритма как дискретной ограниченной случайной величины была построена гистограмма относительных частот, которые наблюдаются в вычислительном эксперименте, и визуально оценен характер распределения. Гипотеза о возможности аппроксимации бета-распределением подтвердилась.

После выполнения основного этапа получено значительное отличие трудоёмкости в худшем случае и доверительной трудоёмкости, которая с коэффициентом доверия 95% не будет превышена.

Литература

[1] Левитин А.В. Алгоритмы: введение в разработку и анализ алгоритмов. М: Издательский дом Вильямс, 2006. 127 с.

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

[3] Стивен Скиена Алгоритмы. Руководство по разработке. 2011

[4] Скворцов А.И. Триангуляция Делоне и её применение. 2002

[5] Алгоритм триангуляции Делоне методом заметающей прямой https://habr.com/ru/post/445048

[6] Triangle.NET https://github.com/garykac/triangle.net

[7] Triangle https://www.cs.cmu.edu/~quake/triangle.html

[8] http://csharphelper.com/blog/2017/07/generate-random-polygons-in-c/

[9] http://www.isa.ru/jitcs/images/stories/2008/02/81_91.pdf