

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
"НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
"ВЫСШАЯ ШКОЛА ЭКОНОМИКИ"

Факультет компьютерных наук  
Департамент программной инженерии

**Отчет по курсовой работе**

на тему: "Исследование чувствительности по данным размера дерева  
решений в задаче коммивояжера"

по направлению подготовки "Программная инженерия"

Выполнил  
студент группы 301ПИ  
образовательной программы  
"Программная инженерия"  
М.И. Фомичёв

---

(Подпись, Дата)

Научный руководитель  
профессор департамента  
программной инженерии, д.т.н.  
М.В. Ульянов

---

(Оценка)

---

(Подпись, Дата)

Москва 2015

# РЕФЕРАТ

Отчет 57 с., 1 ч., 29 рис., 8 табл., 9 источников, 1 прил.

ЗАДАЧА КОММИВОЯЖЁРА, МЕТОД ВЕТВЕЙ И ГРАНИЦ, ДЕРЕВО РЕШЕНИЙ, ПОЛНЫЙ ГРАФ, ГАМИЛЬТОНОВ ЦИКЛ, ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ

Объектом исследования являются деревья решений в задаче коммивояжёра.

Основная цель работы - проанализировать структуру порожденного поискового дерева решений в процессе решения задачи коммивояжёра.

Дополнительная цель - определить влияние предвычисленного тура, рассчитанного с помощью жадного алгоритма, на время решения задачи коммивояжёра.

В процессе работы проводились экспериментальные исследования случайно сгенерированных полных графов.

Результаты исследования можно применять в задачах логистики и прикладного программирования, например, при реализации навигатора.

При дальнейшем исследовании задачи коммивояжёра и дерева решений, стоит обратить внимание на изменение времени поиска оптимального тура при наличии предвычисленного тура, полученного с помощью эмпирического алгоритма.

# СОДЕРЖАНИЕ

<b>Реферат</b>	<b>2</b>
<b>Определения</b>	<b>5</b>
<b>Введение</b>	<b>7</b>
<b>Основная часть</b>	<b>8</b>
1    Задача коммивояжёра . . . . .	8
1.1    Неформальная постановка задачи . . . . .	8
1.2    Обозначения . . . . .	8
1.3    Постановка в терминах теории графов . . . . .	9
1.4    Представление графа . . . . .	9
1.5    Виды задач . . . . .	9
1.6    Принадлежность к классам сложности . . . . .	10
2    Алгоритмы решения задачи коммивояжёра . . . . .	10
2.1    Полный перебор . . . . .	10
2.2    Метод ветвей и границ . . . . .	10
2.3    Метод ветвей и границ с предвычисленным туром . .	11
3    Методология экспериментального исследования . . . . .	11
4    Особенности реализации . . . . .	12
4.1    Структура хранения листьев поискового дерева реше- ний . . . . .	12
4.2    Язык программирования и версия компилятора . . . .	12
5    Результаты экспериментального исследования . . . . .	13
6    Графическая иллюстрация результатов экспериментального исследования . . . . .	26
6.1    Анализ количества порождённых вершин . . . . .	26
6.2    Анализ количества хранимых порождённых вершин .	32

6.3	Анализ максимального количества листьев дерева решений . . . . .	38
6.4	Анализ максимальной глубины дерева решений . . . . .	44
6.5	Графическое представление времени работы . . . . .	50
7	Выводы . . . . .	54
<b>Заключение</b>		<b>55</b>
<b>Список использованных источников</b>		<b>56</b>
<b>Приложение А</b>		<b>57</b>
8	Технические и программные особенности компьютера . . . . .	57

# ОПРЕДЕЛЕНИЯ

В настоящем отчёте о НИР используются следующие термины с соответствующими определениями:

- Граф – это конечное множество  $V$ , называемое множеством вершин, и множество  $E$  (множество ребер) двухэлементных подмножеств множества  $V$  [1, с. 248];
- Полный граф – это граф, у которого любые две вершины соединены ребром [1, с. 250];
- Цикл (теория графов) – называется путь ненулевой длины, соединяющий вершину саму с собой и не содержащий повторяющихся рёбер [1, с. 250];
- Простой цикл – цикл, соединяющий вершину  $v$  саму с собой и не содержащий повторяющихся вершин, кроме  $v$  [1, с. 250];
- Гамильтонов цикл – это простой цикл, который проходит через каждую вершину графа  $G$  [1, с. 601];
- Модель вычисления – это набор базовых операций и издержки на их применения в алгоритме [2, с. 60-72];
- Трудоёмкость алгоритма – это количество базовых операций в принятой модели вычислений, задаваемых алгоритмом на конкретном входе [2, с. 60-72];
- Функция трудоёмкости – это отношение, связывающее входные данные алгоритма с количеством элементарных операций [2, с. 60-72];
- Временная сложность (или просто сложность) алгоритма – асимптотическая оценка функции трудоёмкости [2, с. 60-72];
- Класс P – класс полиномиально-решаемых задач. Задача относится к классу P, если существует константа  $k$  и алгоритм, решающий эту задачу за время  $O(n^k)$ , где  $n$  - для входа [2, с. 36-37];
- Класс NP – класс задач с полиномиально проверяемым решением или

класс полиномиально-проверяемых задач. Задача относится к классу NP, если её решение, полученное некоторым алгоритмом, может быть проверено с полиномиальной временной сложностью [2, с. 37];

- Задача является NP-полной (NPC или NP-Complete) если она принадлежит классу NP и к ней должны полиномиально сводиться все задачи класса NP [2, с. 38-40];
- Задача является NP-трудной (NP-hard) если оптимизационная постановка этой задачи принадлежит к классу NP-полных задач.

# ВВЕДЕНИЕ

В современном мире промедление в секунду, или даже долю секунды, может стоить миллионы рублей. Заинтересованному лицу важно получить точный ответ на вопрос быстро. Но что значит "быстро"?

Мы, люди живущие в эпоху электронных технологий, стремимся получить большую часть ответов от компьютера. Но, к сожалению, даже при нынешних вычислительных способностях, он не может решить большой объём задач за приемлемое для нас время.

Задача коммивояжёра является одной из таких задач. Её исключительной особенностью является то, что для весьма небольшого, по современным меркам, объёма данных ответ можно получить не раньше чем через месяц, не говоря уже о том, что на сегодняшний день не существует способа узнать, хотя бы с точностью до 25%, через сколько будет получено решение.

Исследование проведено для того, чтобы расширить представление о структуре дерева решений в реализации метода ветвей и границ для решения задачи коммивояжера. Результаты данного исследования и труд А.С. Ермошина и В.А. Плиско [3] будут способствовать дальнейшему исследованию, направленному на сокращение временных и вычислительных затрат при решении задачи коммивояжёра.

# ОСНОВНАЯ ЧАСТЬ

## 1 Задача коммивояжера

### 1.1 Неформальная постановка задачи

Торговому агенту (он же *коммивояжёр*) необходимо посетить  $n - 1$  город и вернуться обратно в свой город. Ему известно, что из любого города он может добраться в любой другой город и он знает сколько ему будет стоить поездка (иными словами ему известна  $n^2 - n$  стоимостей переезда). Ему необходимо совершить такой *тур*, чтобы его издержки были минимальными, а в каждом городе (кроме своего) он побывал ровно один раз.

### 1.2 Обозначения

Города будем обозначать арабскими цифрами, например:  $1, 2, 3, \dots, n - 1, n$ .

Город из которого коммивояжёр выезжает будем обозначать  $0$ .

Проезд из города  $i$  в город  $j$  обозначим  $i \rightarrow j$ , а обозначение, что проезд закрыт —  $i \nrightarrow j$ .

Стоимость проезда из города  $i$  в город  $j$  обозначим  $C(i \rightarrow j)$ .

Часть тура будем обозначать  $i \rightarrow k \rightarrow j$ , что означает, что из города  $i$  едем в город  $k$ , а из города  $k$  едем в город  $j$ . Стоит заметить, что запись  $i \nrightarrow j$  — тоже часть тура. Несколько несвязных частей тура записываются через точку с запятой, например:  $i \rightarrow k; p \rightarrow j$

Запись  $C(i \rightarrow k \rightarrow j)$  обозначает стоимость части тура. Очевидно, что:

$$C(i \rightarrow k \rightarrow j) = C(i \rightarrow k) + C(k \rightarrow j) \quad (1)$$

Тур будем обозначать заглавными латинскими буквами, например,  $T = 0 \rightarrow 3 \rightarrow 4 \rightarrow 1 \rightarrow 2 \rightarrow 0$ , обозначает тур из 5 городов (с учётом начального города). Стоимость тура обозначим  $C(T) = C(0 \rightarrow 3 \rightarrow 4 \rightarrow$



$1 \rightarrow 2 \rightarrow 0$ ). Тогда:

$$C(T) = C(0 \rightarrow 3 \rightarrow 4 \rightarrow 1 \rightarrow 2 \rightarrow 0) = C(0 \rightarrow 3) + C(3 \rightarrow 4) + \\ + C(4 \rightarrow 1) + C(1 \rightarrow 2) + C(2 \rightarrow 0) \quad (2)$$

### 1.3 Постановка в терминах теории графов

Если принять, что города - это вершины графа, а проезд между городами обозначим ребрами с весом (вес ребра равен стоимости проезда), то полученный граф будет полным ориентированным асимметричным графом состоящим из  $n$  вершин. Соответственно, задача нахождения тура сводится к задаче нахождения гамильтонова цикла минимальной стоимости.

### 1.4 Представление графа

Полученный граф представим в виде матрицы стоимости  $C$ , где  $C_{ij} = C(i \rightarrow j)$ . Так как поездка из любого города в тот же самый город невозможна, то  $C_{ii} = +\infty \forall i = 0..n - 1$ . Пример матрицы стоимости приведён в таблице 1.

Таблица 1 — Пример матрицы стоимости

	0	1	2	3	4
0	$\infty$	10	8	5	7
1	6	$\infty$	9	5	6
2	8	2	$\infty$	7	4
3	6	4	11	$\infty$	8
4	2	6	12	10	$\infty$

### 1.5 Виды задач

Задача коммивояжёра бывает двух видов: симметричной и несимметричной.

Симметричной называют такую задачу, для которой верно, что  $C_{ij} = C_{ji} \forall i, j = 0..n-1$ . А несимметричной, называют такую задачу, для которой верно, что  $\exists i, j | i, j \subseteq 0..n-1 \& C_{ij} \neq C_{ji}$

## 1.6 Принадлежность к классам сложности

На сегодняшний день не существует алгоритма, который бы решал задачу коммивояжёра за полиномиальное время. А так же в силу особенности задачи, сложность проверки решения не меньше, чем сложность решения, следовательно, на сегодняшний день не существует алгоритма, которой проверял бы ответ на задачу коммивояжёра за полиномиальное время. Из выше сказанного следует, что задача коммивояжёра не принадлежит к классу NP и тем более NP-полных задач. Но оптимизационная постановка задачи коммивояжёра (найти тур который меньше k) принадлежит к классу NP-полных задач [4], следовательно, задача коммивояжёра является NP-трудной задачей.

## 2 Алгоритмы решения задачи коммивояжёра

### 2.1 Полный перебор

Суть алгоритма полного перебора – это сгенерировать все возможные варианты гамильтонова цикла, рассчитать стоимость каждого и выбрать среди всех самый дешёвый. Не смотря на то, что алгоритм звучит очень просто, его сложность равна  $O(n!)$ , что делает его не применимым на практике.

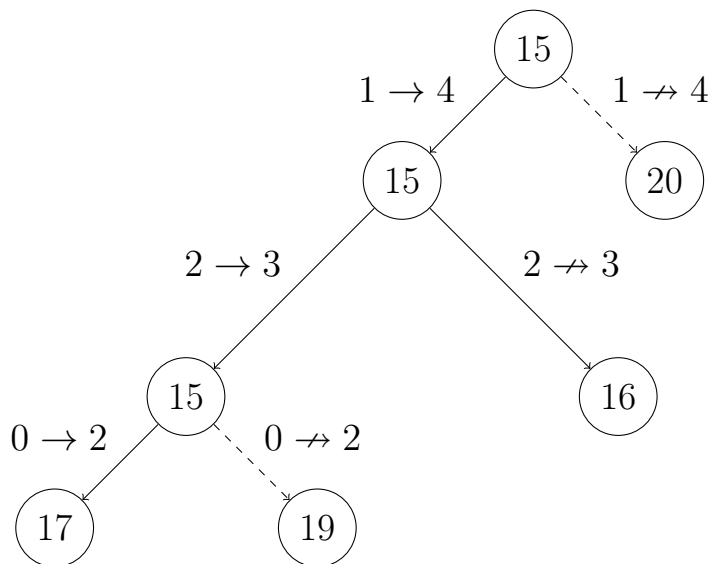
### 2.2 Метод ветвей и границ

Применение метода ветвей и границ [5] для решения задачи коммивояжёра впервые было описано в [6] в 1963 году. С тех пор этот алгоритм стал классическим алгоритмом для решения задачи коммивояжёра. Подробное описание алгоритма можно найти в [2], [6], [7] и [8].

Метод ветвей и границ порождает дерево решений. Одно из таких

деревьев показано на рисунке 1. Если вершина удалена, то ребро дерева обозначается пунктиром. Минимальная стоимость пути обозначается цифрой в вершине. Ветвление обозначается на ребрах дерева.

Рисунок 1 — Пример дерева решений



### 2.3 Метод ветвей и границ с предвычисленным туром

Данный алгоритм отличается от алгоритма основанного на методе ветвей и границ только тем, что вначале работает приближенный алгоритм, и, как результат, мы имеем уже какой-то тур, который позволяет уменьшить дерево решений.

### 3 Методология экспериментального исследования

Для оценки поискового дерева решений рассчитывались следующие параметры:

- количество порождённых вершин дерева решений (обозначим  $R$ );
- максимальное количество вершин дерева решений хранимых в памяти компьютера (обозначим  $R^s$ );
- максимальное количество листьев дерева решений (обозначим  $L$ );
- максимальная глубина дерева решений (обозначим  $D$ ).

Данные показатели рассчитывались для метода ветвей и границ с предвычисленным туром и без него. Длина входа (количество городов) варьи-

руется от 10 до 40 для симметричной и от 10 до 45 для несимметричной задачи. А так же рассчитывалось время работы реализации для каждого конкретного входа. Технические и программные особенности компьютера на котором проводились эксперименты смотреть в Приложении А.1.

## 4 Особенности реализации

### 4.1 Структура хранения листьев поискового дерева решений

Рассмотрим три широко известные структуры данных: бинарная куча, красно – чёрное дерево и АВЛ-дерево. На основе [4] и [9] составлена таблица 2 времени работы операций над структурами данных используемых в алгоритме. Стоит также отметить, что у трёх структур данных одинаковое

Таблица 2 — Сравнение сложности операций над структурами данных

Структура данных	Вставка	Удаление	Поиск мин. элемента
Бинарная куча	$O(\log(n))$	$O(\log(n))$	$O(1)$
Красно-чёрное дерево	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$
АВЛ-дерево	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$

требование к памяти –  $O(n)$ . Из выше изложенного можно допустить, что бинарная куча лучше всего подходит для организации очереди с приоритетом. Разумеется, что этот вопрос требует дополнительной проработки, однако он выходит за рамки данной работы.

Таким образом было принято решение использовать для реализации очереди бинарную кучу.

### 4.2 Язык программирования и версия компилятора

Алгоритмы реализованы на языке высокого уровня C++. Версия компилятора 4.9.2 20150212 (Red Hat 4.9.2-6) (GCC).

## 5 Результаты экспериментального исследования

В таблице 3 показаны результаты экспериментов для несимметричной задачи, усреднённые по 10000 различных входов, для метода ветвей и границ без предвычисленного тура, а в таблице 4 – с предвычисленным туром. В таблице 5 представлено сравнение по времени двух реализаций.

В таблице 6 показаны результаты экспериментов для симметричной задачи, усреднённые по 10000 различных входов, для метода ветвей и границ без предвычисленного тура, а в таблице 7 – с предвычисленным туром. В таблице 8 представлено сравнение по времени двух реализаций.

Таблица 3 — Результаты экспериментов для несимметричной задачи без предвычисленного тура

$n$	$R$	$R^s$	$L$	$D$
10	27,3818	26,0006	12,2086	8,4763
11	34,1209	32,41	15,3252	9,5425
12	41,8944	39,6887	18,8614	10,61
13	51,7003	49,1343	23,5132	11,6684
14	63,7114	60,4051	28,9028	12,7452
15	75,6685	71,6478	34,4567	13,7785
16	94,1878	89,2901	43,0441	14,8479
17	114,4132	108,7286	52,6099	15,8991
18	138	131,1474	63,6907	16,9466
19	167,666	159,2927	77,488	17,995
20	204,6709	194,4844	94,6051	19,0623
21	247,1701	234,7785	114,4252	20,1158
22	300,0079	285,2461	139,2768	21,17
23	362,2493	345,2066	168,7904	22,2096
24	432,3995	411,3024	201,1305	23,2573
25	515,7569	490,2203	240,2409	24,3034
26	617,9083	589,5278	289,1755	25,353
27	747,5442	711,9779	349,4744	26,4102
28	905,3015	863,4112	424,3289	27,4437
29	1042,9325	996,1789	489,89	28,4771

Продолжение таблицы 3

$n$	$R$	$R^s$	$L$	$D$
30	1307,0225	1246,8379	613,2943	29,5551
31	1583,4909	1510,852	743,2093	30,6041
32	1812,4283	1730,6447	851,1413	31,6204
33	2316,3902	2207,7186	1087,8725	32,7164
34	2680,3074	2556,0427	1260,4198	33,7282
35	3270,9746	3136,0451	1547,8948	34,7923
36	3957,6486	3770,8231	1859,821	35,8193
37	4688,3636	4477,4941	2209,1355	36,8631
38	5578,101	5339,393	2632,3848	37,9124
39	6774,9806	6480,2998	3200,6113	38,9523
40	8026,1177	7663,2104	3790,317	40,0086
41	9452,7056	9027,8152	4462,9908	41,0349
42	11593,2346	11079,1673	5471,925	42,0977
43	13944,6094	13336,7138	6598,0777	43,1274
44	16978,4687	16240,4008	8025,3017	44,1745
45	20112,6697	19205,3559	9494,3118	45,2081

Таблица 4 — Результаты экспериментов для несимметричной задачи с пред-  
вычисленным туром

$n$	$R$	$R^s$	$L$	$D$
10	25,4895	24,2749	10,6052	8,3077
11	32,4131	30,8562	13,8545	9,4214
12	40,3752	38,311	17,53	10,5288
13	50,2043	47,7721	22,1756	11,6083
14	62,4286	59,2593	27,7572	12,6989
15	74,4482	70,5666	33,3575	13,7458
16	93,0955	88,3138	42,0456	14,8304
17	113,4523	107,8772	51,7384	15,8843
18	137,107	130,3565	62,8765	16,9349
19	166,85	158,5581	76,7334	17,9865
20	203,9896	193,8736	93,9852	19,0577
21	246,4994	234,1818	113,8079	20,1133
22	299,3952	284,7077	138,7144	21,1681
23	361,717	344,7423	168,2992	22,2082
24	431,928	410,9164	200,6966	23,2552
25	515,3161	489,822	239,8362	24,3009
26	617,5029	589,1489	288,7908	25,353
27	747,2335	711,7077	349,1984	26,4099
28	905,0563	863,1886	424,105	27,4434
29	1042,7127	995,9809	489,691	28,4771



Продолжение таблицы 4

$n$	$R$	$R^s$	$L$	$D$
30	1306,7882	1246,6251	613,0775	29,5551
31	1583,2619	1510,6456	742,9965	30,6041
32	1812,2085	1730,4627	850,952	31,6204
33	2316,2553	2207,5977	1087,7501	32,7164
34	2680,0874	2555,8416	1260,2163	33,7282
35	3270,7984	3135,8808	1547,7309	34,7923
36	3957,5309	3770,7168	1859,7201	35,8193
37	4688,2658	4477,411	2209,0506	36,8631
38	5578,0274	5339,3264	2632,3208	37,9124
39	6774,9331	6480,26	3200,5728	38,9523
40	8026,0066	7663,1067	3790,2125	40,0086
41	9452,6302	9027,7455	4462,9215	41,0349
42	11593,1997	11079,135	5471,896	42,0977
43	13944,5564	13336,6632	6598,0296	43,1274
44	16978,428	16240,3716	8025,2725	44,1745
45	20112,6339	19205,3236	9494,2807	45,2081

Таблица 5 — Результаты экспериментов для несимметричной задачи с предвычисленным туром

$n$	Время работы реализации без предвычисленного тура (мс)	Время работы реализации с предвычисленным туром (мс)
10	0,061	0,053
11	0,089	0,078
12	0,130	0,115
13	0,182	0,162
14	0,267	0,239
15	0,359	0,323
16	0,511	0,463
17	0,966	0,884
18	1,359	1,241
19	1,971	1,807
20	1,793	1,664
21	2,394	2,246
22	3,990	3,699
23	4,253	4,043
24	5,574	5,348
25	7,190	6,948
26	11,133	10,354
27	15,002	13,755
28	16,173	15,793
29	19,954	19,518

Продолжение таблицы 5

<i>n</i>	Время работы реализации без предвычисленного тура (мс)	Время работы реализации с предвычисленным туром (мс)
30	34,630	30,373
31	45,112	39,220
32	55,561	47,746
33	76,844	65,644
34	77,064	75,132
35	121,948	103,600
36	130,482	127,440
37	164,236	160,733
38	244,868	212,464
39	270,393	265,953
40	339,627	334,978
41	424,346	418,872
42	595,439	552,380
43	703,705	697,637
44	902,180	896,657
45	1170,816	1127,744

Таблица 6 — Результаты экспериментов для симметричной задачи без пред-  
вычисленного тура

$n$	$R$	$R^s$	$L$	$D$
10	60,6656	57,1511	27,4256	9,3623
11	83,1028	78,2509	37,5745	10,5072
12	111,6211	105,2197	50,7682	11,6485
13	150,5694	141,7687	68,458	12,7881
14	200,6982	188,8226	91,2283	13,9211
15	266,5282	250,381	121,3013	15,0465
16	359,9694	337,1213	162,6887	16,1863
17	466,4938	437,9139	212,2118	17,3126
18	612,0243	574,7419	278,3175	18,4546
19	806,0066	756,6092	366,467	19,5652
20	1067,0361	999,7661	484,328	20,7256
21	1405,9564	1315,4887	637,1501	21,833
22	1828,9312	1714,1206	831,6662	22,9444
23	2371,2432	2219,559	1077,8127	24,051
24	3168,1383	2960,7212	1435,6464	25,2177
25	4121,509	3856,0181	1875,6476	26,3376

Продолжение таблицы 6

$n$	$R$	$R^s$	$L$	$D$
26	5502,5642	5150,7069	2500,0136	27,4545
27	7000,0088	6563,9129	3189,8337	28,5712
28	9197,3386	8596,3882	4183,0461	29,7034
29	12131,0809	11325,9179	5513,2402	30,8229
30	15496,1668	14475,3185	7048,064	31,9154
31	20883,335	19558,536	9526,4308	33,0894
32	27097,9246	25312,0693	12317,1501	34,2079
33	34445,071	32190,1363	15672,1022	35,2667
34	46233,6206	43194,6862	21029,6141	36,434
35	59912,533	55727,2798	27044,2205	37,5591
36	77451,4717	72247,9337	35256,3031	38,6684
37	101615,9913	94658,9135	45893,4239	39,7985
38	132926,2973	123820,1696	60171,3039	40,9295
39	168365,3634	156791,813	76589,5852	42,03
40	227485,608	211435,0585	102778,1995	43,1731

Таблица 7 — Результаты экспериментов для симметричной задачи с пред-  
вычисленным туром

$n$	$R$	$R^s$	$L$	$D$
10	55,3891	52,4077	22,5828	9,2057
11	77,5985	73,3321	32,3807	10,3985
12	106,0158	100,2274	45,3808	11,5714
13	144,6419	136,4911	62,6976	12,7291
14	194,7882	183,5629	85,4444	13,8859
15	260,4026	244,8911	115,2617	15,0152
16	353,4437	331,4345	156,191	16,1654
17	460,2528	432,4842	206,056	17,2958
18	605,5565	568,9941	271,8777	18,4435
19	799,4167	750,757	359,8829	19,5532
20	1060,7784	994,3004	478,127	20,7235
21	1399,3263	1309,6358	630,5563	21,8278
22	1822,0212	1708,0039	824,7019	22,9409
23	2365,0549	2214,0196	1071,6816	24,0465
24	3161,6144	2955,2231	1429,156	25,2147
25	4114,7633	3850,0197	1869,0025	26,3366

Продолжение таблицы 7

$n$	$R$	$R^s$	$L$	$D$
26	5496,2083	5145,3212	2493,6972	27,4539
27	6990,548	6555,5205	3180,3281	28,5707
28	9190,0179	8589,8916	4175,7945	29,7018
29	12125,3044	11320,771	5507,6327	30,8248
30	15489,4968	14469,4072	7041,4889	31,9145
31	20879,7083	19555,4535	9523,202	33,0891
32	27092,1789	25306,9079	12311,6309	34,2076
33	34438,8371	32184,3584	15665,924	35,2667
34	46227,3666	43189,3981	21023,4252	36,4322
35	59905,8539	55721,0735	27037,8322	37,5584
36	77440,7015	72239,7077	35245,5427	38,6676
37	101611,191	94654,6959	45888,9009	39,7984
38	132922,8609	123817,495	60168,4726	40,9292
39	168360,1334	156787,4123	76584,5603	42,029
40	227479,5296	211429,4127	102772,0582	43,1726

Таблица 8 — Результаты экспериментов для симметричной задачи с предвычисленным туром

$n$	Время работы реализации без предвычисленного тура (мс)	Время работы реализации с предвычисленным туром (мс)
10	60,6656	57,1511
11	83,1028	78,2509
12	111,6211	105,2197
13	150,5694	141,7687
14	200,6982	188,8226
15	266,5282	250,381
16	359,9694	337,1213
17	466,4938	437,9139
18	612,0243	574,7419
19	806,0066	756,6092
20	1067,0361	999,7661
21	1405,9564	1315,4887
22	1828,9312	1714,1206
23	2371,2432	2219,559
24	3168,1383	2960,7212
25	4121,509	3856,0181



Продолжение таблицы 8

<i>n</i>	Время работы реализации без предвычисленного тура (мс)	Время работы реализации с предвычисленным туром (мс)
26	5502,5642	5150,7069
27	7000,0088	6563,9129
28	9197,3386	8596,3882
29	12131,0809	11325,9179
30	15496,1668	14475,3185
31	20883,335	19558,536
32	27097,9246	25312,0693
33	34445,071	32190,1363
34	46233,6206	43194,6862
35	59912,533	55727,2798
36	77451,4717	72247,9337
37	101615,9913	94658,9135
38	132926,2973	123820,1696
39	168365,3634	156791,813
40	227485,608	211435,0585

## 6 Графическая иллюстрация результатов экспериментального исследования

### 6.1 Анализ количества порождённых вершин

На рисунках 2, 3, 4, 5, 6, 7 показана зависимость количества порождённых вершин от количества городов для разных типов задач коммивояжёра с использованием различных алгоритмов решения.

Рисунок 2 — Зависимость количества порождённых вершин от количества городов для несимметричной задачи без предвычисленного тура

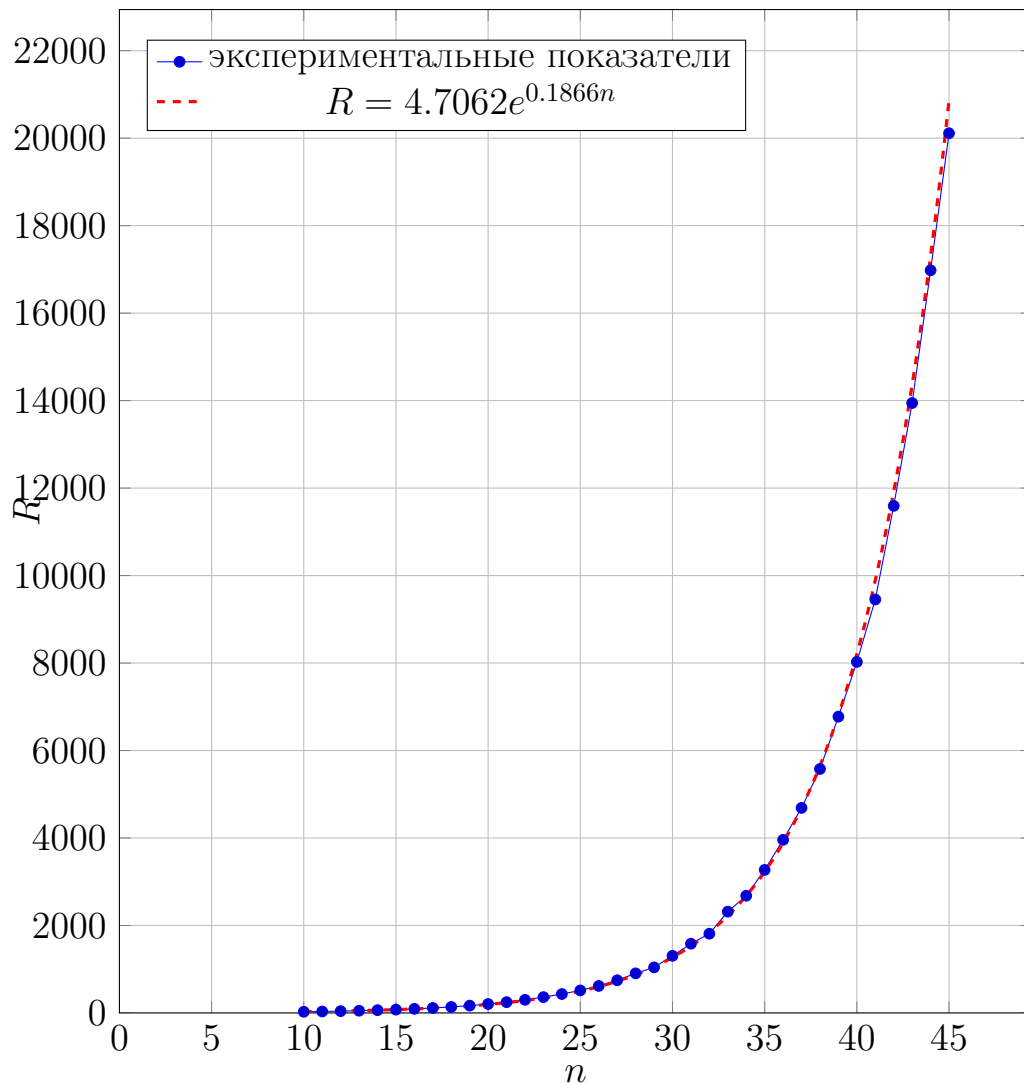


Рисунок 3 — Зависимость количества порождённых вершин от количества городов для несимметричной задачи с предвычисленным туром

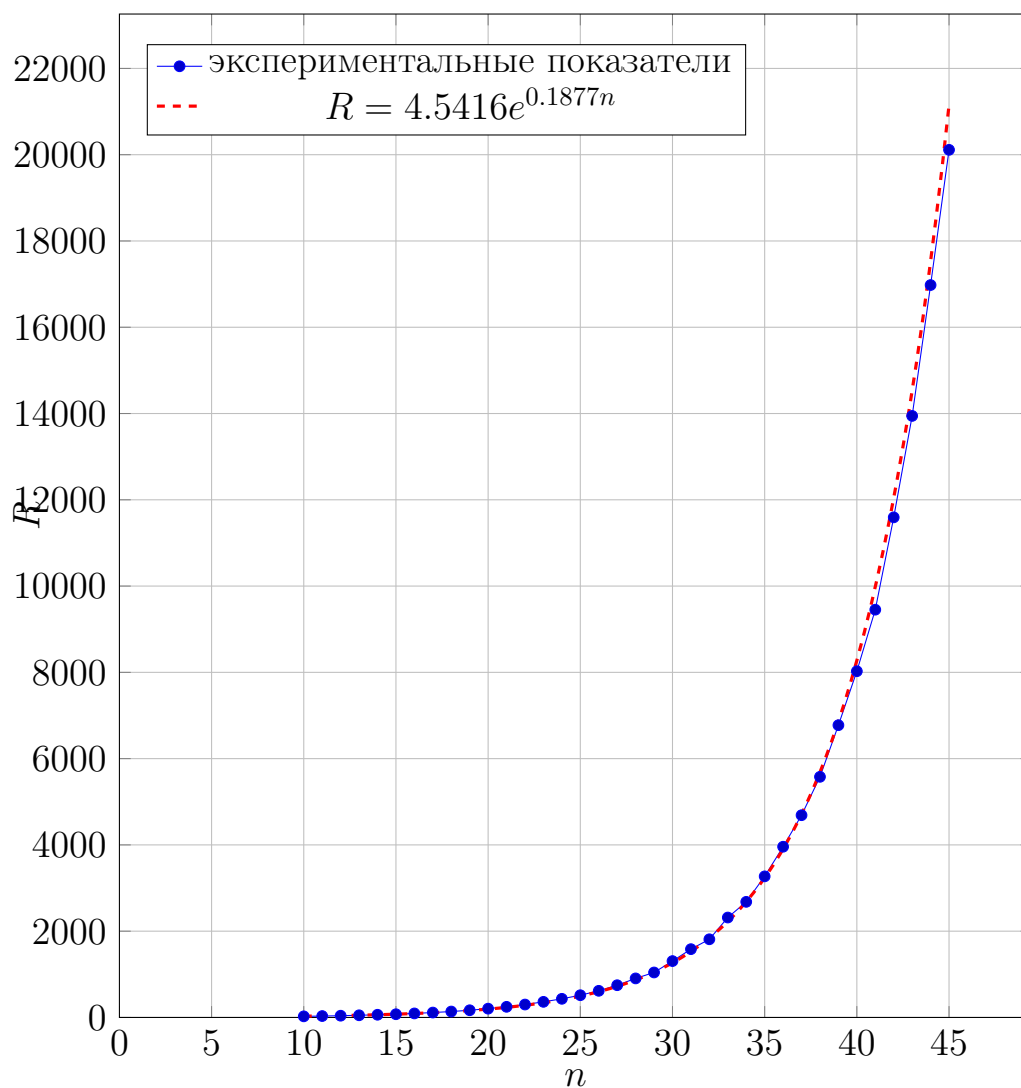


Рисунок 4 — Разность между количеством порождённых вершин для несимметричной задачи без предвычисленного тура и с предвычисленным туром

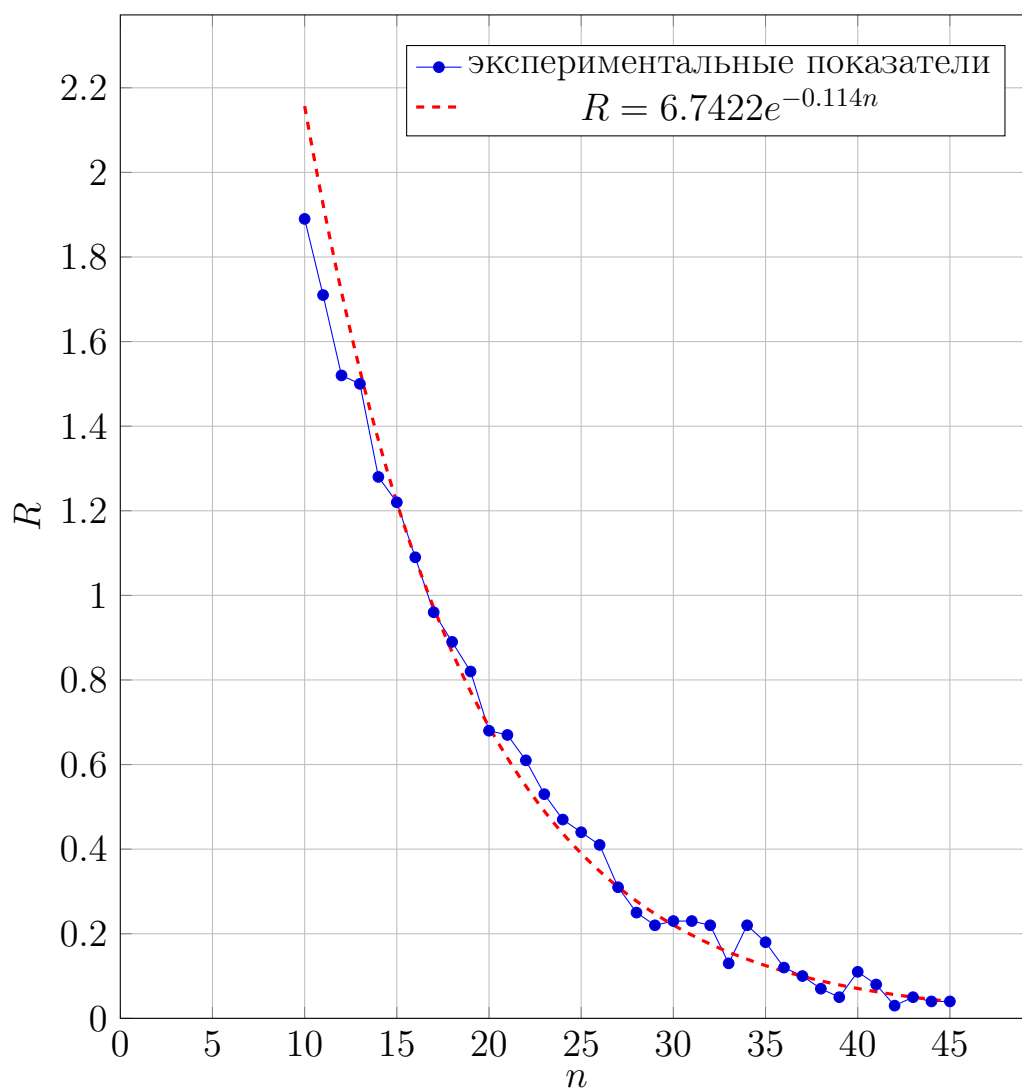


Рисунок 5 — Зависимость количества порождённых вершин от количества городов для симметричной задачи без предвычисленного тура

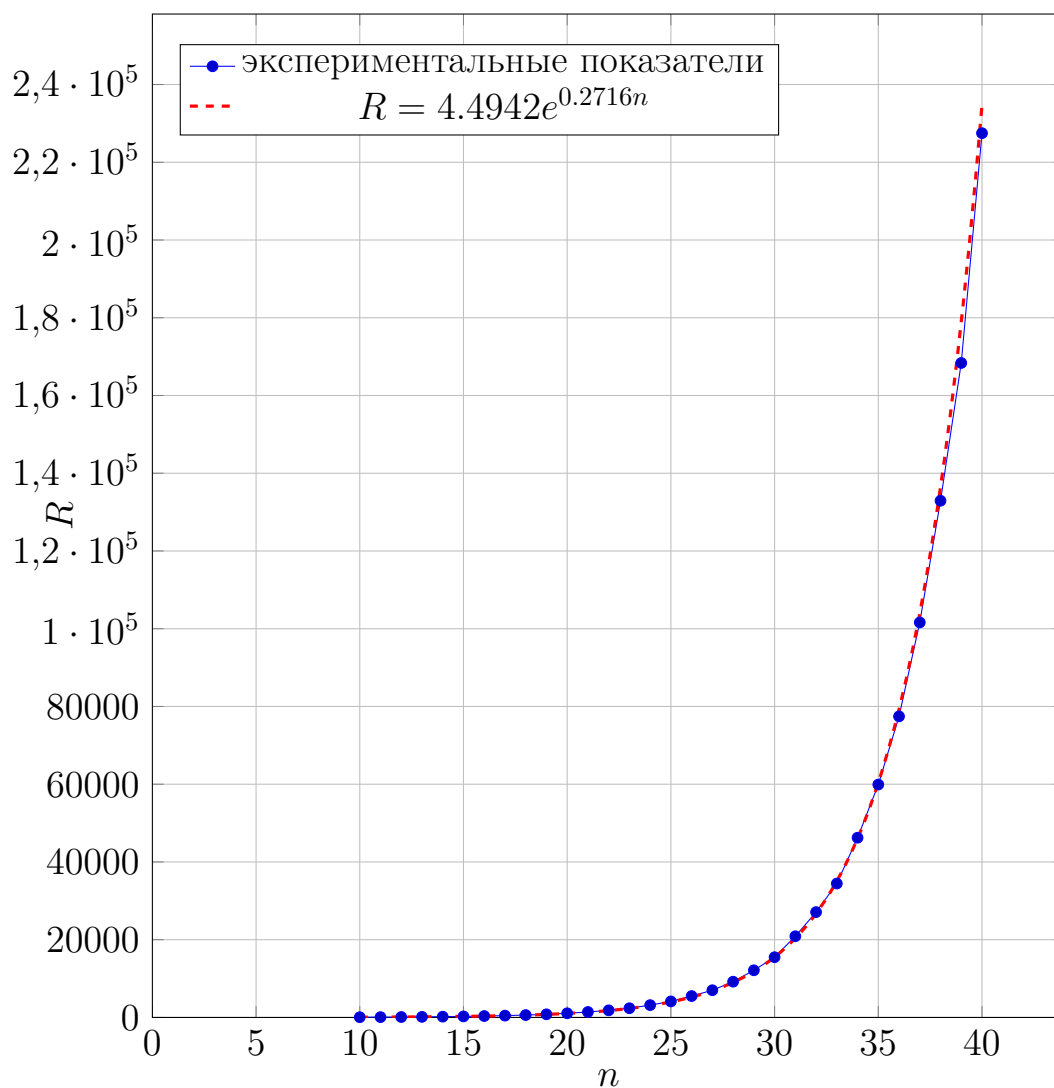


Рисунок 6 — Зависимость количества порождённых вершин от количества городов для симметричной задачи с предвычисленным туром

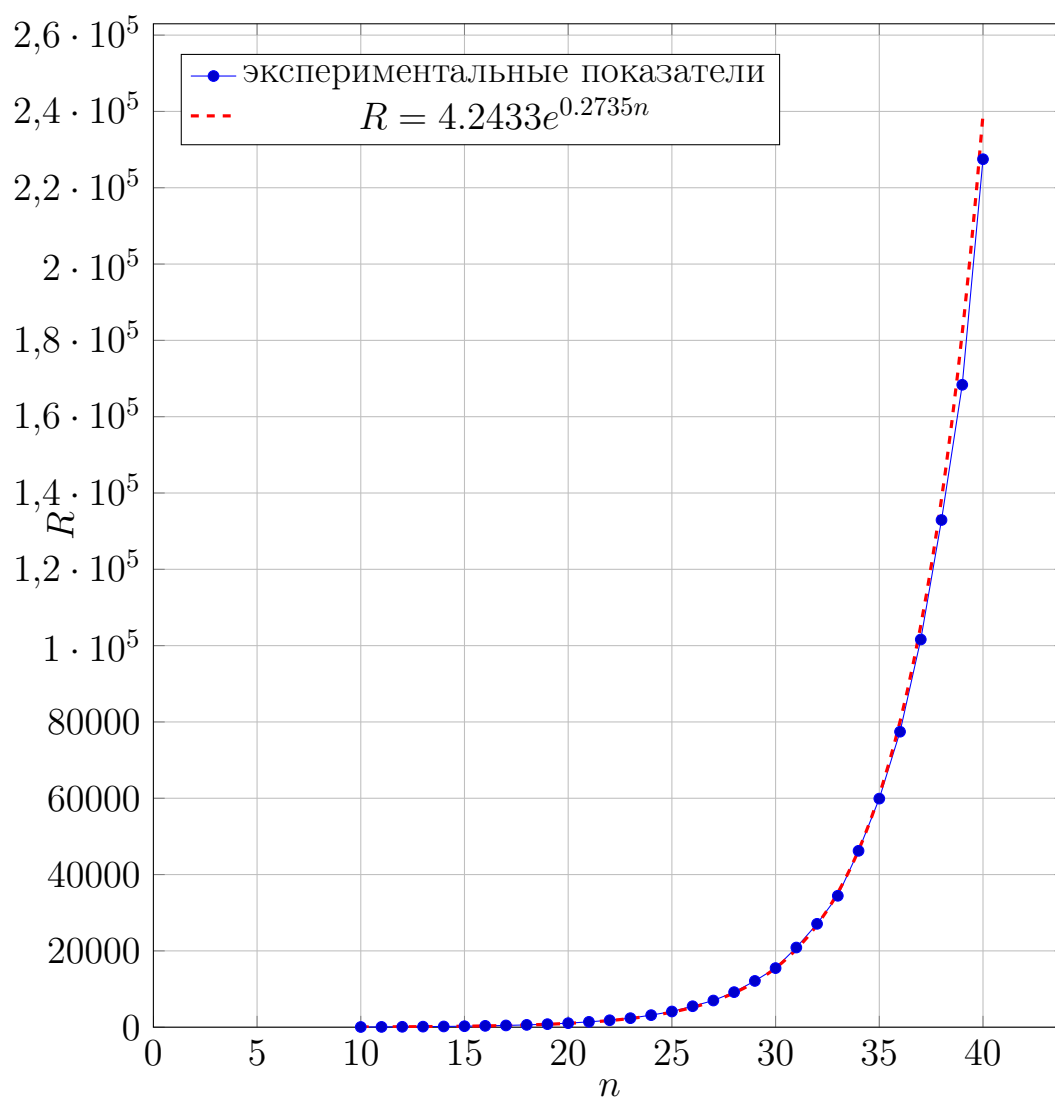
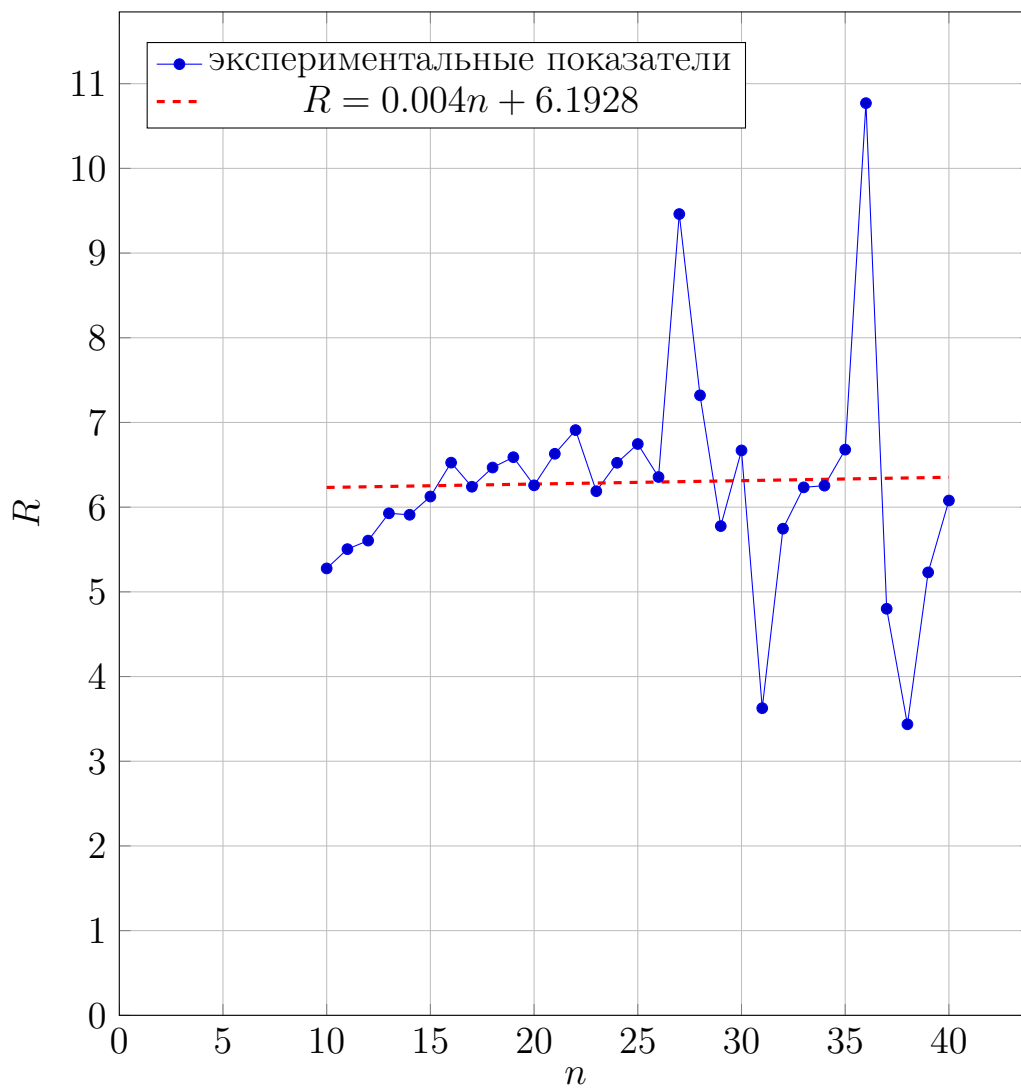


Рисунок 7 — Разность между количеством порождённых вершин для симметричной задачи без предвычисленного тура и с предвычисленным туром



## 6.2 Анализ количества хранимых порождённых вершин

На рисунках 8, 9, 10, 11, 12, 13 показана зависимость количества хранимых порождённых вершин от количества городов для разных типов задач коммивояжёра с использованием различных алгоритмов решения.

Рисунок 8 — Зависимость количества хранимых порождённых вершин от количества городов для несимметричной задачи без предвычисленного тура

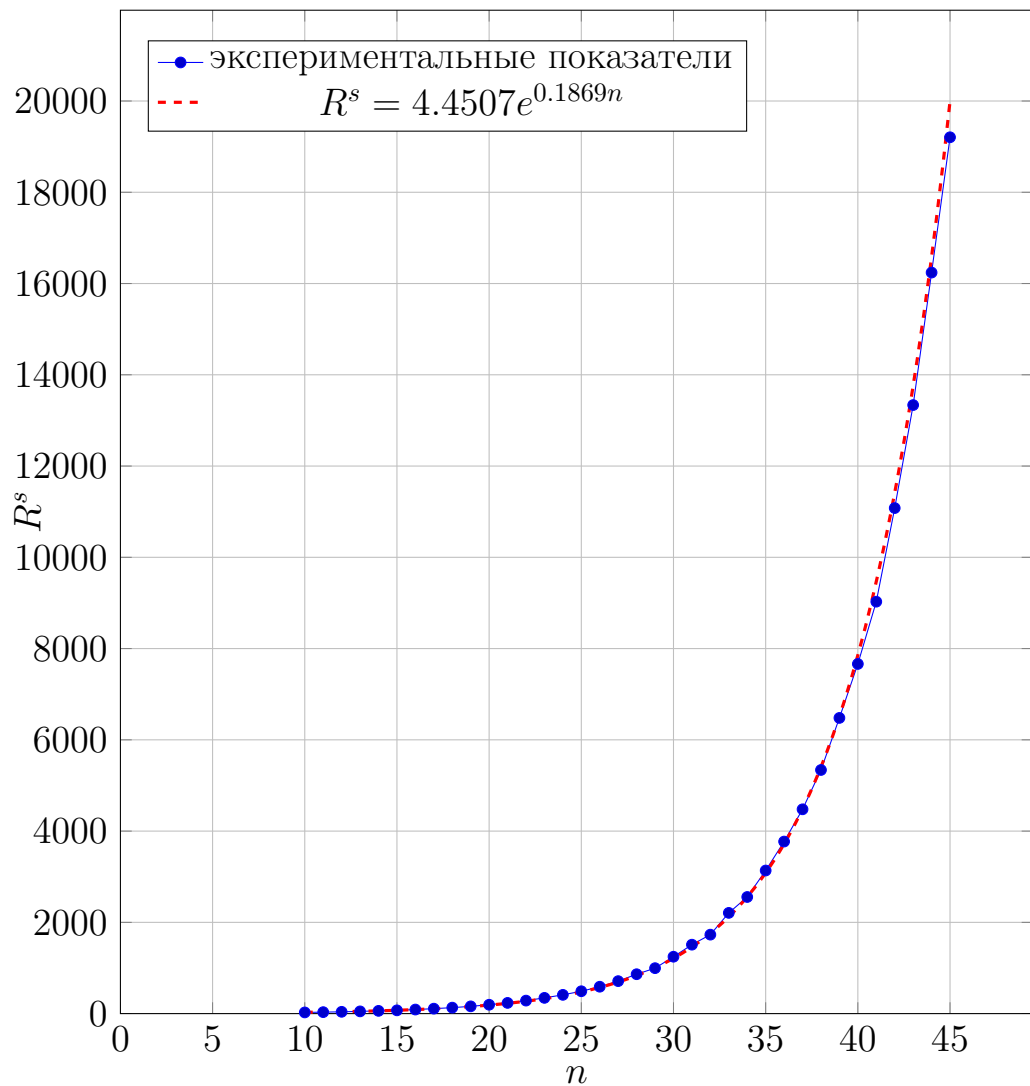




Рисунок 9 — Зависимость количества хранимых порождённых вершин от количества городов для несимметричной задачи с предвычисленным туром

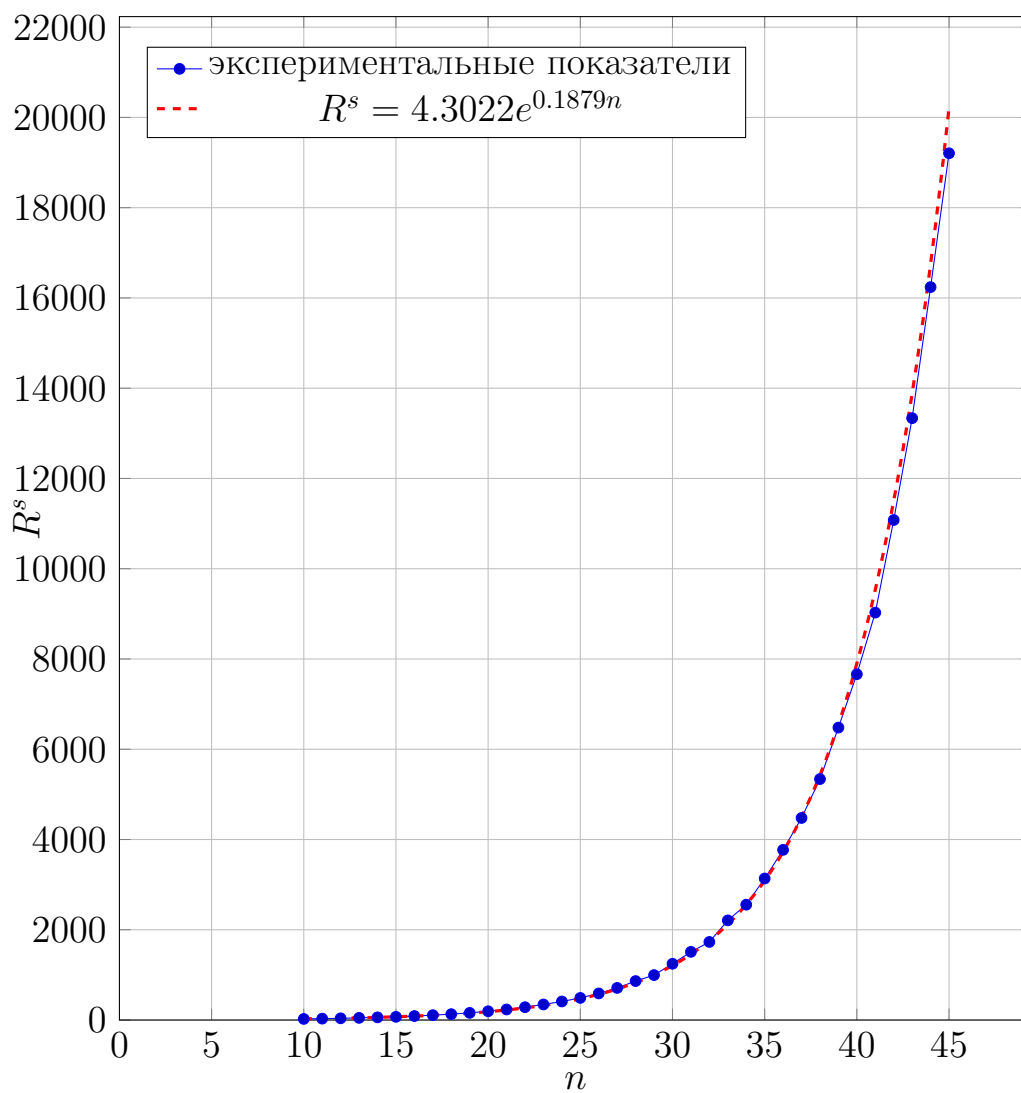


Рисунок 10 — Разность между количеством хранимых порождённых вершин для несимметричной задачи без предвычисленного тура и с предвычисленным туром

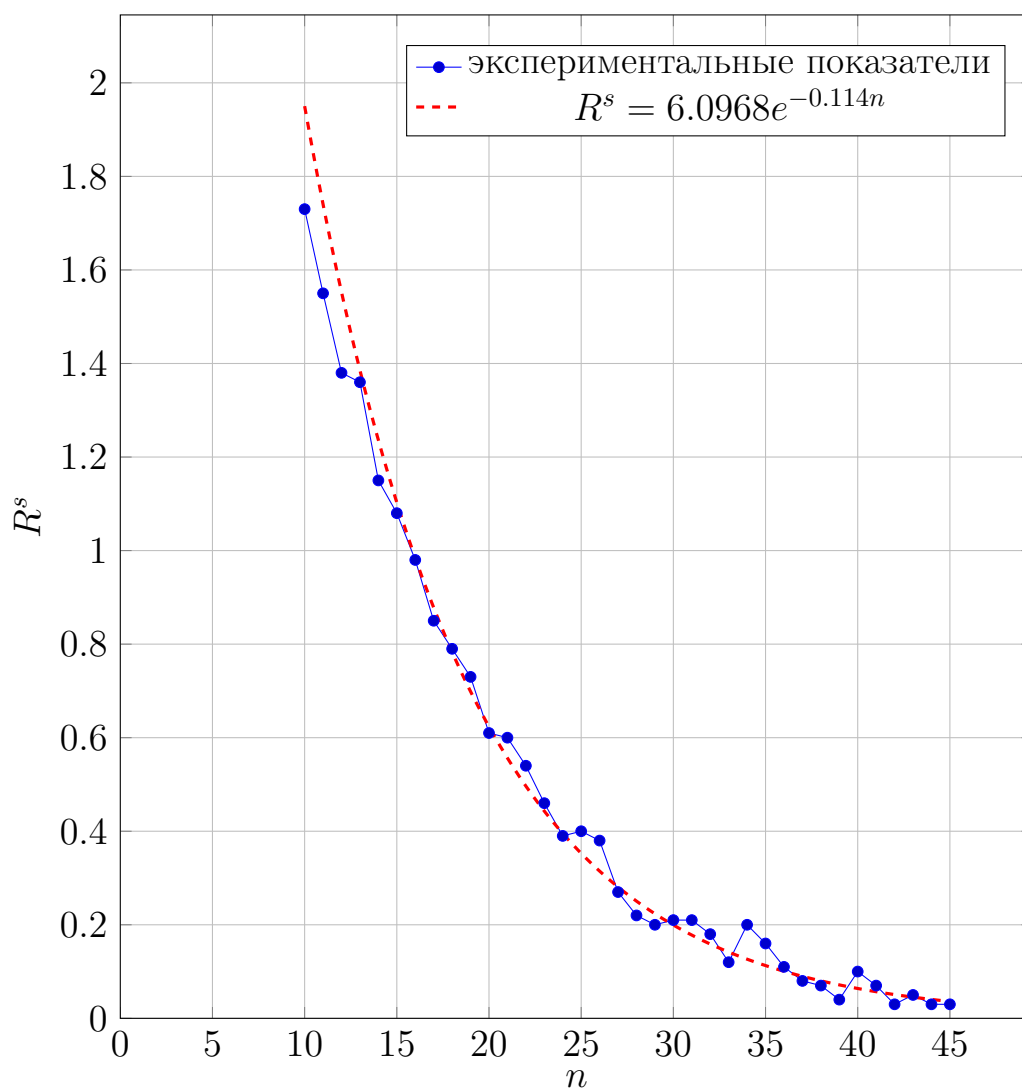


Рисунок 11 — Зависимость количества порождённых вершин от количества городов для симметричной задачи без предвычисленного тура

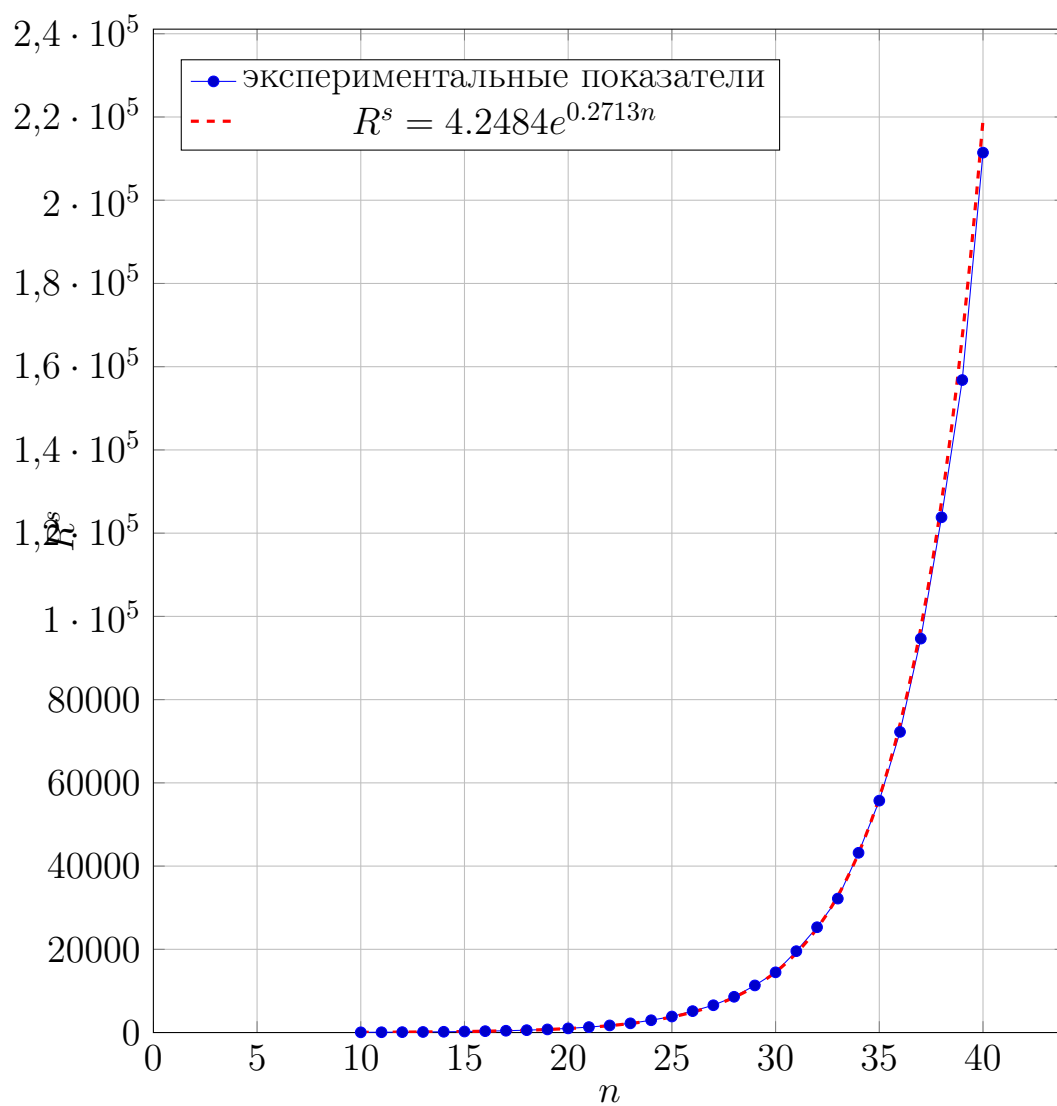


Рисунок 12 — Зависимость количества порождённых вершин от количества городов для симметричной задачи с предвычисленным туром

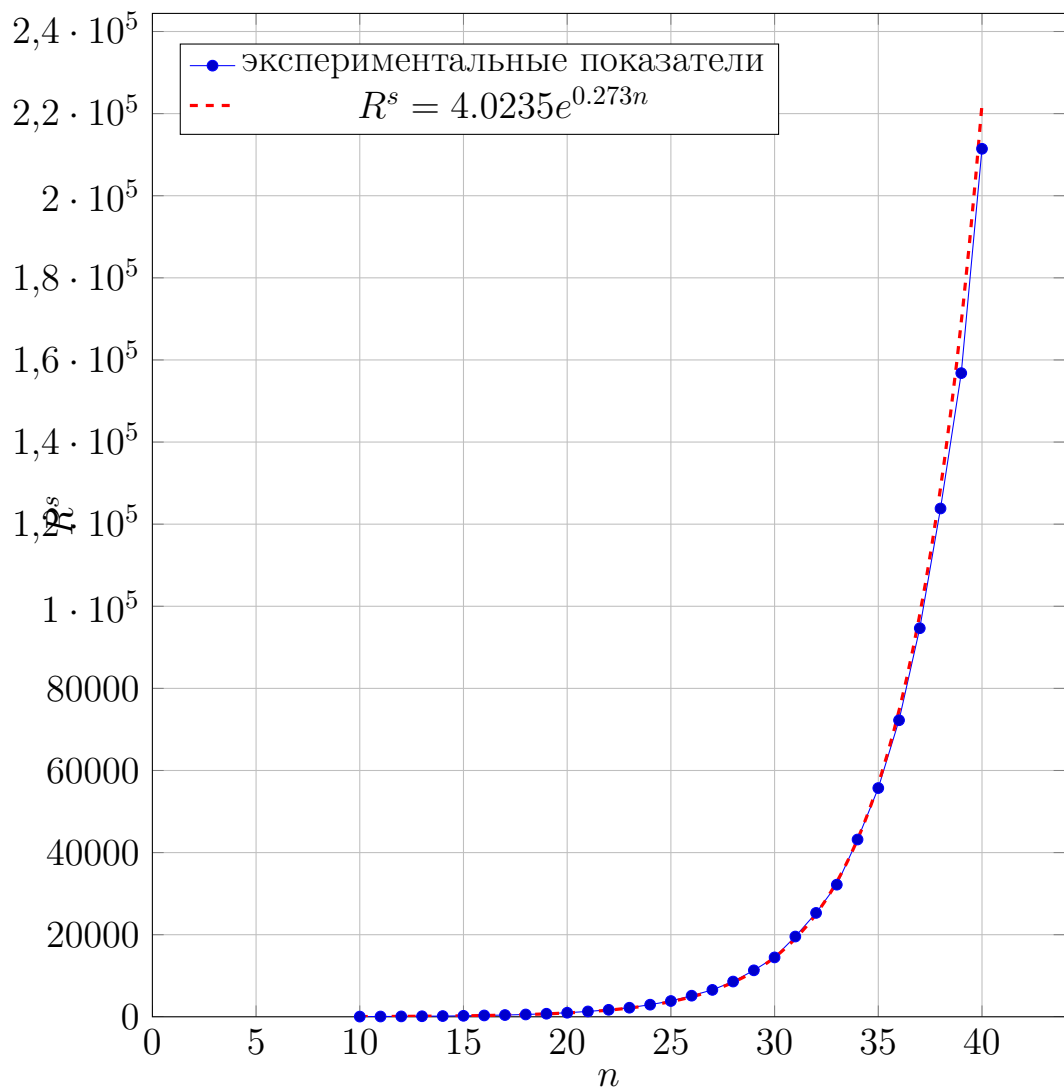
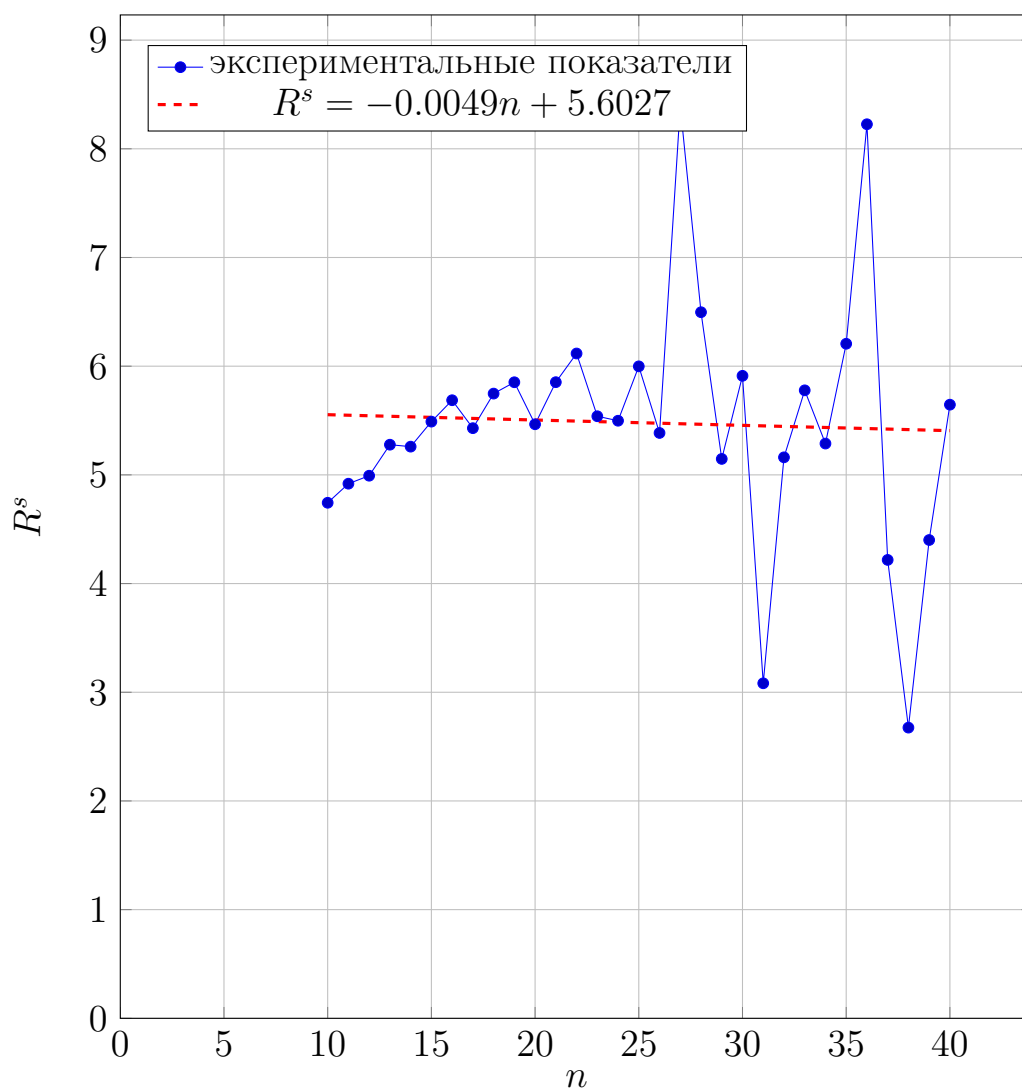


Рисунок 13 — Разность между количеством хранимых порождённых вершин для симметричной задачи без предвычисленного тура и с предвычисленным туром



### 6.3 Анализ максимального количества листьев дерева решений

На рисунках 14, 15, 16, 17, 18, 19 показана зависимость максимального количества листьев дерева решений от количества городов для разных типов задач коммивояжёра с использованием различных алгоритмов решения.

Рисунок 14 — Зависимость максимального количества листьев от количества городов для несимметричной задачи без предвычисленного тура

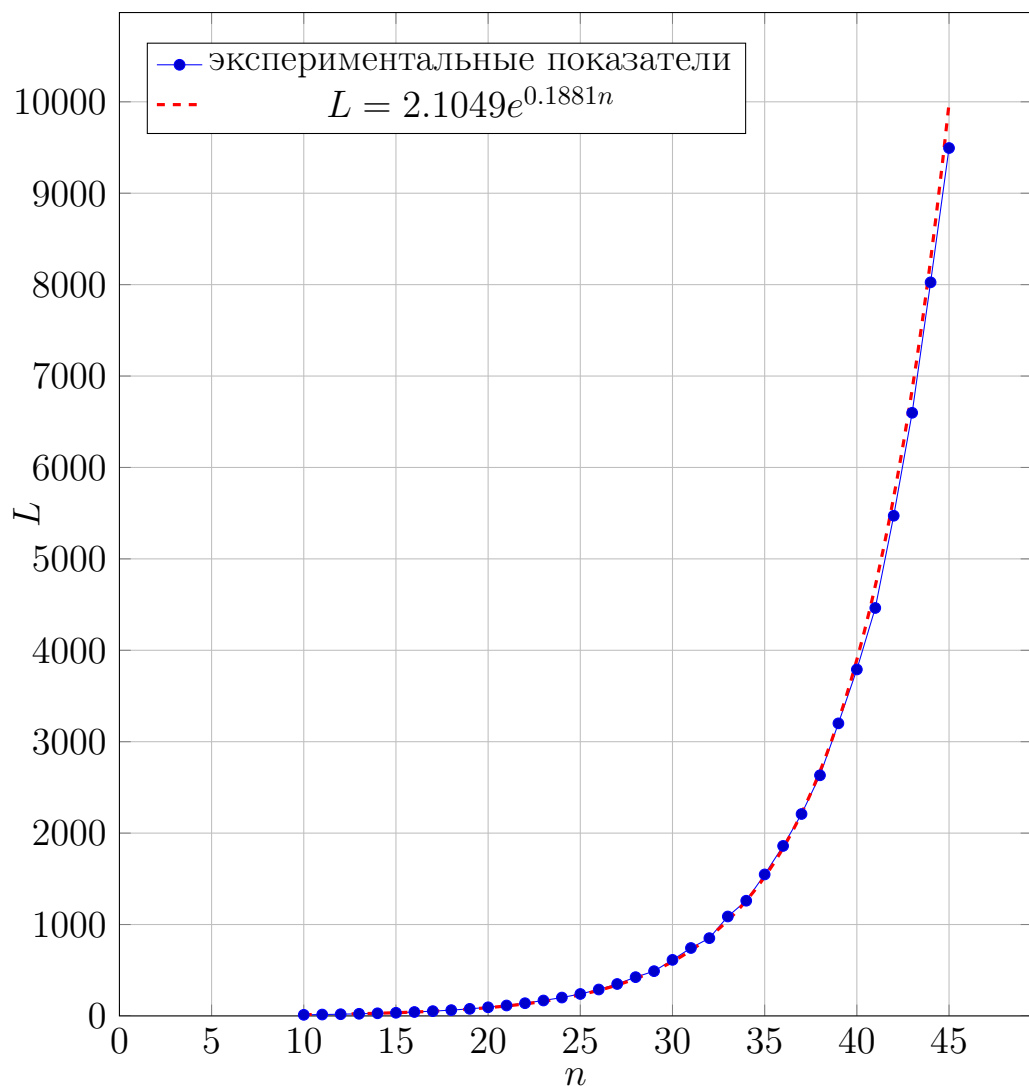


Рисунок 15 — Зависимость максимального количества листьев от количества городов для несимметричной задачи с предвычисленным туром

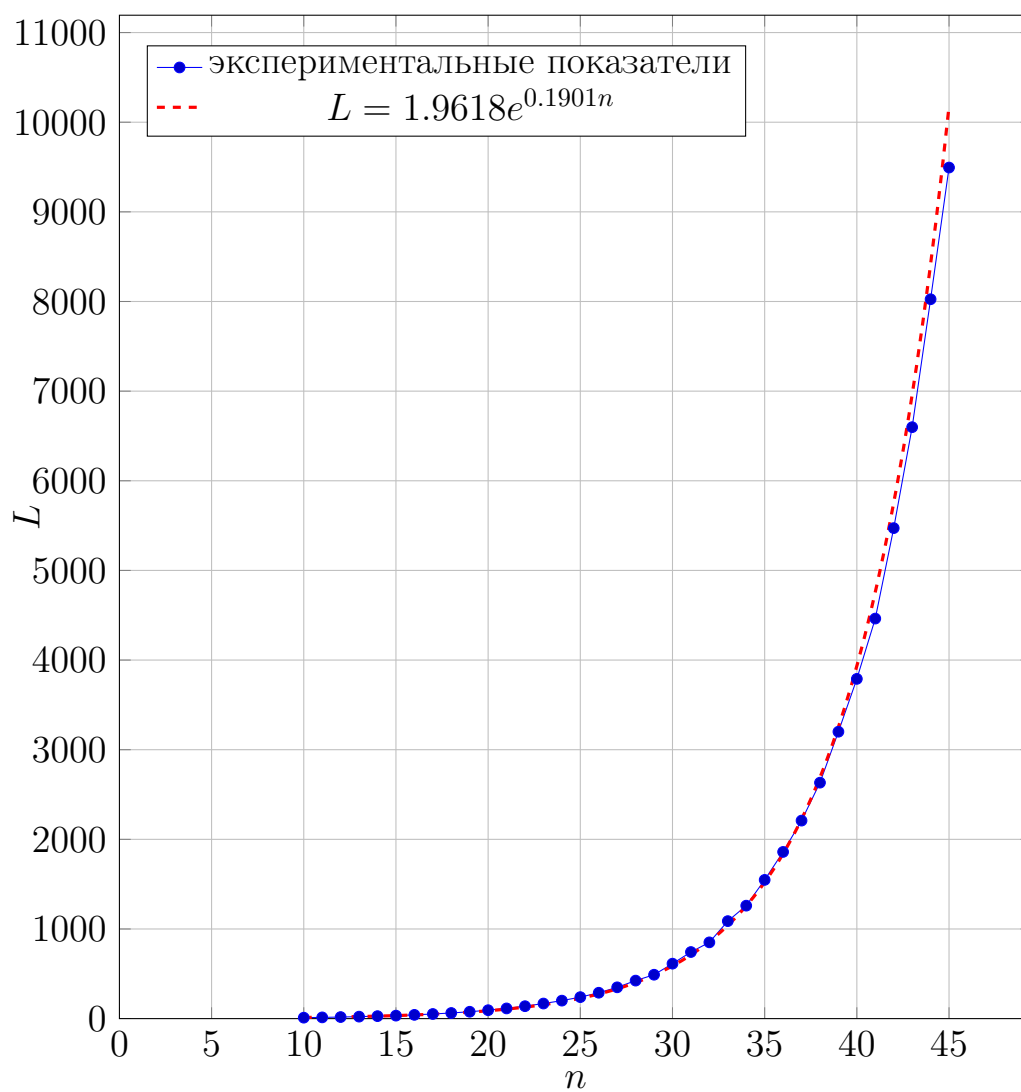


Рисунок 16 — Разность между максимальным количеством листьев для несимметричной задачи без предвычисленного тура и с предвычисленным туром

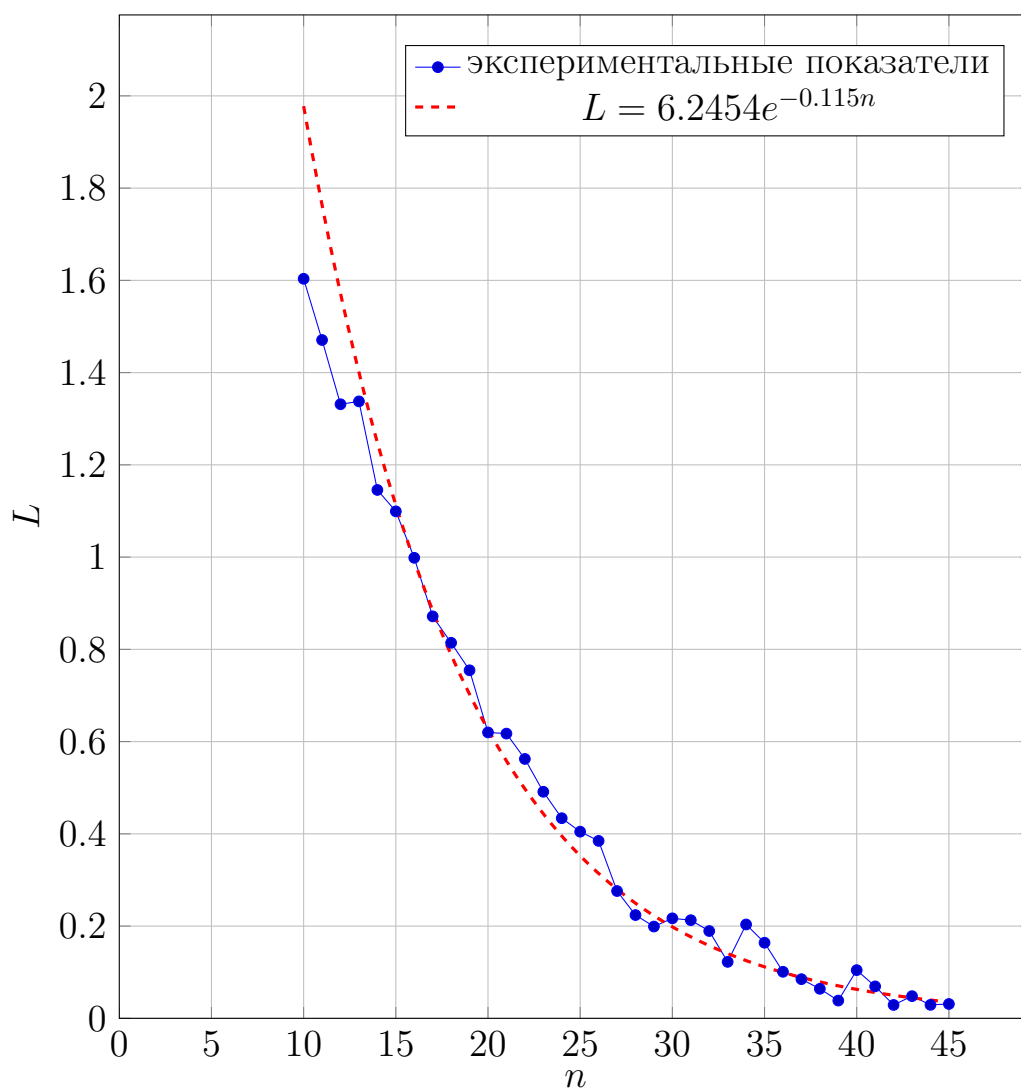




Рисунок 17 — Зависимость максимального количества листьев от количества городов для симметричной задачи без предвычисленного тура

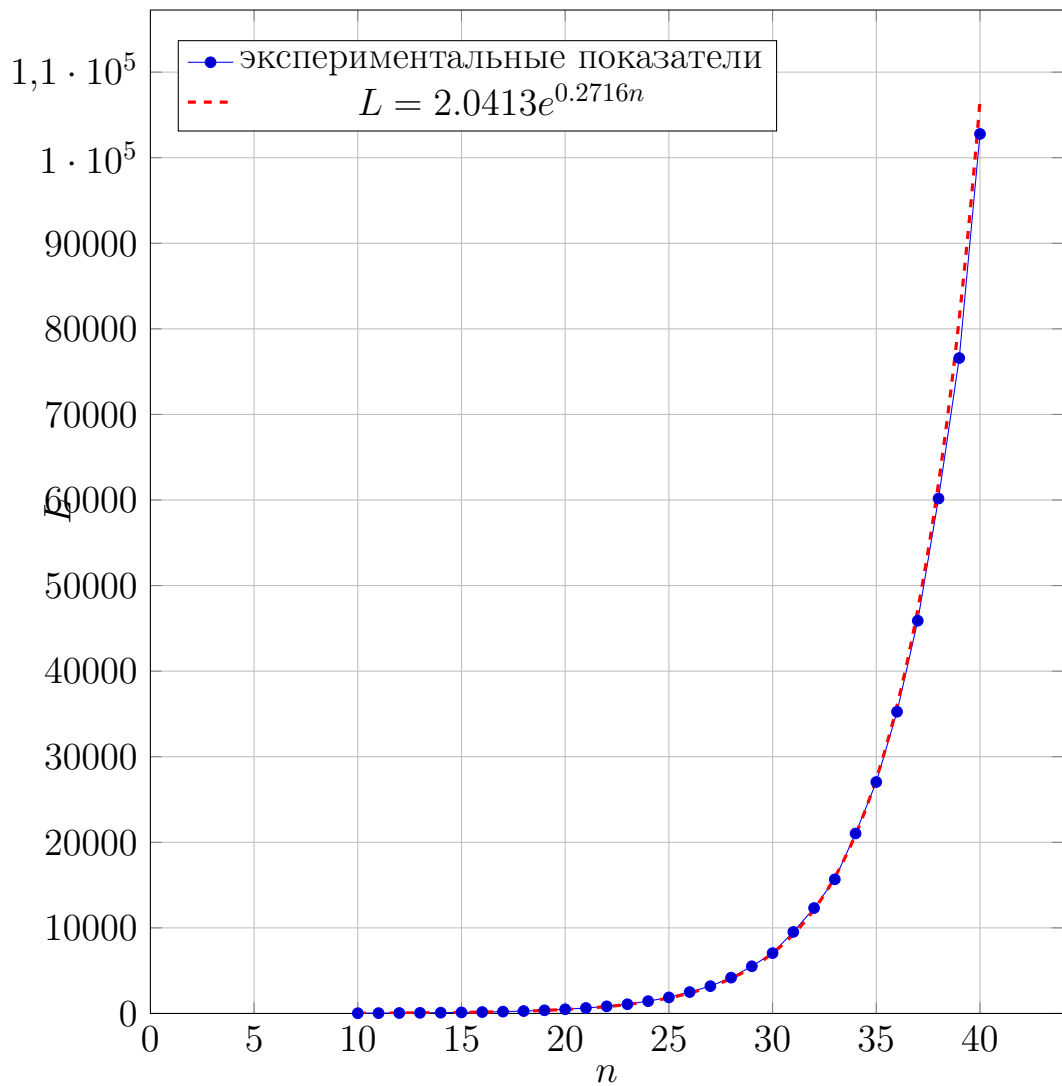


Рисунок 18 — Зависимость максимального количества листьев от количества городов для симметричной задачи с предвычисленным туром

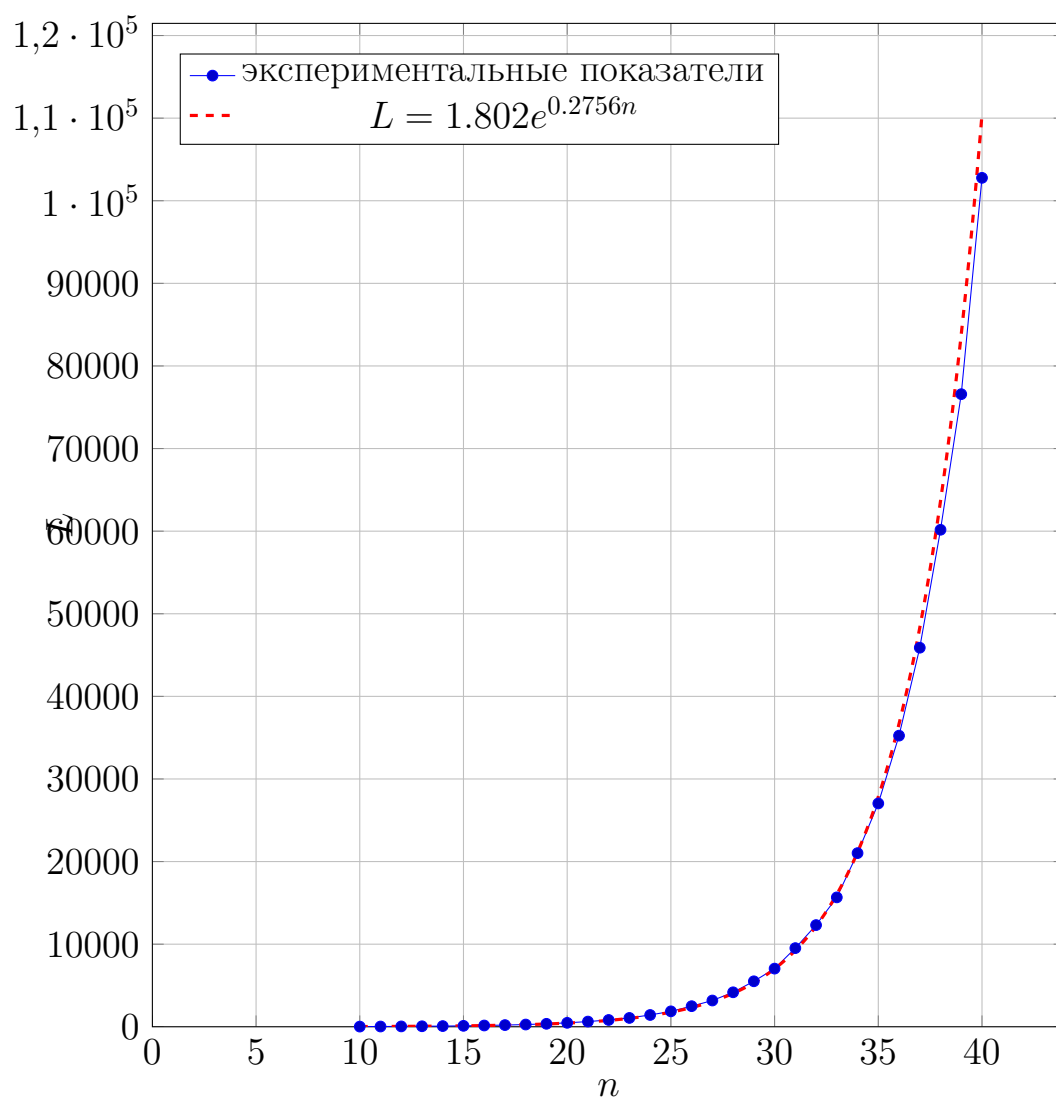
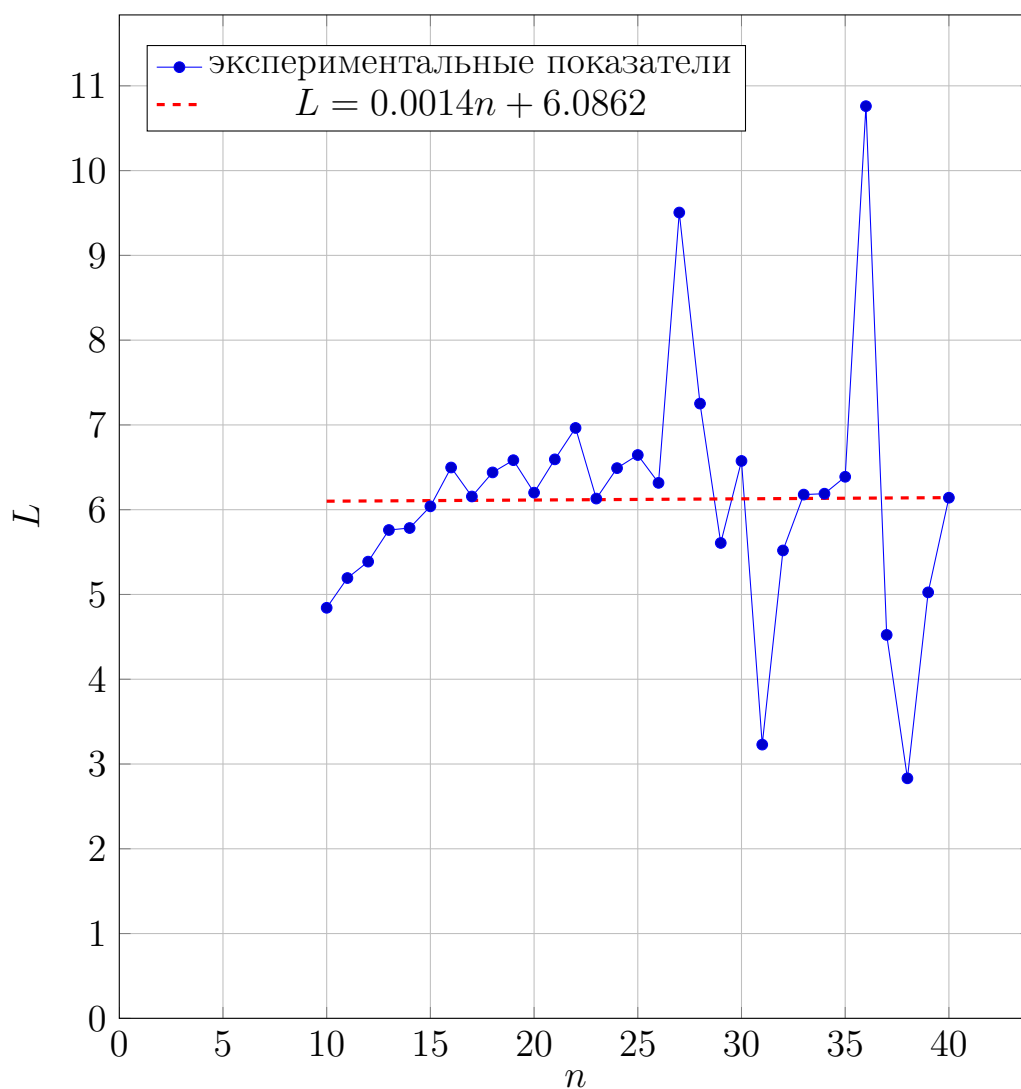


Рисунок 19 — Разность между максимальным количеством листьев для симметричной задачи без предвычисленного тура и с предвычисленным туром



## 6.4 Анализ максимальной глубины дерева решений

На рисунках 20, 21, 22, 23, 24, 25 показана зависимость максимального количества листьев дерева решений от количества городов для разных типов задач коммивояжёра с использованием различных алгоритмов решения.

Рисунок 20 — Зависимость максимальной глубины от количества городов для несимметричной задачи без предвычисленного тура

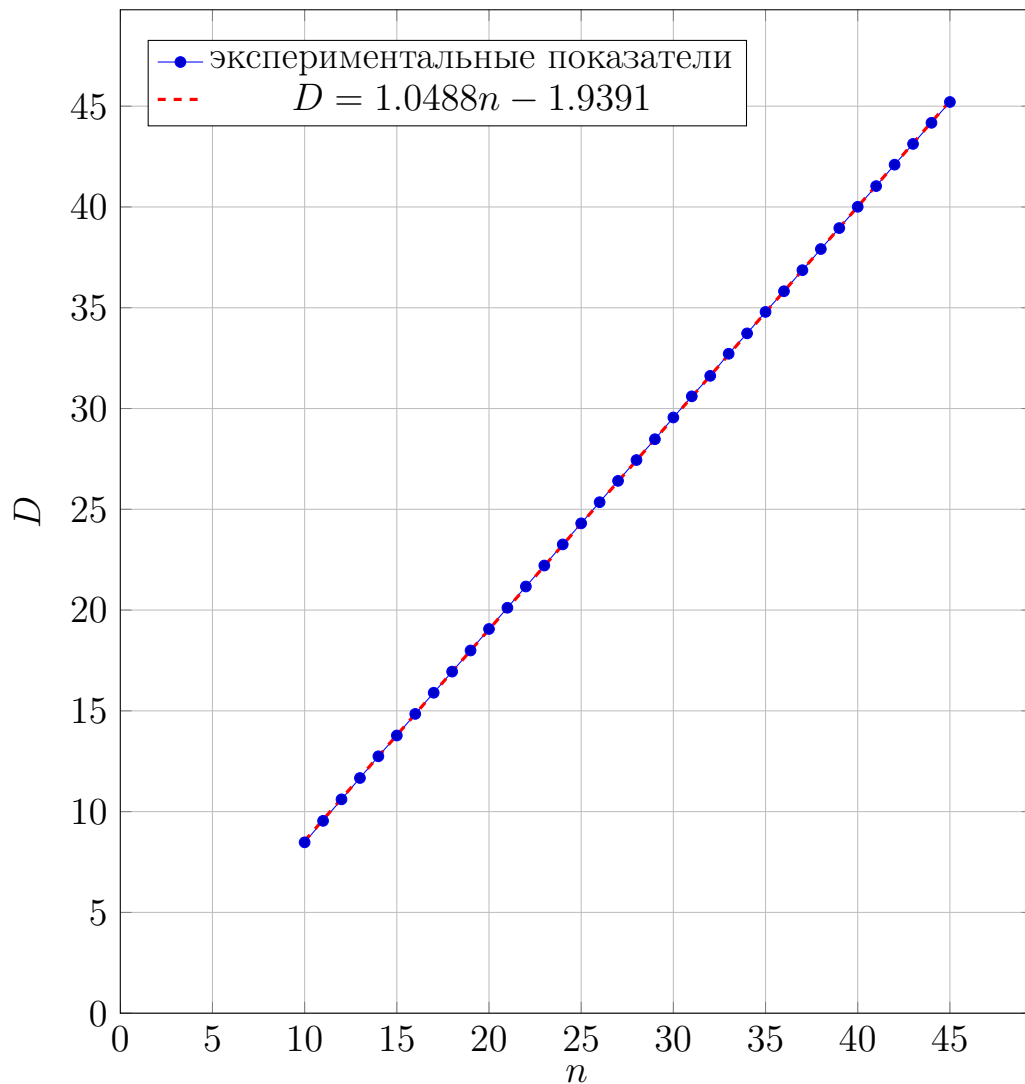


Рисунок 21 — Зависимость максимальной глубины от количества городов для несимметричной задачи с предвычисленным туром

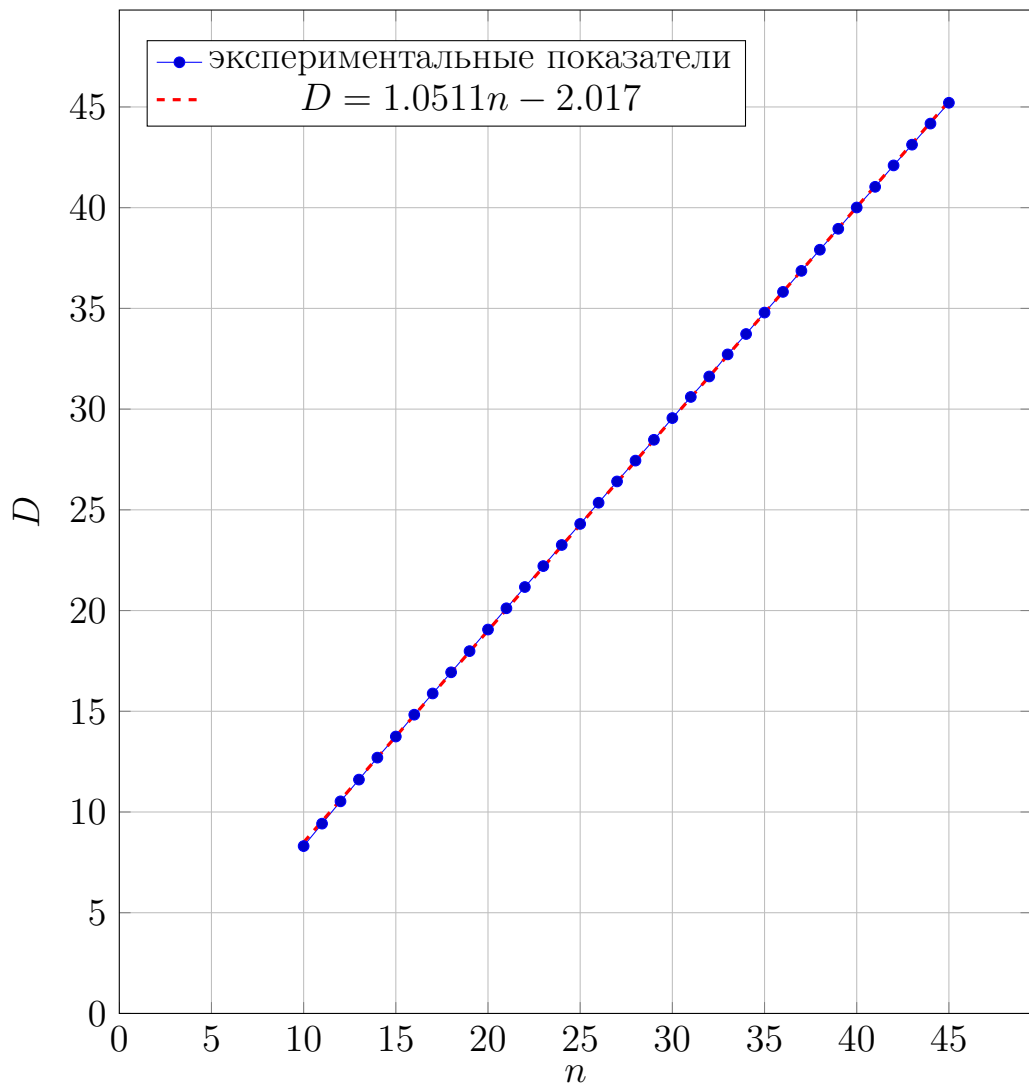


Рисунок 22 — Разность между максимальной глубиной дерева решений для несимметричной задачи без предвычисленного тура и с предвычисленным туром

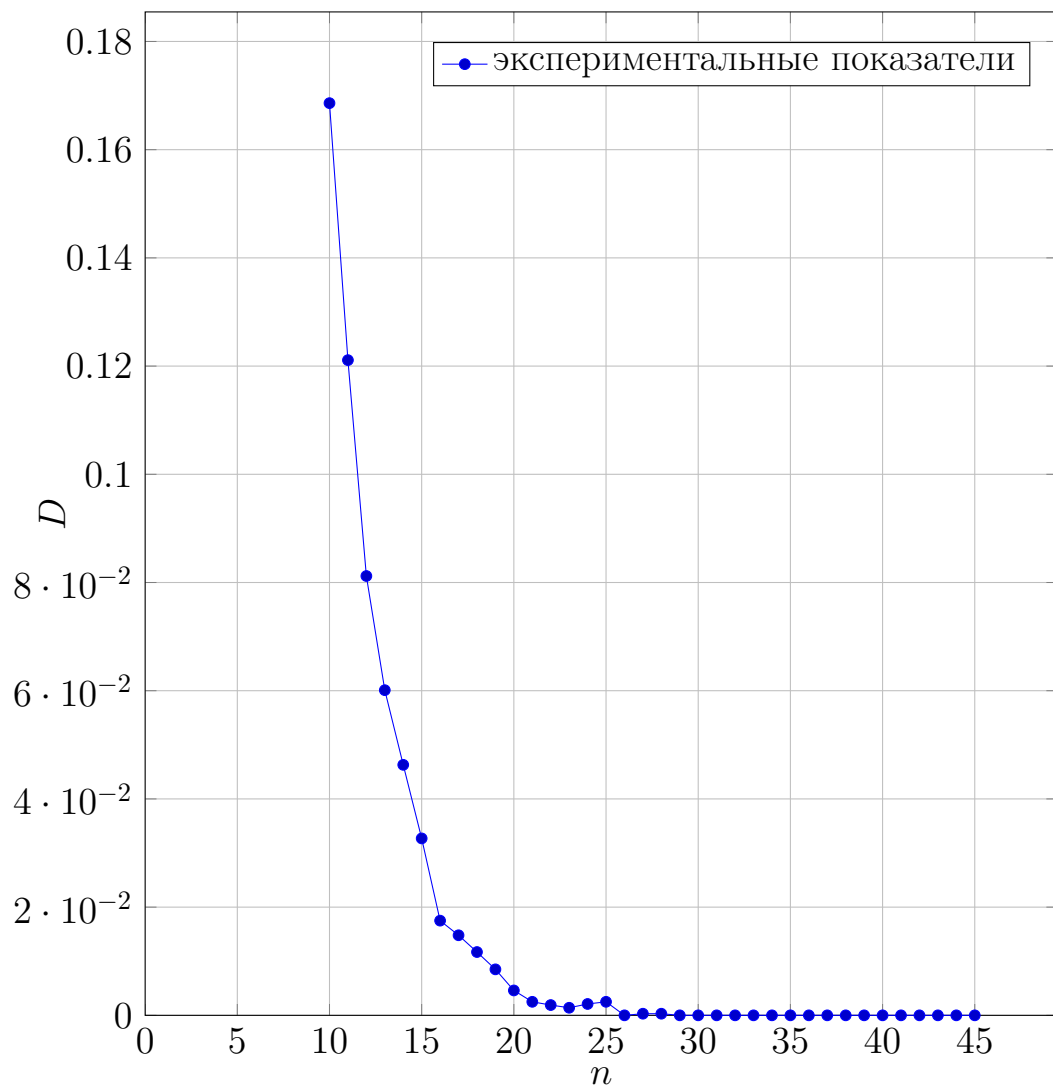


Рисунок 23 — Зависимость максимальной глубины дерева решений от количества городов для симметричной задачи без предвычисленного тура

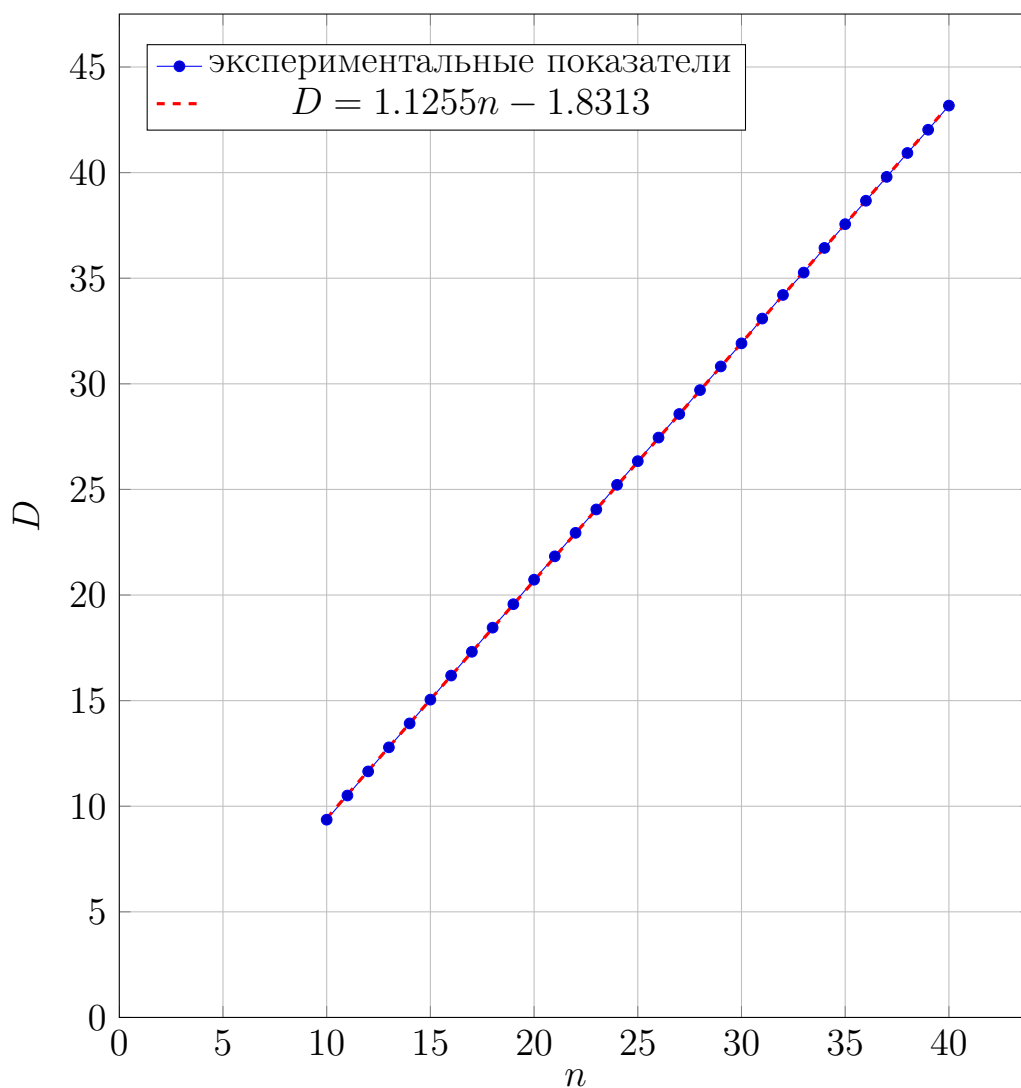


Рисунок 24 — Зависимость максимальной глубины дерева решений от количества городов для симметричной задачи с предвычисленным туром

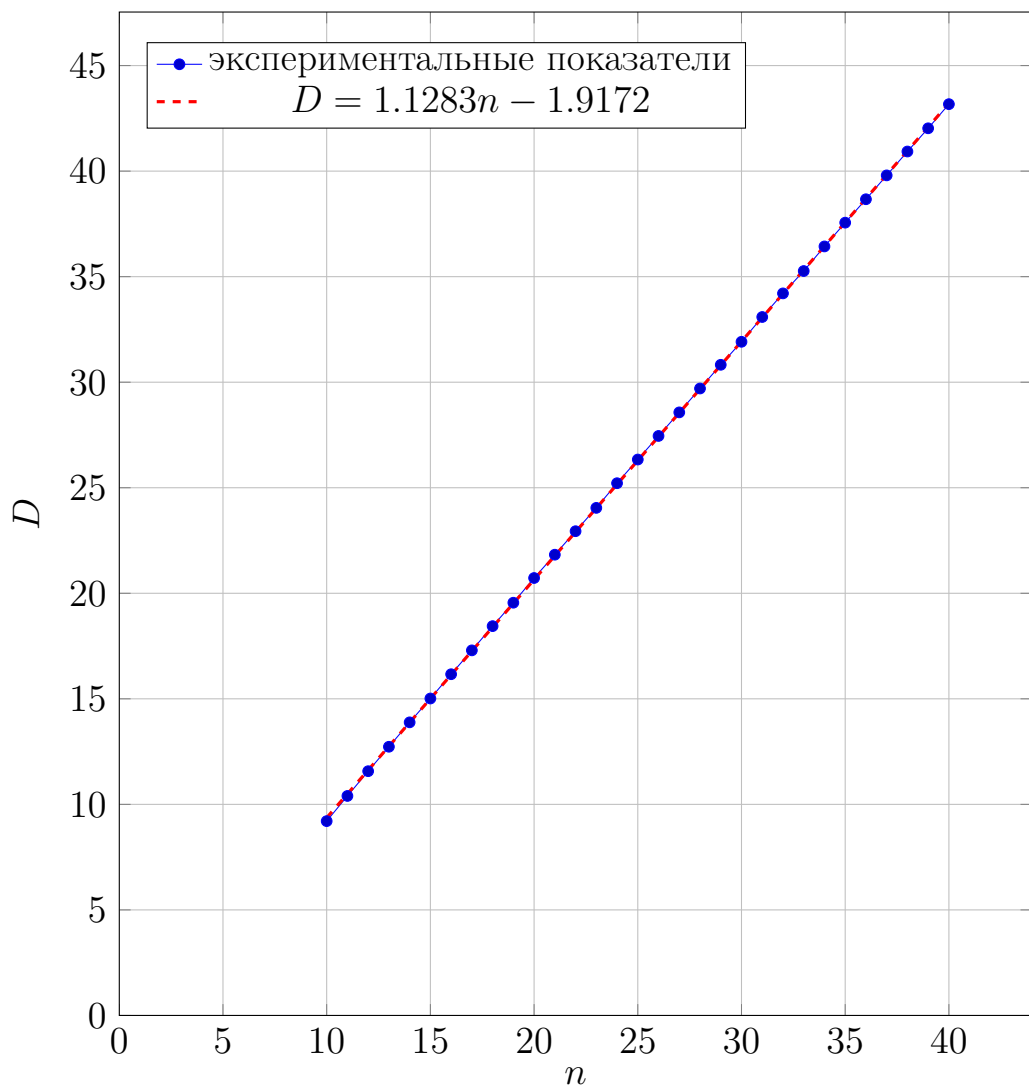
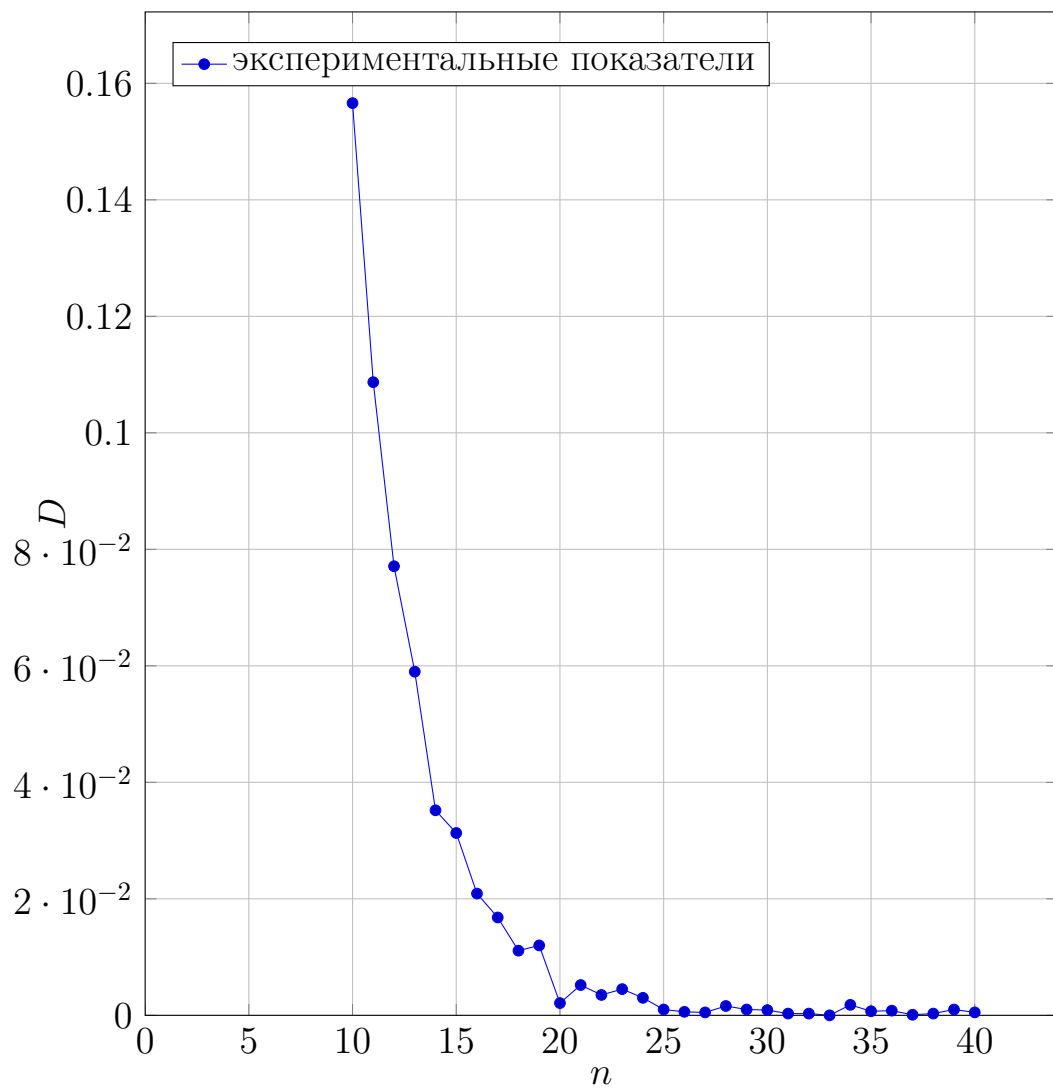




Рисунок 25 — Разность между максимальной глубиной дерева решений для симметричной задачи без предвычисленного тура и с предвычисленным туром



## 6.5 Графическое представление времени работы

На рисунках 26, 27, 28, 29 показана зависимость времени работы алгоритмов от количества городов для разных типов задач коммивояжёра.

Рисунок 26 — Зависимость времени работы алгоритма без предвычисленного тура от количества городов для несимметричной задачи

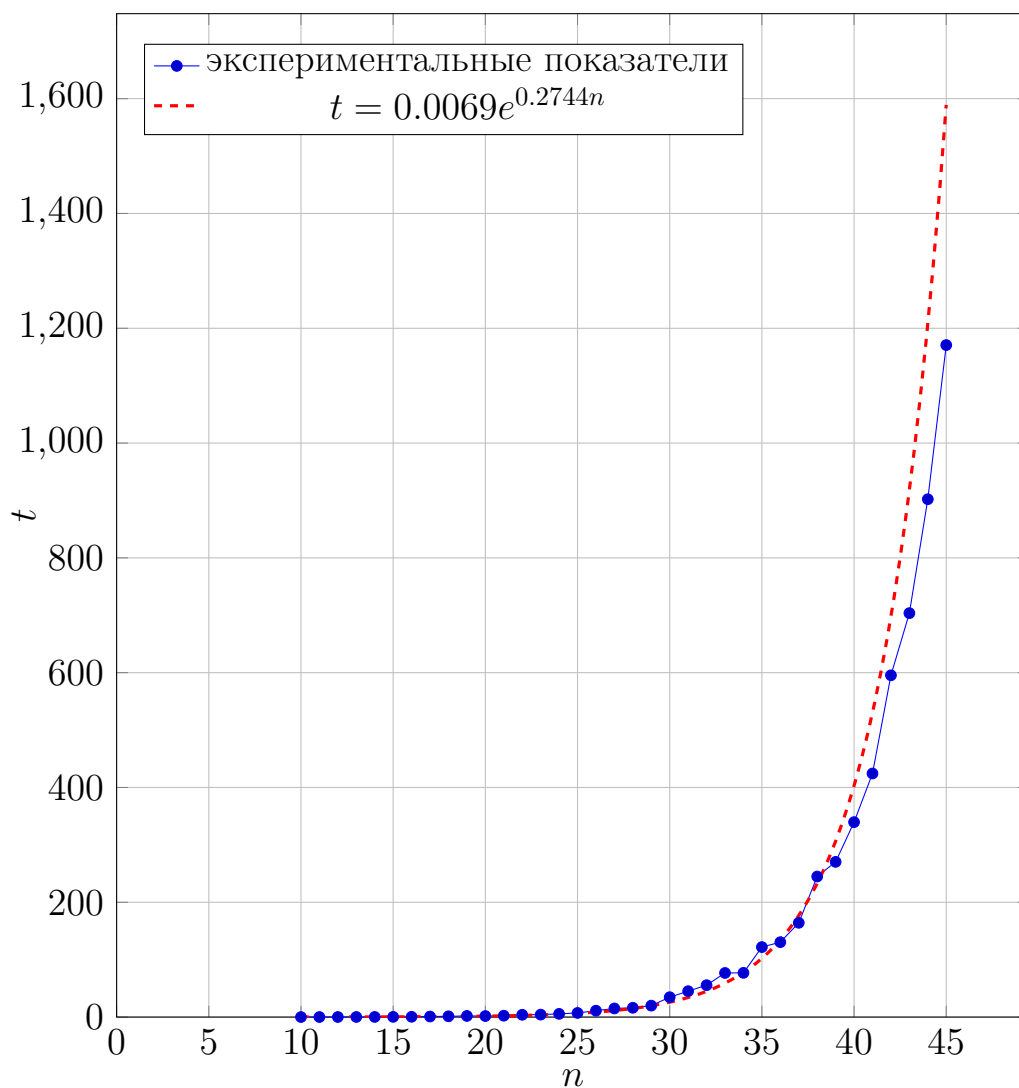


Рисунок 27 — Зависимость времени работы алгоритма с предвычисленным туром от количества городов для несимметричной задачи

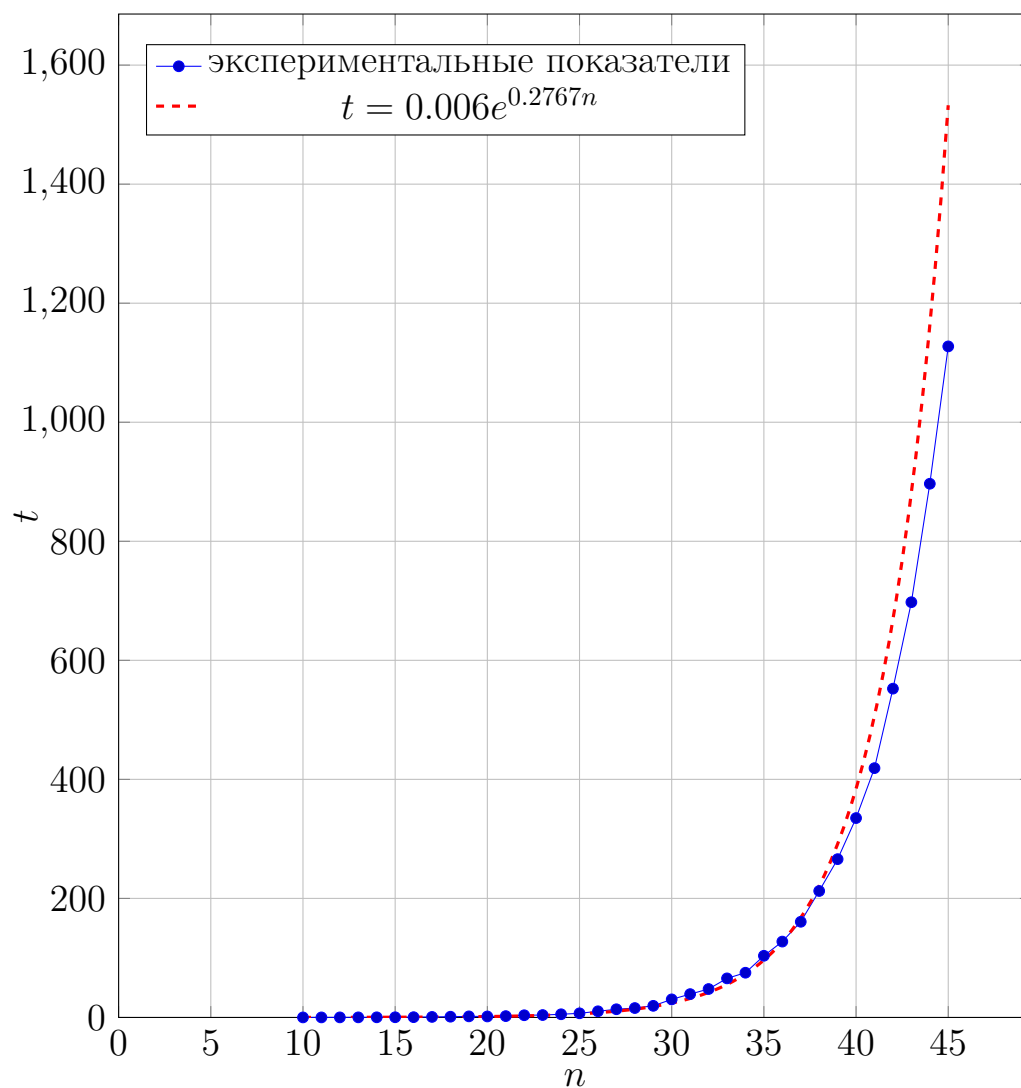


Рисунок 28 — Зависимость времени работы алгоритма без предвычисленного тура от количества городов для симметричной задачи

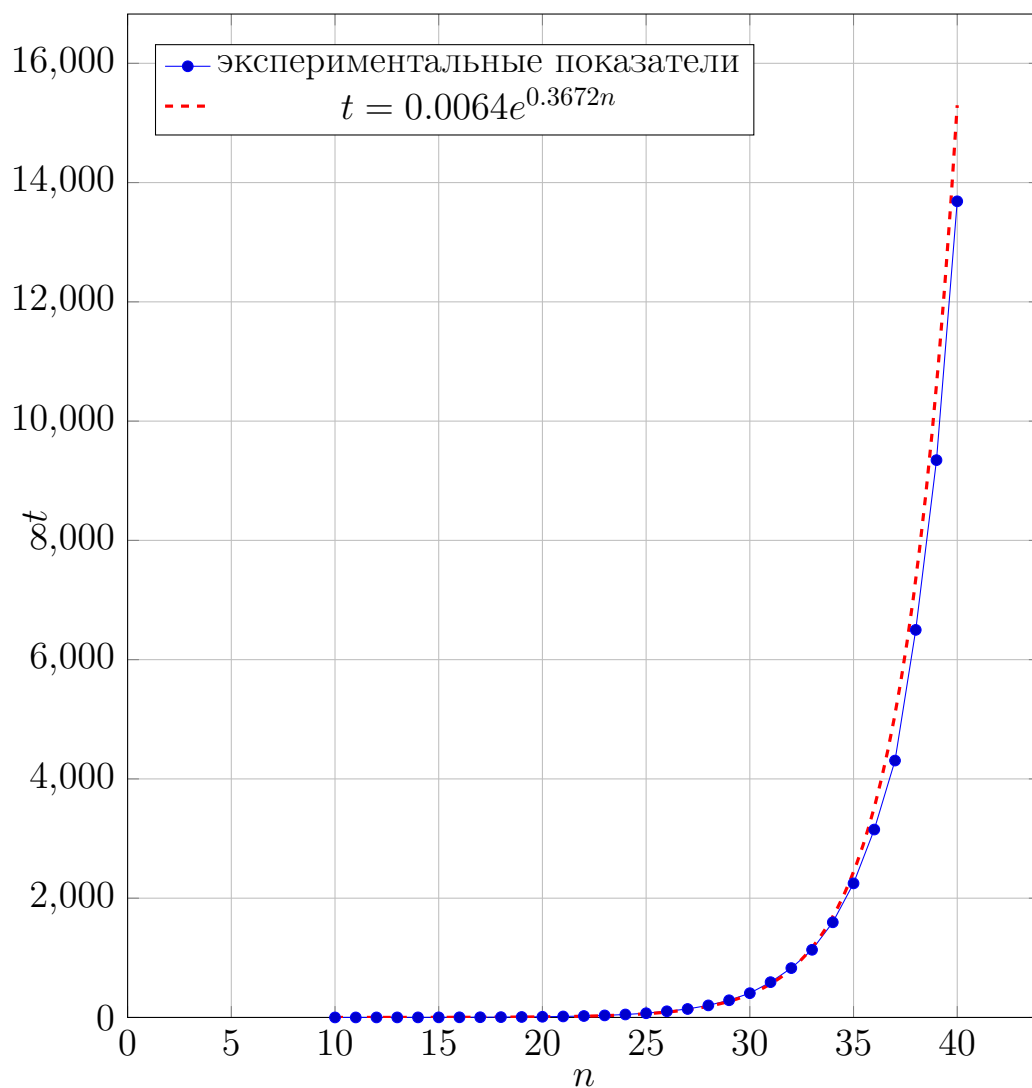
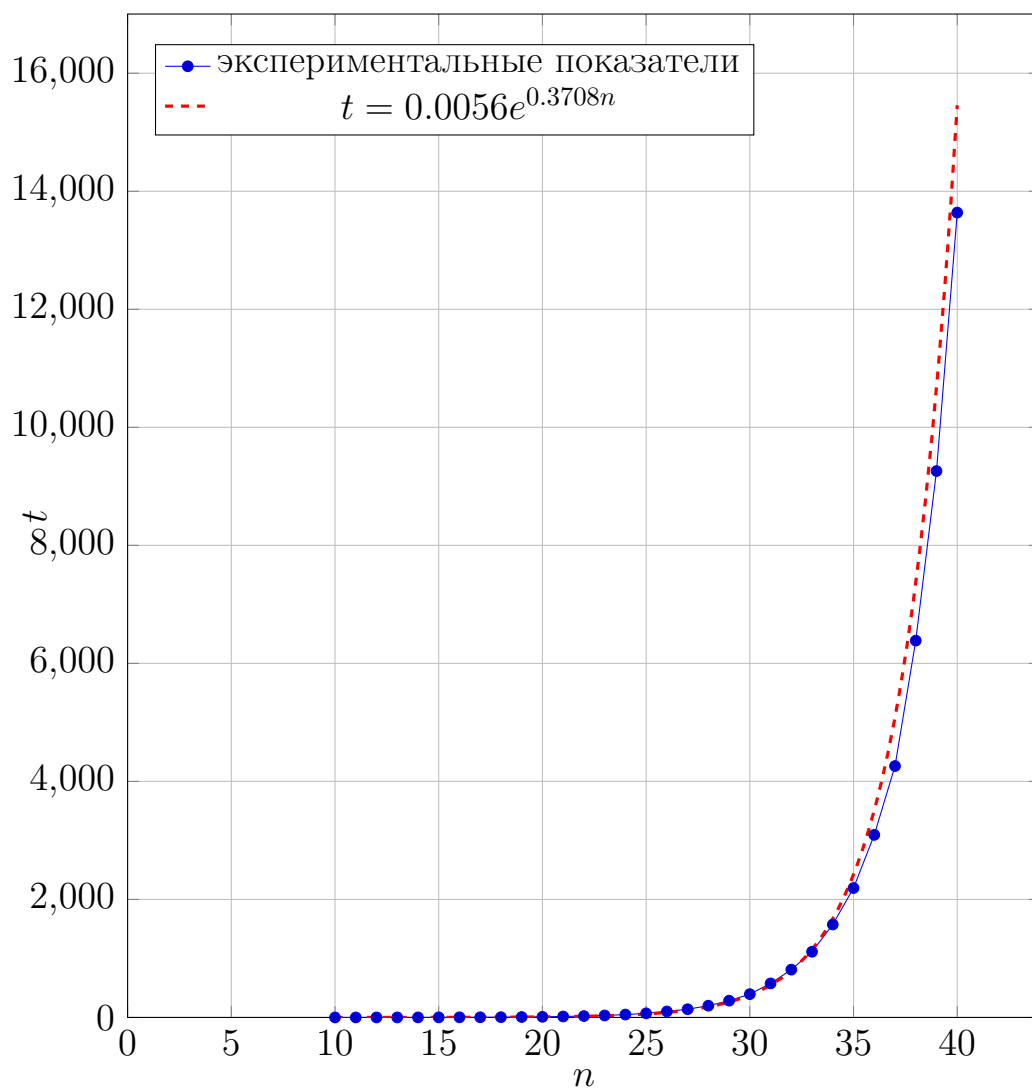


Рисунок 29 — Зависимость времени работы алгоритма с предвычисленным туром от количества городов для несимметричной задачи



## 7 Выводы

На основании полученных экспериментальных результатов и их анализа можно сделать следующие выводы:

- размер дерева решений сильно чувствителен по длине входа и по размаху варьирования при фиксированной длине входа;
- в случаях, когда поиск тура занимает существенно больше времени, чем в среднем для этой длины входа, то порождённое дерево поиска обладает большей шириной, но не глубиной;
- с увеличением длины входа эффективность от предвычисленного тура уменьшается;
- можно предположить, что начиная с какой-то длины входа на глубину дерева не будет влиять предвычисленный тур;
- для симметричной задачи деревья имеют более сложную структуру, чем несимметричной;
- в среднем глубина дерева линейно зависит от длины входа на всем рассматриваемом диапазоне.

## ЗАКЛЮЧЕНИЕ

Поставленные задачи выполнены. Представление о структуре дерева расширилось (более подробно описано в разделе 7). Дальнейшее исследование стоит направить в сторону выбора алгоритма поиска предвычисленного тура.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Дискретная математика и комбинаторика / Д.А. Андерсон. – М. : Издательский дом "Вильямс", 2004 – 960 с.: ил. – Парал. тит. англ. ISBN 5-8459-0498-6 (рус.)
- 2 Ресурсно-эффективные компьютерные алгоритмы. Разработка и анализ. / М.В. Ульянов – М. : ФИЗМАТЛИТ, 2008 – 304 с. ISBN 978-5-9221-0950-5
- 3 А.С. Ермошин, В.А. Плиско Исследование дерева решений метода ветвей и границ в задаче коммивояжера // Программное и информационное обеспечение систем различного назначения на базе персональных ЭВМ: Межвузовский сборник научных трудов / Под ред. д. т. н., проф. Михайлова Б. М. — М.: МГАПИ, 2006. Вып. 9. С. 76–82.
- 4 Алгоритмы: построение и анализ. /Т. Кормен [и др.]; 3-е издание: Пер. с англ. — М.: Издательский дом "Вильямс", 2014. — 1328 с. :ил. — Парал. тит. англ. ISBN 978-5-8459-1794-2 (рус.)
- 5 Land A.H., Doig A.G. An automatic method of solving discrete programming problems // *Econometrica*. – 1960. – V. 28 №3. – P. 497-520.
- 6 Little J.D.C., Murty K.G., Sweeney D.W., Karel C. An algorithm for the traveling salesman problem // *Operat. Res.*, 11 (1963), pp.972-989.
- 7 Введение в разработку и анализ алгоритмов. / С. Гудман, С. Хидетними — М.: Мир, 1981. — 368 с.
- 8 Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы / И.Х. Сигал, А.П. Иванова. – 2-е изд. б испр и доп. – М.: ФИЗМАТЛИТ, 2007. –304 с. – ISBN 978-5-9221-0808-9
- 9 Алгоритмы и структуры данных. Новая версия для Оберона / Н. Вирт Пер. с англ. Ткачев Ф.В. – М.: ДМК Пресс, 2010. – 272 с.



## ПРИЛОЖЕНИЕ А

### 8 Технические и программные особенности компьютера

Эксперименты проводились на персональном компьютере со следующими характеристиками:

- процессор Intel i7 3770К 3800 ГГц;
- оперативная память Kingston КНХ1600С9D3Р1 16 ГБ;
- материнская плата GIGABYTE GA-Z77X-D3Н;
- операционная система: Fedora 21 workstation.