



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Московский государственный технический университет имени  
Н. Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н. Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

## Отчёт по лабораторной работе №7 по дисциплине «Анализ алгоритмов»

**Тема:** Решение задачи коммивояжёра методом полного перебора и муравьиным алгоритмом

**Студент:** Карпова Е. О.

**Группа:** ИУ7-52Б

**Оценка (баллы):** \_\_\_\_\_

**Преподаватели:** Волкова Л. Л., Строганов Ю. В.

Москва — 2022 г.

# Оглавление

<b>Введение</b>	<b>3</b>
<b>1. Аналитическая часть</b>	<b>4</b>
1.1. Задача коммивояжёра . . . . .	4
1.1.1. Алгоритм полного перебора . . . . .	5
1.1.2. Муравьиный алгоритм . . . . .	5
<b>2. Конструкторская часть</b>	<b>8</b>
2.1. Разработка алгоритма полного перебора для решения задачи коммивояжёра	8
2.2. Разработка муравьиного алгоритма для решения задачи коммивояжёра . . .	9
2.3. Модель вычислений . . . . .	12
2.3.1. Трудоёмкость алгоритма полного перебора . . . . .	13
2.4. Трудоёмкость муравьиного алгоритма . . . . .	13
<b>3. Технологическая часть</b>	<b>17</b>
3.1. Требования к ПО . . . . .	17
3.2. Средства реализации . . . . .	17
3.3. Реализация алгоритмов . . . . .	18
3.4. Тестирование . . . . .	22
<b>4. Экспериментальная часть</b>	<b>23</b>
4.1. Технические характеристики . . . . .	23
4.2. Параметризация реализации муравьиного алгоритма . . . . .	23
4.2.1. Параметризация для графа №1 . . . . .	24
4.2.2. Параметризация для графа №2 . . . . .	25
4.2.3. Параметризация для графа №3 . . . . .	26
4.3. Измерение времени выполнения реализаций алгоритмов . . . . .	27
<b>Заключение</b>	<b>30</b>
<b>Список использованных источников</b>	<b>31</b>
<b>Приложение А</b>	<b>33</b>

# Введение

В октябре 1985 года появилась первая в мировой литературе монография [9], посвященная целиком задаче коммивояжёра. Она была написана большим коллективом известных специалистов [10]. Сейчас задача коммивояжёра занимает центральное место среди труднорешаемых задач дискретной оптимизации [10].

Задача имеет разные формулировки, но в контексте данной лабораторной работы будет рассмотрена графовая: необходимо найти кратчайший путь прохода по всем заданным пунктам такой, чтобы каждый пункт был посещён ровно один раз и конечным пунктом оказался тот, с которого был начат обход. Условия данной задачи можно реализовать при помощи взвешенного графа, вершины которого представляют пункты, а веса рёбер — расстояния между соответствующими пунктами [11]. Тогда задачу можно свести к поиску кратчайшего Гамильтонова цикла [12] для неориентированного графа.

Цель работы: получение навыков программирования, тестирования полученного программного продукта и проведения замеров времени выполнения по результатам работы программы на примере реализации решения задачи коммивояжёра при помощи алгоритма полного перебора и муравьиного алгоритма.

Задачи работы:

- 1) изучение теоретических основ задачи коммивояжёра, алгоритма полного перебора и муравьиного алгоритма;
- 2) описание алгоритма полного перебора и муравьиного алгоритма;
- 3) реализация данных алгоритмов;
- 4) выполнение оценки трудоёмкости реализаций алгоритмов;
- 5) проведение параметризации для муравьиного алгоритма;
- 6) проведение замеров времени работы (в мкс) данных алгоритмов на наилучшей комбинации параметров;
- 7) получение графической зависимости измеряемой величины от количества вершин в графе;
- 8) проведение сравнительного анализа двух представленных реализаций на основе полученной зависимости.

# 1. Аналитическая часть

В данном разделе будут рассмотрены теоретические основы задачи коммивояжёра, алгоритма полного перебора и муравьиного алгоритма для решения данной задачи.

## 1.1. Задача коммивояжёра

Будет рассматриваться графовая формулировка задачи коммивояжёра: необходимо найти кратчайший путь прохода по всем заданным пунктам такой, чтобы каждый пункт был посещён ровно один раз и конечным пунктом оказался тот, с которого был начат обход [11].

В данной работе условия задачи реализуются при помощи взвешенного неориентированного графа, вершины которого представляют города Африки, а веса рёбер — расстояния между соответствующими городами. Вес ребра будет больше, если прямой путь между городами лежит через пустыню, и меньше, если путь проходит через водоём (кроме рек).

На карте Африки были выбраны города, соединённые линией на рисунке 1.1.

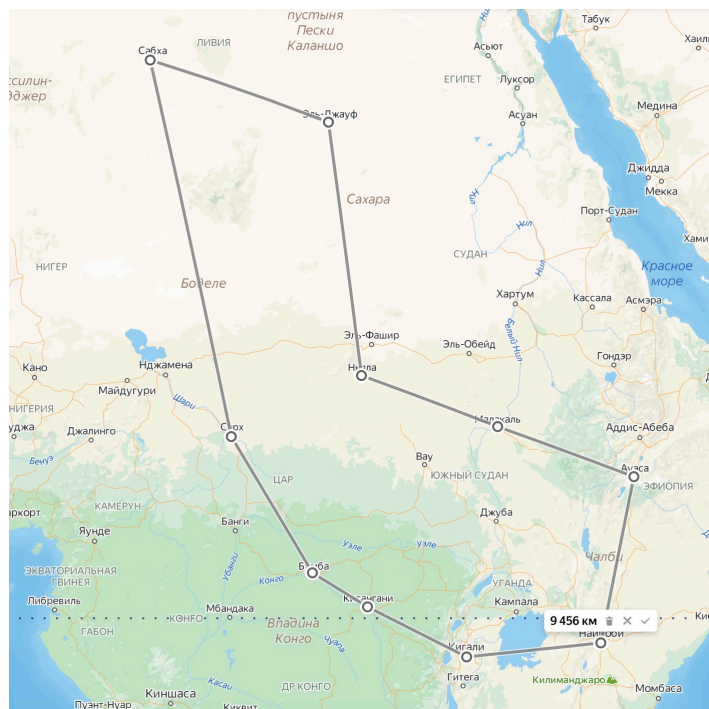


Рисунок 1.1 — Участок карты Африки с выбранными городами [13]

Тогда задача сводится к поиску кратчайшего Гамильтонова цикла, то есть простого элементарного цикла, проходящего через каждую вершину графа [12], для неориентированного графа.

### 1.1.1. Алгоритм полного перебора

Алгоритм полного перебора заключается в переборе всех возможных комбинаций последовательности посещаемых пунктов. Так как число возможных Гамильтоновых циклов на графе конечно, применение этого алгоритма допустимо.

На вход алгоритму подаётся число городов и матрица смежности графа. Если положить количество городов равным  $N$ , то в результате работы алгоритма будут рассмотрены все возможные перестановки из  $N - 1$  элементов. В контексте данной задачи элементами будут номера населённых пунктов — вершин графа. Для каждой перестановки рассчитывается суммарная длина перемещений. В результате будет выбрана последовательность посещения населённых пунктов такая, при которой суммарное перемещение будет наименьшим.

### 1.1.2. Муравьиный алгоритм

Муравьиный алгоритм основывается на имитации механизмов самоорганизации муравьев, обеспечивающих достижение системой определённой цели благодаря взаимодействию низкоуровневых элементов [14]. Эти элементы, то есть муравьи, коммуницируют за счёт стигмерджей — тип взаимодействия, при котором одна сторона изменяет определённую часть окружающей среды, а остальные используют эту информацию, когда находятся в данной окрестности. Биологически это осуществимо через выделение муравьём феромона — стойкого вещества, которое, будучи воспринятым другими муравьями, указывает им, куда стоит двигаться, подчиняясь стадному инстинкту. Чем больше феромона в определённой области, тем больше муравьёв ранее было в этой области. Соответственно привлекательность этой области для следующего муравья повышается.

В ходе решения задачи моделируется решение задачи коммивояжёра муравьиной колонией за  $t$  суток. Пусть задан граф из  $N$  городов и дана его матрица смежности  $D$ . Полагается, что колония обладает некоторой глобальной памятью — информация о распределении феромона по рёбрам графа должна где-то храниться, например, в виде матрицы. Матрица феромона заполняется некоторым небольшим значением по умолчанию. В начале каждых суток каждый муравей колонии, размер которой устанавливается равным количеству вершин графа, то есть  $N$ , размещается в одном из городов. Начальные позиции муравьёв не совпадают. За день каждый муравей проходит полный Гамильтонов цикл по графу, действуя независимо от других особей.

Для муравья устанавливаются следующие компетенции:

- Зрение — означает, что муравей может оценить ребро по его длине: более длинное ребро будет менее привлекательным для муравья. В связи с этим, желание муравья посетить  $j$ -ый город описывается соотношением

$$\eta_{ij} = \frac{1}{D_{ij}}, \quad \text{где } D_{ij} \text{ — длина ребра между пунктами } i \text{ и } j. \quad (1.1)$$

- Память — означает, что муравей помнит, какие пункты посещал ранее. Уже посещённые пункты вычёркиваются из списка непосещённых пунктов, в который изначально заносятся номера всех вершин графа, кроме начальной позиции данного муравья. Далее список непосещённых вершин для  $k$ -ого муравья будет обозначаться  $J_k$ .
- Обоняние — означает, что муравей способен чувствовать концентрацию феромона  $\tau_{ij}$  на ребре между пунктами  $i$  и  $j$ . Тогда чем выше концентрация феромона на ребре, тем больше желание муравья пойти по данному ребру.

Учитывая перечисленные компетенции можно вывести формулу для расчёта вероятности посещения  $k$ -ым муравьём, находящимся в вершине  $i$ , вершины  $j$ :

$$P_{ij,k}(t) = \begin{cases} 0, & \text{если } j \in J_k, \\ \frac{[\eta_{ij}]^\beta \cdot [\tau_{ij}]^\alpha}{\sum_{q \notin J_k} [\eta_{iq}]^\beta \cdot [\tau_{iq}]^\alpha}, & \text{иначе.} \end{cases} \quad (1.2)$$

В формуле (1.2)  $\alpha$  — это коэффициент стадности, задающий вес концентрации феромона при вычислении вероятности, а  $\beta$  — коэффициент жадности, определяющий вес длины ребра, причём  $\alpha, \beta \in (0, 1)$ . В случае, когда  $\alpha = 0$ , алгоритм вырождается в жадный, то есть всегда будет выбираться ближайший к текущей позиции город. В случае, когда  $\beta = 0$ , алгоритм становится стадным, что влечёт за собой преждевременный приход к одному субоптимальному решению.

Если муравей сумел пройти маршрут, удовлетворяющий понятию Гамильтонова цикла, на пройденных им рёбрах должна возрасти концентрация феромона в соответствии со следующей формулой:

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij,k}(t), \quad (1.3)$$

где  $\rho \in (0, 1)$  — коэффициент испарения феромона с течением времени,  $m$  — количество муравьёв в колонии,  $\tau_{ij}(t+1)$  — концентрация феромона на ребре между пунктами  $i$  и  $j$  в наступающий день,  $\tau_{ij}(t)$  — концентрация феромона на ребре между пунктами  $i$  и  $j$  в текущий день.

$j$  в уходящий день,  $\tau_{ij,k}(t)$  — концентрация феромона на ребре между пунктами  $i$  и  $j$  в наступающий день для  $k$ -ого муравья, а  $\Delta\tau_{ij,k}(t)$  определяется по формуле

$$\Delta\tau_{ij,k}(t) = \begin{cases} \frac{Q}{L_k(t)}, & (i, j) \in T_k(t), \\ 0, & (i, j) \notin T_k(t). \end{cases} \quad (1.4)$$

В формуле (1.4)  $Q$  — параметр, имеющий значение порядка длины оптимального пути,  $L_k(t)$  — длина маршрута  $k$ -ого муравья,  $T_k(t)$  — маршрут, пройденный  $k$ -ым муравьём к моменту времени  $t$ .

Значение  $Q$  может быть рассчитано единожды в начале работы алгоритма по формуле

$$Q = \frac{\sum_{i=1}^n \sum_{j=1}^n D_{ij}}{N}. \quad (1.5)$$

Формулой (1.3) учитывается естественное испарение феромона за ночь и его прирост за счёт проходов муравьиной колонии за предыдущий день. В случае, если значение феромона на ребре стало слишком мало, нужно вновь восстановить его до значения по умолчанию, иначе некоторые непосещённые пункты станут недостижимыми.

Для улучшения временных характеристик вводят элитных муравьев, которые дополнительно усиливают рёбра наилучшего маршрута, найденного с начала работы алгоритма. Для  $e$  элитных муравьёв рёбра наилучшего маршрута получают следующее усиление концентрации феромона:

$$\Delta\tau_e = e \cdot \frac{Q}{L^+}, \quad \text{где } L^+ \text{ — длина наилучшего маршрута.} \quad (1.6)$$

В результате работы алгоритма находится кратчайший маршрут среди найденных муравьями. Полученный в результате работы алгоритма маршрут не обязательно совпадёт с решением, которое можно получить с помощью алгоритма полного перебора, ввиду неточности муравьиного алгоритма [14]. Он гарантирует лишь получение близкого к идеальному значения.

## 2. Конструкторская часть

В данном разделе будут представлены схемы реализаций алгоритма полного перебора и муравьиного алгоритма для решения задачи коммивояжёра.

### 2.1. Разработка алгоритма полного перебора для решения задачи коммивояжёра

На рисунке 2.1 представлена схема реализации алгоритма полного перебора для решения задачи коммивояжёра.

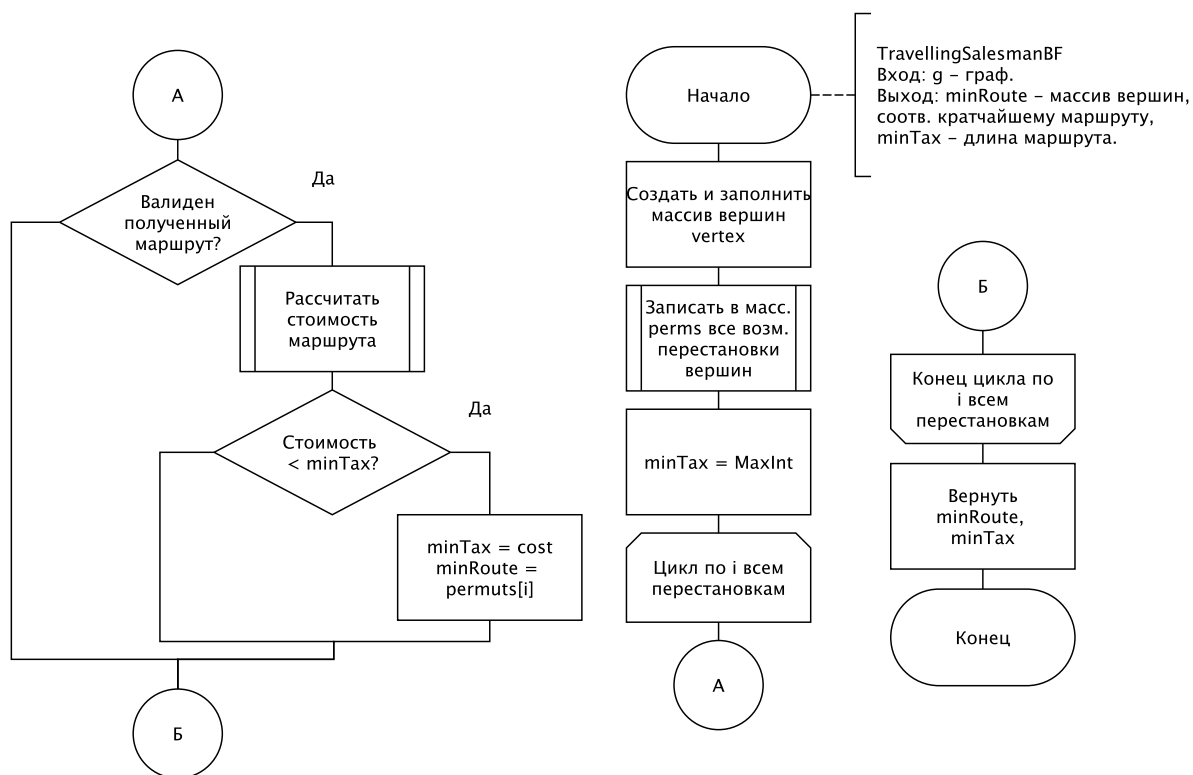


Рисунок 2.1 — Схема реализации алгоритма полного перебора для решения задачи коммивояжёра



## 2.2. Разработка муравьиного алгоритма для решения задачи коммивояжёра

На рисунке 2.2 представлена схема реализации муравьиного алгоритма для решения задачи коммивояжёра, а на рисунках 2.3 – 2.4 — схемы необходимых для его реализации подпрограмм: поиска муравьём маршрута и поиска муравьём следующей вершины соответственно.

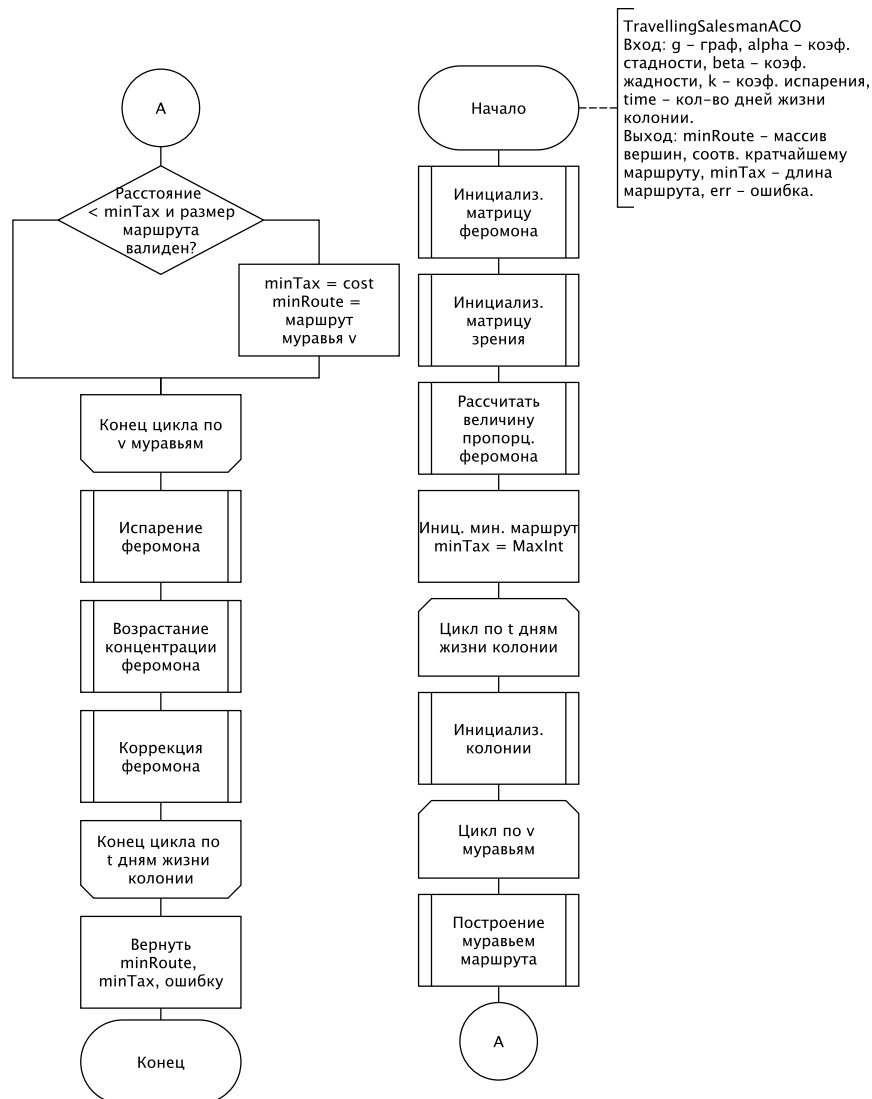


Рисунок 2.2 — Схема реализации муравьиного алгоритма для решения задачи коммивояжёра

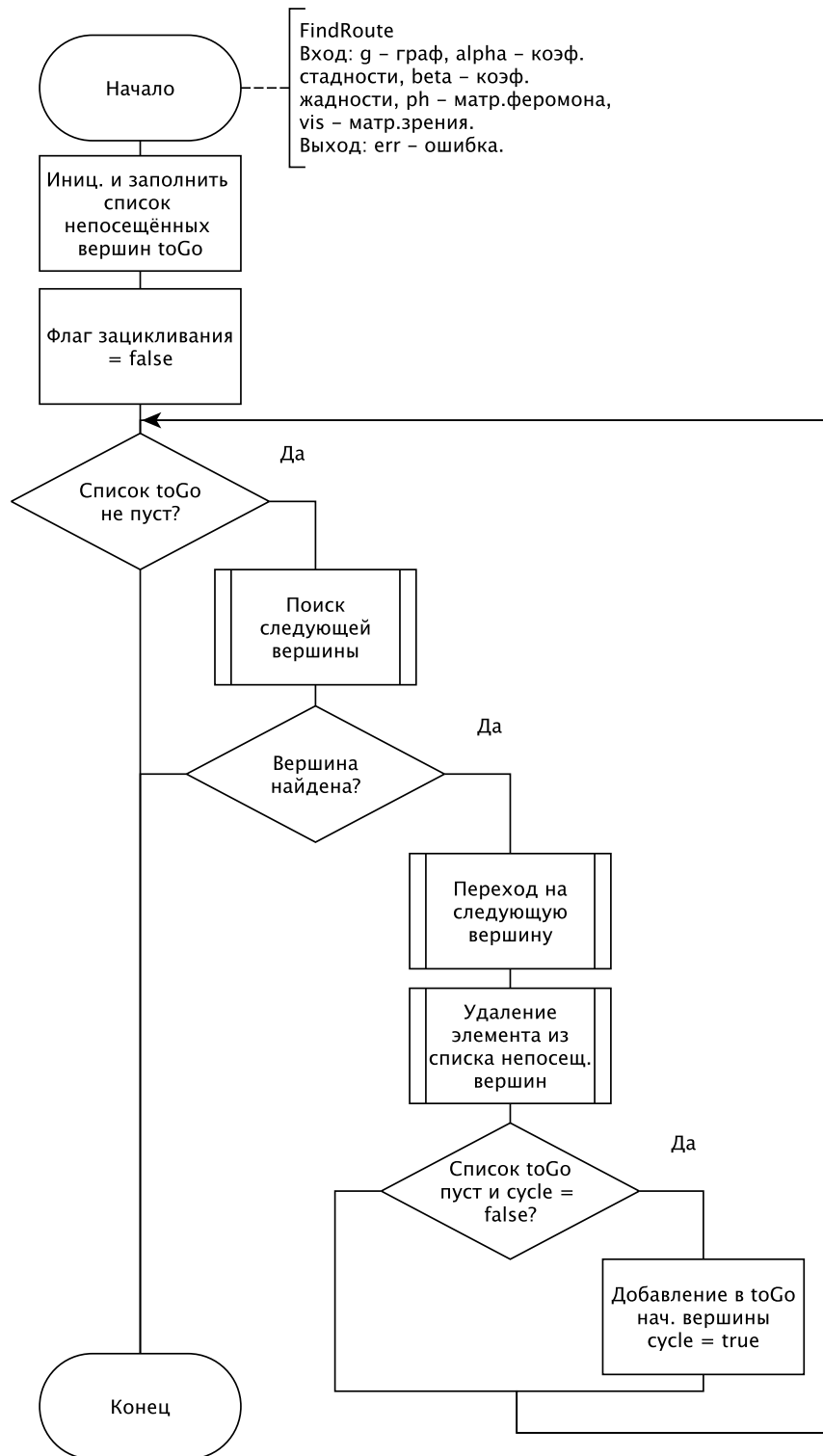


Рисунок 2.3 — Схема реализации алгоритма поиска муравьём маршрута

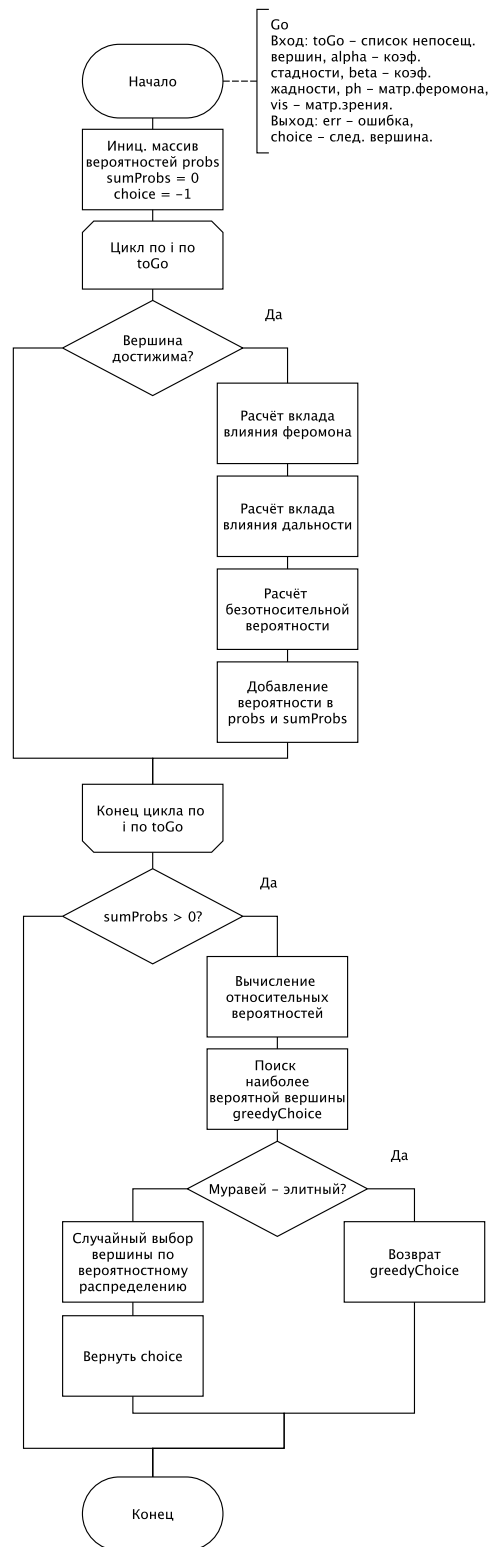


Рисунок 2.4 — Схема реализации алгоритма поиска муравьём следующей вершины

## 2.3. Модель вычислений

Для вычисления трудоёмкости данных алгоритмов необходимо ввести модель вычислений.

Обозначим трудоёмкость как  $f_a$ , где  $a$  — индекс, указывающий операцию, блок кода или оператор, для которого вычисляется трудоёмкость.

Определим трудоёмкость базовых операций как:

$$\begin{array}{llll}
 f_+ = 1 & f_- = 1 & f_{+=} = 1 & f_{-=} = 1 \\
 f_{:=} = 1 & f_{<<} = 1 & f_{>>} = 1 & f_{[]} = 1 \\
 f_{++} = 1 & f_{--} = 1 & f_{>} = 1 & f_{<} = 1 \\
 f_{>=} = 1 & f_{<=} = 1 & f_{!=} = 1 & f_{==} = 1 \\
 f_{.} = 2 & f_{/} = 2 & f_{\%} = 2 & 
 \end{array} \tag{2.1}$$

Определим трудоёмкость вызова функции как 0.

Определим трудоёмкость условия как

$$f_{if} = f_{cc} + \begin{cases} \min(f_1, f_2), & \text{в лучшем случае,} \\ \max(f_1, f_2), & \text{в худшем случае,} \end{cases} \tag{2.2}$$

где приняты следующие обозначения:

- $f_{cc}$  — трудоёмкость вычисления условия;
- $f_1$  — трудоёмкость блока после *if*;
- $f_2$  — трудоёмкость блока после *else*.

Определим трудоёмкость цикла как

$$f_{loop} = f_{init} + f_{first-cmp} + n \cdot (f_{body} + f_{inc} + f_{cmp}), \tag{2.3}$$

где приняты следующие обозначения:

- $f_{init}$  — трудоёмкость инициализации;
- $f_{first-cmp}$  — трудоёмкость первого сравнения;
- $f_{body}$  — трудоёмкость тела цикла;
- $n$  — количество итераций цикла;
- $f_{inc}$  — трудоёмкость изменения индекса;
- $f_{cmp}$  — трудоёмкость сравнения.

### 2.3.1. Трудоемкость алгоритма полного перебора

Трудоемкость алгоритма полного перебора решения задачи коммивояжера состоит из следующих составляющих.

Трудоемкость внешнего цикла по  $i \in [0..N! - 1]$ :

$$f_{loop} = 2 + N! \cdot (2 + f_{body}) \quad (2.4)$$

Далее рассмотрена трудоемкость для тела цикла в зависимости от результата проверки условия на существование заданного пути в графе:

$$f_{body} = f_{if} + \begin{cases} f_{true}, & \text{если путь существует,} \\ 0, & \text{иначе.} \end{cases} \quad (2.5)$$

Трудоемкость проверки условия:

$$f_{if} = 2 + N \cdot (2 + 8). \quad (2.6)$$

Трудоемкость действий при выполнении условия:

$$f_{true} = 1 + 2 + N \cdot (2 + 10). \quad (2.7)$$

Тогда трудоемкость алгоритма полного перебора решения задачи коммивояжера для лучшего случая (путь в графе не существует) составит

$$f_{brute\_force} = 2 + N! \cdot (2 + 2 + N \cdot (2 + 8)) = 2 + 4N! + 10N \cdot N! = O(N!), \quad (2.8)$$

а трудоемкость для худшего случая (путь в графе существует):

$$\begin{aligned} f_{brute\_force} &= 2 + N! \cdot (2 + 2 + N \cdot (2 + 8) + 1 + 2 + N \cdot (2 + 10)) = \\ &= 2 + 7N! + 22N \cdot N! = O(N!). \end{aligned} \quad (2.9)$$

## 2.4. Трудоемкость муравьиного алгоритма

Введём некоторые обозначения:  $N$  — размер графа (равен размеру муравьиной колонии),  $T$  — количество дней жизни колонии.

Далее приведён расчёт трудоемкости муравьиного алгоритма для решения задачи коммивояжера.

Трудоемкость внешней функции для муравьиного алгоритма:

$$f_{outer} = f_{ph} + f_{vis} + f_q + 2 + f_{loop-t} + 5. \quad (2.10)$$

Трудоёмкость функции инициализации матрицы феромона:

$$f_{ph} = 1 + 2 + N \cdot (2 + 3) + 2 + N \cdot (2 + 2 + N \cdot (2 + 3)) = 5 + 7 \cdot N + 5 \cdot N^2. \quad (2.11)$$

Трудоёмкость функции инициализации матрицы видимости:

$$f_{vis} = 1 + 2 + N \cdot (2 + 3) + 2 + N \cdot (2 + 2 + N \cdot (2 + 3)) = 5 + 7 \cdot N + 5 \cdot N^2. \quad (2.12)$$

Трудоёмкость функции инициализации коэффициента пропорциональности феромона:

$$f_q = 1 + 2 + N \cdot (2 + 2 + N \cdot (2 + 3)) + 1 = 4 + 4 \cdot N + 5 \cdot N^2. \quad (2.13)$$

Трудоёмкость цикла по дням:

$$f_{loop-t} = 2 + T \cdot (f_{col} + f_{loop-n} + f_{vap} + f_{inc} + f_{cor}). \quad (2.14)$$

Трудоёмкость функции создания колонии:

$$f_{col} = 1 + 2 + N \cdot (2 + 7) = 3 + 9 \cdot N. \quad (2.15)$$

Трудоёмкость функции испарения феромона:

$$f_{vap} = 2 + N \cdot (2 + 2 + N \cdot (2 + 7)) = 2 + 4 \cdot N + 9 \cdot N^2. \quad (2.16)$$

Трудоёмкость функции прироста феромона:

$$f_{inc} = 2 + N \cdot (2 + 2 + (N - 1) \cdot (2 + 9)) = 2 - 7 \cdot N + 11 \cdot N^2. \quad (2.17)$$

Трудоёмкость функции коррекции феромона в лучшем случае (не нужна коррекция) составит:

$$f_{cor} = 2 + N \cdot (2 + 2 + N \cdot 3) = 2 + 4 \cdot N + 3 \cdot N^2, \quad (2.18)$$

а в худшем (нужна коррекция всех):

$$f_{cor} = 2 + N \cdot (2 + 2 + N \cdot 6) = 2 + 4 \cdot N + 6 \cdot N^2. \quad (2.19)$$

Трудоёмкость цикла по количеству муравьёв в колонии:

$$f_{loop-n} = 2 + N \cdot (2 + f_{route} + 8). \quad (2.20)$$

Трудоёмкость функции поиска муравьём маршрута:

$$\begin{aligned} f_{route} &= 1 + 2 + N \cdot (2 + 2) + 1 + f_{go}^{N-1} + \dots + f_{go}^1 + f_{go}^1 + N \cdot (5 + f_{rm} + 3) + 3 = \\ &= 7 + 4 \cdot N + f_{go}^{N-1} + \dots + f_{go}^1 + f_{go}^1 + N \cdot (f_{rm} + 8). \end{aligned} \quad (2.21)$$

Трудоёмкость функции поиска муравьём следующей вершины для  $n$  непосещённых вершин в лучшем случае (для 20% элитных муравьёв):

$$f_{go} = 2 + 2 + (n - 1) \cdot (2 + 15) + 1 + 2 + 2 + (n - 1) \cdot (2 + 2 + 1) + 1 = -12 + 22 \cdot n, \quad (2.22)$$

а в худшем (для обычных):

$$f_{go} = 2 + 2 + (n - 1) \cdot (2 + 15) + 1 + 2 + 2 + (n - 1) \cdot (2 + 2 + 4) + 2 + 3 + 2 + 4 \cdot n + 2 = -7 + 29 \cdot n. \quad (2.23)$$

Трудоёмкость удаления элемента:

$$f_{rm} = 2 + N \cdot (2 + 9). \quad (2.24)$$

Далее ведётся расчёт для лучшего случая.

Подставляется 2.22 в 2.21:

$$f_{route} = 7 + 4 \cdot N + (N - 1) \cdot (-12) + (N - 1) \cdot 22 \cdot \frac{N - 1 + 1}{2} + 10 + N \cdot (f_{rm} + 8). \quad (2.25)$$

Подставляется 2.24 в 2.25:

$$f_{route} = 7 + 4 \cdot N + (N - 1) \cdot (-12) + (N - 1) \cdot 22 \cdot \frac{N - 1 + 1}{2} + 10 + N \cdot (11 \cdot N + 10), \quad (2.26)$$

что равняется

$$f_{route} = 5 - 9 \cdot N + 22 \cdot N^2. \quad (2.27)$$

Для худшего случая аналогично получается

$$f_{route} = 7 + 4 \cdot N + (N - 1) \cdot (-7) + (N - 1) \cdot 29 \cdot \frac{N - 1 + 1}{2} + 10 + N \cdot (11 \cdot N + 10), \quad (2.28)$$

что равняется

$$f_{route} = 24 - 27.5 \cdot N + 25.5 \cdot N^2. \quad (2.29)$$

Подставив полученное в 2.20 получим для лучшего случая:

$$f_{loop-n} = 2 + N \cdot (15 - 9 \cdot N + 22 \cdot N^2) = 2 + 15 \cdot N - 9 \cdot N^2 + 22 \cdot N^3, \quad (2.30)$$

а для худшего:

$$f_{loop-n} = 2 + N \cdot (34 - 27.5 \cdot N + 25.5 \cdot N^2) = 2 + 34 \cdot N - 27.5 \cdot N^2 + 25.5 \cdot N^3. \quad (2.31)$$

Далее выполним подстановку результатов в 2.14 и приведём подобные слагаемые. Для лучшего случая:

$$f_{loop-t} = 2 + 11 \cdot T + 25 \cdot N \cdot T + 14 \cdot N^2 \cdot T + 22 \cdot N^3 \cdot T, \quad (2.32)$$

а для худшего:

$$f_{loop-t} = 2 + 11 \cdot T + 44 \cdot N \cdot T - 1.5 \cdot N^2 \cdot T + 25.5 \cdot N^3 \cdot T. \quad (2.33)$$

Подставив полученное в формулу 2.10 для лучшего случая получается

$$f_{outer} = 23 + 18 \cdot N + 15 \cdot N^2 + 11 \cdot T + 25 \cdot N \cdot T + 14 \cdot N^2 \cdot T + 22 \cdot N^3 \cdot T, \quad (2.34)$$

а для худшего

$$f_{outer} = 23 + 18 \cdot N + 15 \cdot N^2 + 11 \cdot T + 44 \cdot N \cdot T - 1.5 \cdot N^2 \cdot T + 25.5 \cdot N^3 \cdot T. \quad (2.35)$$

Такую трудоёмкость можно оценить как  $O(N^3 \cdot T)$ .



## 3. Технологическая часть

В данном разделе будет представлена реализация алгоритмов решения задачи коммивояджёра. Также будут указаны обязательные требования к ПО, средства реализации алгоритмов и результаты проведённого тестирования программы.

### 3.1. Требования к ПО

Для программы выделен перечень требований:

- программой обрабатываются графы, записанные в файле указанной директории в виде симметричной матрицы смежности с предварительным указанием размерности;
- программой принимается режим запуска (обычный или для замеров), значения параметров  $\alpha$ ,  $\beta$ ,  $\rho$ ,  $time$ , описанные ранее;
- программой производится аварийное завершение с текстом об ошибке при ошибках;
- программой проводится модульное тестирование функций реализации решения задачи коммивояджёра;
- программой производятся замеры времени выполнения;
- программой на экран выводится результат работы — последовательность вершин кратчайшего маршрута и его длина.

### 3.2. Средства реализации

Для реализации данной работы был выбран язык программирования Go [1]. Выбор обусловлен наличием в *Go* библиотек для тестирования ПО и проведения замеров времени выполнения в том числе при помощи вставок на других языках программирования, а также необходимых для реализации поставленных цели и задач средств. В качестве среды разработки была выбрана *GoLand* [3].

### 3.3. Реализация алгоритмов

В листингах 3.1 – 3.5 представлены реализации функций для решения задачи коммивояджёра и необходимые для реализации муравьиного алгоритма подпрограммы: поиска муравьём маршрута и поиска муравьём следующей вершины маршрута.

Листинг 3.1 — Листинг функции реализации алгоритма полного перебора для решения задачи коммивояджёра

```
func TravellingSalesmanBF(g graph.Graph) ([]int, int) {
    var minRoute []int

    verts := make([]int, g.Size)
    for i := range verts {
        verts[i] = i
    }

    permuts := permutations(verts)
    minTax := math.MaxInt

    for i := range permuts {
        if g.IsOKRoute(permuts[i]) {
            cost := g.RouteTotalTax(permuts[i])

            if cost < minTax {
                minTax = cost
                minRoute = permuts[i]
            }
        }
    }

    return minRoute, minTax
}
```

Листинг 3.2 — Листинг функции реализации муравьиного алгоритма для решения задачи коммивояжера

```
func TravellingSalesmanACO(g graph.Graph, alpha, beta, k float64, time int)
([]int, int, error) {
    pheromone := createPheromoneMatrix(g.Size)
    vision := createVisionMatrix(g)
    minRoute := make([]int, g.Size)
    minTax := math.MaxInt
    q := getQ(g)

    for t := 0; t < time; t++ {
        c := createColony(g.Size)
        for _, v := range c.Members {
            err := v.FindRoute(g, pheromone, vision, alpha,
                beta)
            if err != nil {
                return nil, -1, fmt.Errorf("route error:
                    %w", err)
            }
            if v.Tax < minTax && (len(v.Route) == g.Size+1 ||
                g.Size == 1) {
                minTax = v.Tax
                minRoute = v.Route
            }
        }
        vaporize(pheromone, k)
        increase(pheromone, c, q)
        correction(pheromone)
    }
    if len(minRoute) == 0 {
        return nil, -1, errors.New("min route error")
    }
    return minRoute[:len(minRoute)-1], minTax, nil
}
```

Листинг 3.3 — Листинг функции реализации поиска муравьём маршрута

```
func (a *Ant) FindRoute(g graph.Graph, ph [][]float64, vis [][]float64,
alpha, beta float64) error {
    toGo := make([]int, 0)
    for i := range g.Connection {
        if i != a.Pos {
            toGo = append(toGo, i)
        }
    }
    cycle := false
    for len(toGo) != 0 {
        next, err := a.Go(toGo, ph, vis, alpha, beta)
        if err != nil {
            return err
        }
        a.Move(g.Connection[a.Pos][next], next)
        toGo = removeElement(toGo, next)
        if len(toGo) == 0 && !cycle {
            toGo = append(toGo, a.Route[0])
            cycle = true
        }
    }
    return nil
}
```

Листинг 3.4 — Листинг функции реализации поиска муравьём следующей вершины  
(начало)

```
func (a *Ant) Go(toGo []int, ph [][]float64, vis [][]float64, alpha, beta
float64) (int, error) {
    probs := make([]float64, 0)
    sumProbs := 0.0
    for _, v := range toGo {
        if vis[a.Pos][v] != -1 {
            greed := math.Pow(vis[a.Pos][v], beta)

```

Листинг 3.5 — Листинг функции реализации поиска муравьём следующей вершины  
(продолжение листинга ??code:go1)

```
        herd := math.Pow(ph[a.Pos][v], alpha)
        prob := greed * herd
        probs = append(probs, prob)
        sumProbs += prob
    }
}
if sumProbs <= 0 {
    return -1, errors.New("probability error")
}

maxProb := 0.0
greedChoice := 0
for i, v := range probs {
    v /= sumProbs
    if v > maxProb {
        maxProb = v
        greedChoice = toGo[i]
    }
}
if a.IsElite {
    return greedChoice, nil
}

choice := 0
curSum := 0.0
rand.Seed(time.Now().UnixNano())
randPoint := rand.Float64() * sumProbs
for curSum < randPoint {
    curSum += probs[choice]
    choice++
}
return toGo[choice-1], nil
}
```

### 3.4. Тестирование

В таблице 3.1 представлены тесты для решения задачи коммивояджёра с помощью алгоритма полного перебора и муравьиного алгоритма. Все тесты пройдены успешно. Тестирование проводилось на матрицах, заполненных случайными числами по методологии чёрного ящика. Для муравьиного алгоритма вводились следующие параметры:  $\alpha = 0.5$ ,  $\beta = 0.5$ ,  $\rho = 0.5$ ,  $time = 10$ .

Таблица 3.1 — Тесты для решения задачи коммивояджёра

№	Матрица смежности графа	Результат
1	$\begin{pmatrix} 0 \end{pmatrix}$	$\llbracket 0 \rrbracket$
2	$\begin{pmatrix} 0 & 34 \\ 34 & 0 \end{pmatrix}$	$[0 \ 1] \ 68$
3	$\begin{pmatrix} 0 & 24 & 6 \\ 24 & 0 & 13 \\ 6 & 13 & 0 \end{pmatrix}$	$[0 \ 2 \ 1] \ 43$
4	$\begin{pmatrix} 0 & 12 & 5 & 23 & 56 \\ 12 & 0 & 31 & 4 & 13 \\ 5 & 31 & 0 & 8 & 2 \\ 23 & 4 & 8 & 0 & 11 \\ 56 & 13 & 2 & 11 & 0 \end{pmatrix}$	$[3 \ 4 \ 2 \ 0 \ 1] \ 34$
5	$\begin{pmatrix} 0 & 45 & 12 & 67 & 88 & 22 & 14 & 4 \\ 45 & 0 & 5 & 7 & 89 & 34 & 121 & 7 \\ 12 & 5 & 0 & 23 & 45 & 32 & 43 & 12 \\ 67 & 7 & 23 & 0 & 44 & 44 & 32 & 2 \\ 88 & 89 & 45 & 44 & 0 & 4 & 56 & 21 \\ 22 & 34 & 32 & 44 & 4 & 0 & 22 & 47 \\ 14 & 121 & 43 & 32 & 56 & 22 & 0 & 9 \\ 4 & 7 & 12 & 2 & 21 & 47 & 9 & 0 \end{pmatrix}$	$[5 \ 4 \ 7 \ 3 \ 1 \ 2 \ 0 \ 6] \ 87$

## 4. Экспериментальная часть

В данном разделе описана параметризация реализации муравьиного алгоритма, проведённые замеры и представлены результаты исследования. Также будут уточнены характеристики устройства, на котором проводились замеры.

### 4.1. Технические характеристики

Технические характеристики устройства, на котором выполнялось тестирование [4]:

- операционная система *macOS Monterey* 12.4;
- 8 ГБ оперативной памяти;
- процессор *Apple M2* (базовая частота — 2400 МГц, но поддержка технологии *Turbo Boost* позволяет достигать частоты в 3500 МГц [5]).

### 4.2. Параметризация реализации муравьиного алгоритма

Целью проведения параметризации является определение таких сочетаний входных параметров, при которых алгоритм даёт наилучшие и стабильные результаты. В результате параметризации получена таблица со следующими столбцами:

- $\alpha$  — коэффициент стадности;
- *time* — количество дней жизни колонии;
- *ideal* — результат работы алгоритма полного перебора;
- *diff* — погрешность муравьиного алгоритма.

Параметризация проведена для трёх графов городов Африки, составленных по указанным выше принципам.

#### 4.2.1. Параметризация для графа №1

Матрица смежности для графа №1 имеет следующий вид:

$$\begin{pmatrix} 0 & 525 & 907 & 1639 \\ 525 & 0 & 606 & 965 \\ 907 & 606 & 0 & 357 \\ 1639 & 965 & 357 & 0 \end{pmatrix}. \quad (4.1)$$

В таблице 4.1 приведена усечённая выборка параметров, которые наилучшим образом решают поставленную задачу, что значит, что отсутствует погрешность относительно идеального результата.

Таблица 4.1 — Результаты параметризующего запуска для графа №1 из 4 вершин

$\alpha$	$\beta$	$\rho$	$time$	$ideal$	$diff$
0.1	0.9	0.1	1	2754	0
0.1	0.9	0.1	2	2754	0
0.1	0.9	0.1	3	2754	0
0.1	0.9	0.2	1	2754	0
0.1	0.9	0.2	2	2754	0
0.3	0.7	0.6	1	2754	0
0.3	0.7	0.6	2	2754	0
0.3	0.7	0.6	3	2754	0
0.4	0.6	0.1	1	2754	0
0.4	0.6	0.1	2	2754	0
0.4	0.6	0.1	3	2754	0
0.4	0.6	0.2	1	2754	0
0.4	0.6	0.2	2	2754	0
0.5	0.5	0.2	3	2754	0
0.5	0.5	0.3	1	2754	0
0.5	0.5	0.3	2	2754	0
0.5	0.5	0.3	3	2754	0
0.5	0.5	0.4	1	2754	0
0.5	0.5	0.4	2	2754	0
0.8	0.2	0.3	1	2754	0
0.8	0.2	0.3	2	2754	0



### 4.2.2. Параметризация для графа №2

Матрица смежности для графа №2 имеет следующий вид:

$$\begin{pmatrix} 0 & 525 & 907 & 1639 & 2346 & 5141 & 4123 & 945 \\ 525 & 0 & 606 & 965 & 1789 & 4728 & 3879 & 1370 \\ 907 & 606 & 0 & 357 & 1221 & 4095 & 3421 & 1649 \\ 1639 & 965 & 357 & 0 & 885 & 3757 & 3172 & 1874 \\ 2346 & 1789 & 1221 & 885 & 0 & 2631 & 2263 & 2221 \\ 5141 & 4728 & 4095 & 3757 & 2631 & 0 & 1223 & 4384 \\ 4123 & 3879 & 3421 & 3172 & 2263 & 1223 & 0 & 3253 \\ 945 & 1370 & 1649 & 1874 & 2221 & 4384 & 3253 & 0 \end{pmatrix}. \quad (4.2)$$

В таблице 4.2 приведена усечённая выборка параметров, которые наилучшим образом решают поставленную задачу, что значит, что отсутствует погрешность относительно идеального результата.

Таблица 4.2 — Результаты параметризующего запуска для графа №2 из 8 вершин

$\alpha$	$\beta$	$\rho$	$time$	$ideal$	$diff$
0.1	0.9	0.3	1	10 425	0
0.1	0.9	0.3	2	10 425	0
0.1	0.9	0.3	3	10 425	0
0.2	0.8	0.6	2	10 425	0
0.2	0.8	0.6	3	10 425	0
0.2	0.8	0.7	1	10 425	0
0.6	0.4	0.1	2	10 425	0
0.6	0.4	0.1	3	10 425	0
0.6	0.4	0.2	1	10 425	0
0.7	0.3	0.1	2	10 425	0
0.7	0.3	0.1	3	10 425	0
0.7	0.3	0.2	1	10 425	0
0.8	0.2	0.2	3	10 425	0
0.8	0.2	0.3	1	10 425	0
1.0	0.0	0.4	3	10 425	0
1.0	0.0	0.5	1	10 425	0
1.0	0.0	0.5	2	10 425	0

### 4.2.3. Параметризация для графа №3

Матрица смежности для графа №3 имеет следующий вид:

$$\begin{pmatrix} 0 & 525 & 907 & 1639 & 2346 & 5141 & 4123 & 945 & 1327 & 1970 \\ 525 & 0 & 606 & 965 & 1789 & 4728 & 3879 & 1370 & 899 & 1655 \\ 907 & 606 & 0 & 357 & 1221 & 4095 & 3421 & 1649 & 1237 & 1290 \\ 1639 & 965 & 357 & 0 & 885 & 3757 & 3172 & 1874 & 1318 & 1129 \\ 2346 & 1789 & 1221 & 885 & 0 & 2631 & 2263 & 2221 & 1465 & 1024 \\ 5141 & 4728 & 4095 & 3757 & 2631 & 0 & 1223 & 4384 & 3445 & 2588 \\ 4123 & 3879 & 3421 & 3172 & 2263 & 1223 & 0 & 3253 & 2408 & 1760 \\ 945 & 1370 & 1649 & 1874 & 2221 & 4384 & 3253 & 0 & 795 & 1597 \\ 1327 & 899 & 1237 & 1318 & 1465 & 3445 & 2408 & 795 & 0 & 1028 \\ 1970 & 1655 & 1290 & 1129 & 1024 & 2588 & 1760 & 1597 & 1028 & 0 \end{pmatrix}. \quad (4.3)$$

В таблице 4.3 приведена усечённая выборка параметров, которые наилучшим образом решают поставленную задачу, что значит, что отсутствует погрешность относительно идеального результата.

Таблица 4.3 — Результаты параметризующего запуска для графа №3 из 10 вершин

$\alpha$	$\beta$	$\rho$	$time$	$ideal$	$diff$
0.1	0.9	0.5	3	10 755	0
0.1	0.9	0.6	1	10 755	0
0.2	0.8	0.1	3	10 755	0
0.3	0.7	1.0	2	10 755	0
0.4	0.6	0.2	2	10 755	0
0.4	0.6	0.2	3	10 755	0
0.5	0.5	0.7	3	10 755	0
0.6	0.4	0.1	2	10 755	0
0.7	0.3	0.1	2	10 755	0
0.7	0.3	0.1	3	10 755	0
0.8	0.2	0.1	2	10 755	0
0.8	0.2	0.1	3	10 755	0
0.9	0.1	0.5	3	10 755	0
0.9	0.1	1.0	2	10 755	0
0.9	0.1	1.0	3	10 755	0

### 4.3. Измерение времени выполнения реализаций алгоритмов

Замеры времени реализаций алгоритмов производилось при помощи импортируемой в Go функции *clock\_gettime* языка C [15]. Эта функция при использовании макроса *CLOCK\_PROCESS\_CPUTIME\_ID* возвращает затраченное на работу процесса процессорное время в формате структуры *struct timespec*, в которой хранятся результаты замеров из двух частей — в секундах и наносекундах (см. листинг 4.6).

Листинг 4.6 — Листинг структуры *struct timespec*

```
struct timespec {  
    time_t    tv_sec;        /* seconds */  
    long      tv_nsec;       /* nanoseconds */  
};
```

Количество повторений замера для одинаковых входных данных — 100.

Функция, возвращающая текущее процессорное время, приведена в листинге 4.7.

Листинг 4.7 — Листинг функции, возвращающей текущее процессорное время

```
#include <pthread.h>  
#include <time.h>  
#include <stdio.h>  
  
static long long getCPUNs(){  
    struct timespec time;  
    if (clock_gettime(CLOCK_PROCESS_CPUTIME_ID, &time)) {  
        perror("can't measure time");  
        return 0;  
    }  
    return time.tv_sec * 1000000000LL + time.tv_nsec;  
}
```

Функция единичного замера времени выполнения в наносекундах приведена в листинге 4.8 (в примере функция замеряет время работы реализации алгоритма полного перебора).

Листинг 4.8 — Листинг функции единичного замера времени выполнения в наносекундах

```
func RunBenchmarkBF(g graph.Graph) int {  
    t := 0  
  
    start := C.getCPUNs()  
    for i := 0; i < NumTests; i++ {  
        bf.TravellingSalesmanBF(g)  
    }  
    end := C.getCPUNs()  
    t = int(end-start) / NumTests  
  
    return t  
}
```

Замеры проводились на графах, вершины которых представляют города Африки, а веса рёбер — расстояния между соответствующими городами. Вес ребра будет больше, если прямой путь между городами лежит через пустыню, и меньше, если путь проходит через водоём (кроме рек).

В таблице 4.4 представлены результаты замеров времени работы реализованных алгоритмов (в мкс.) при разном количестве вершин во входном графе, а на рисунке 4.1 приведен график, отражающий зависимость времени выполнения реализаций решения задачи коммивояжёра от количества вершин во входном графе. В легендах графика и в таблице обозначение *ACO* значит муравьиный алгоритм, а *BF* — алгоритм полного перебора.

Таблица 4.4 — Результаты замеров времени работы конвейера при разном количестве входных заявок (мкс.)

Кол-во вершин графа	ACO	BF
1	217	0
2	912	0
3	1780	0
4	2987	1
5	4371	19
6	6114	51
7	8068	377
8	10 531	6732
9	13 119	57 514
10	15 990	622 262

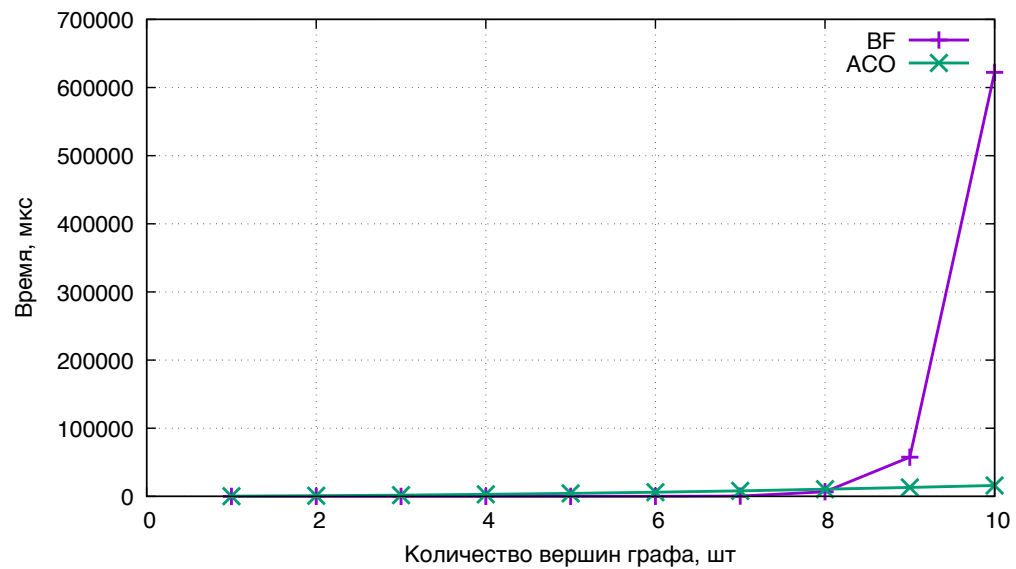


Рисунок 4.1 — Зависимость времени выполнения реализаций решения задачи коммивояжёра от количества вершин во входном графе

# Заключение

Из проведённых замеров времени работы реализаций решения задачи коммивояжёра с помощью алгоритма полного перебора и муравьиного алгоритма можно сделать следующие выводы. Муравьиный алгоритм работает дольше при малых количествах вершин в графе. Например, на количестве вершин, равном 5, муравьиный алгоритм выполняется в 230 раз медленнее алгоритма полного перебора. При больших значениях количества вершин в графе муравьиный алгоритм по временной эффективности начинает значительно превосходить алгоритм полного перебора. Например, на количестве вершин, равном 10, муравьиный алгоритм работает в 38 раз быстрее алгоритма полного перебора.

В ходе выполнения лабораторной работы была достигнута поставленная цель: были получены навыки программирования, тестирования полученного программного продукта и проведения замеров времени выполнения по результатам работы программы на примере реализации решения задачи коммивояжёра при помощи алгоритма полного перебора и муравьиного алгоритма.

В процессе выполнения лабораторной работы были также реализованы все поставленные задачи, а именно:

- были изучены теоретические основы задачи коммивояжёра, алгоритма полного перебора и муравьиного алгоритма;
- были описаны алгоритм полного перебора и муравьиный алгоритм;
- была выполнена программная реализация данных алгоритмов;
- была выполнена оценка трудоёмкости реализаций алгоритмов;
- была проведена параметризация для муравьиного алгоритма;
- была выполнена программная реализация параллельных конвейерных вычислений с не менее чем тремя линиями;
- были проведены замеры времени работы (в мкс) для данных алгоритмов;
- была получена зависимость замеряемой величины от количества вершин в графе, предоставляемом на вход алгоритму;
- был проведен сравнительный анализ двух представленных реализаций решения задачи коммивояжёра на основе полученной зависимости.

# Список использованных источников

1. Документация по языку программирования *Go* [Электронный ресурс]. Режим доступа: <https://go.dev/doc/> (дата обращения: 20.09.2022).
2. Документация по пакетам языка программирования *Go* [Электронный ресурс]. Режим доступа: <https://pkg.go.dev> (дата обращения: 20.09.2022).
3. GoLand: IDE для профессиональной разработки на *Go* [Электронный ресурс]. Режим доступа: <https://www.jetbrains.com/ru-ru/go/> (дата обращения: 20.09.2022).
4. Техническая спецификация ноутбука *MacBook Air* [Электронный ресурс]. Режим доступа: <https://support.apple.com/kb/SP869> (дата обращения: 20.09.2022).
5. *Apple M2* [Электронный ресурс]. Режим доступа: <https://www.notebookcheck.net/Apple-M2-Processor-Benchmarks-and-Specs.632312.0.html> (дата обращения: 10.10.2022).
6. Donovan A. A. A., Kernighan B. W. The Go programming language. – Addison-Wesley Professional, 2015.
7. *gnuplot homepage* [Электронный ресурс]. Режим доступа: <http://www.gnuplot.info> (дата обращения: 10.10.2022).
8. Исходный код *src/testing/benchmark.go* [Электронный ресурс]. Режим доступа: <https://go.dev/src/testing/benchmark.go> (дата обращения: 10.10.2022).
9. Lawler E. L., Lensira J. K., Rinnooy Kan A. H. G., Shmoys D. B. ed. The traveling salesman problem. A guided tour of combinatorial optimization. N. Y.: J. Wiley & Sons, 1985.
10. Меламед И. И., Сергеев С. И., Сигал И. Х. Задача коммивояжера. Вопросы теории // Автоматика и телемеханика. – 1989. – №. 9. – С. 3-33.
11. Ульянов М. В. Ресурсно-эффективные компьютерные алгоритмы. Разработка и анализ // М.: ФИЗМАТЛИТ. – 2008. – Т. 304.
12. Пархоменко П. П. Классификация гамильтоновых циклов в двоичных гиперкубах // Автоматика и телемеханика. – 2001. – №. 6. – С. 136-150.

13. Яндекс.Карты [Электронный ресурс]. Режим доступа: <https://yandex.ru/maps> (дата обращения: 10.12.2022).
14. Штовба С. Д. Муравьиные алгоритмы // Exponenta Pro. Математика в приложениях. – 2003. – Т. 4. – №. 4. – С. 70-75.
15. `clock_gettime(3)` — Linux manual page [Электронный ресурс]. Режим доступа: [https://man7.org/linux/man-pages/man3/clock\\_gettime.3.html](https://man7.org/linux/man-pages/man3/clock_gettime.3.html) (дата обращения: 10.10.2022).



# Приложение А

## Параметризация для графа №1

В таблице 4.1 представлены полные результаты параметризующего запуска для графа №1 из 4 вершин.

Таблица 5.5 — Результаты параметризующего запуска для графа №1 из 4 вершин

$\alpha$	$\beta$	$\rho$	$time$	$ideal$	$diff$
0.1	0.9	0.1	2	2754	0
0.1	0.9	0.1	3	2754	0
0.1	0.9	0.2	1	2754	0
0.1	0.9	0.2	2	2754	0
0.1	0.9	0.2	3	2754	0
0.1	0.9	0.3	1	2754	0
0.1	0.9	0.3	2	2754	0
0.1	0.9	0.3	3	2754	0
0.1	0.9	0.4	1	2754	0
0.1	0.9	0.4	2	2754	0
0.1	0.9	0.4	3	2754	0
0.1	0.9	0.5	1	2754	373
0.1	0.9	0.5	2	2754	0
0.1	0.9	0.5	3	2754	0
0.1	0.9	0.6	1	2754	0
0.1	0.9	0.6	2	2754	0
0.1	0.9	0.6	3	2754	0
0.1	0.9	0.7	1	2754	0
0.1	0.9	0.7	2	2754	0
0.1	0.9	0.7	3	2754	0
0.1	0.9	0.8	1	2754	0
0.1	0.9	0.8	2	2754	0
0.1	0.9	0.8	3	2754	0

Продолжение таблицы 5.5

0.1	0.9	0.9	1	2754	0
0.1	0.9	0.9	2	2754	0
0.1	0.9	0.9	3	2754	0
0.1	0.9	1.0	1	2754	0
0.1	0.9	1.0	2	2754	0
0.1	0.9	1.0	3	2754	0
0.2	0.8	0.1	1	2754	0
0.2	0.8	0.1	2	2754	0
0.2	0.8	0.1	3	2754	0
0.2	0.8	0.2	1	2754	0
0.2	0.8	0.2	2	2754	0
0.2	0.8	0.2	3	2754	0
0.2	0.8	0.3	1	2754	0
0.2	0.8	0.3	2	2754	0
0.2	0.8	0.3	3	2754	0
0.2	0.8	0.4	1	2754	373
0.2	0.8	0.4	2	2754	0
0.2	0.8	0.4	3	2754	0
0.2	0.8	0.5	1	2754	373
0.2	0.8	0.5	2	2754	0
0.2	0.8	0.5	3	2754	0
0.2	0.8	0.6	1	2754	0
0.2	0.8	0.6	2	2754	0
0.2	0.8	0.6	3	2754	0
0.2	0.8	0.7	1	2754	0
0.2	0.8	0.7	2	2754	0
0.2	0.8	0.7	3	2754	0
0.2	0.8	0.8	1	2754	373
0.2	0.8	0.8	2	2754	0
0.2	0.8	0.8	3	2754	0
0.2	0.8	0.9	1	2754	0
0.2	0.8	0.9	2	2754	0
0.2	0.8	0.9	3	2754	0

Продолжение таблицы 5.5

0.2	0.8	1.0	1	2754	0
0.2	0.8	1.0	2	2754	0
0.2	0.8	1.0	3	2754	0
0.3	0.7	0.1	1	2754	0
0.3	0.7	0.1	2	2754	0
0.3	0.7	0.1	3	2754	0
0.3	0.7	0.2	1	2754	373
0.3	0.7	0.2	2	2754	0
0.3	0.7	0.2	3	2754	0
0.3	0.7	0.3	1	2754	0
0.3	0.7	0.3	2	2754	0
0.3	0.7	0.3	3	2754	0
0.3	0.7	0.4	1	2754	0
0.3	0.7	0.4	2	2754	0
0.3	0.7	0.4	3	2754	0
0.3	0.7	0.5	1	2754	0
0.3	0.7	0.5	2	2754	0
0.3	0.7	0.5	3	2754	0
0.3	0.7	0.6	1	2754	0
0.3	0.7	0.6	2	2754	0
0.3	0.7	0.6	3	2754	0
0.3	0.7	0.7	1	2754	373
0.3	0.7	0.7	2	2754	0
0.3	0.7	0.7	3	2754	0
0.3	0.7	0.8	1	2754	0
0.3	0.7	0.8	2	2754	0
0.3	0.7	0.8	3	2754	0
0.3	0.7	0.9	1	2754	0
0.3	0.7	0.9	2	2754	0
0.3	0.7	0.9	3	2754	0
0.3	0.7	1.0	1	2754	0
0.3	0.7	1.0	2	2754	0
0.3	0.7	1.0	3	2754	0

Продолжение таблицы 5.5

0.4	0.6	0.1	1	2754	0
0.4	0.6	0.1	2	2754	0
0.4	0.6	0.1	3	2754	0
0.4	0.6	0.2	1	2754	0
0.4	0.6	0.2	2	2754	0
0.4	0.6	0.2	3	2754	0
0.4	0.6	0.3	1	2754	0
0.4	0.6	0.3	2	2754	0
0.4	0.6	0.3	3	2754	0
0.4	0.6	0.4	1	2754	0
0.4	0.6	0.4	2	2754	0
0.4	0.6	0.4	3	2754	0
0.4	0.6	0.5	1	2754	0
0.4	0.6	0.5	2	2754	0
0.4	0.6	0.5	3	2754	0
0.4	0.6	0.6	1	2754	0
0.4	0.6	0.6	2	2754	0
0.4	0.6	0.6	3	2754	373
0.4	0.6	0.7	1	2754	373
0.4	0.6	0.7	2	2754	0
0.4	0.6	0.7	3	2754	0
0.4	0.6	0.8	1	2754	0
0.4	0.6	0.8	2	2754	0
0.4	0.6	0.8	3	2754	0
0.4	0.6	0.9	1	2754	0
0.4	0.6	0.9	2	2754	0
0.4	0.6	0.9	3	2754	0
0.4	0.6	1.0	1	2754	0
0.4	0.6	1.0	2	2754	0
0.4	0.6	1.0	3	2754	373
0.5	0.5	0.1	1	2754	0
0.5	0.5	0.1	2	2754	0
0.5	0.5	0.1	3	2754	0

Продолжение таблицы 5.5

0.5	0.5	0.2	1	2754	373
0.5	0.5	0.2	2	2754	0
0.5	0.5	0.2	3	2754	0
0.5	0.5	0.3	1	2754	0
0.5	0.5	0.3	2	2754	0
0.5	0.5	0.3	3	2754	0
0.5	0.5	0.4	1	2754	0
0.5	0.5	0.4	2	2754	0
0.5	0.5	0.4	3	2754	0
0.5	0.5	0.5	1	2754	373
0.5	0.5	0.5	2	2754	0
0.5	0.5	0.5	3	2754	0
0.5	0.5	0.6	1	2754	0
0.5	0.5	0.6	2	2754	0
0.5	0.5	0.6	3	2754	0
0.5	0.5	0.7	1	2754	373
0.5	0.5	0.7	2	2754	0
0.5	0.5	0.7	3	2754	0
0.5	0.5	0.8	1	2754	373
0.5	0.5	0.8	2	2754	0
0.5	0.5	0.8	3	2754	0
0.5	0.5	0.9	1	2754	0
0.5	0.5	0.9	2	2754	0
0.5	0.5	0.9	3	2754	0
0.5	0.5	1.0	1	2754	0
0.5	0.5	1.0	2	2754	0
0.5	0.5	1.0	3	2754	0
0.6	0.4	0.1	1	2754	0
0.6	0.4	0.1	2	2754	0
0.6	0.4	0.1	3	2754	0
0.6	0.4	0.2	1	2754	0
0.6	0.4	0.2	2	2754	0
0.6	0.4	0.2	3	2754	0

Продолжение таблицы 5.5

0.6	0.4	0.3	1	2754	0
0.6	0.4	0.3	2	2754	0
0.6	0.4	0.3	3	2754	0
0.6	0.4	0.4	1	2754	0
0.6	0.4	0.4	2	2754	0
0.6	0.4	0.4	3	2754	0
0.6	0.4	0.5	1	2754	0
0.6	0.4	0.5	2	2754	0
0.6	0.4	0.5	3	2754	0
0.6	0.4	0.6	1	2754	0
0.6	0.4	0.6	2	2754	0
0.6	0.4	0.6	3	2754	0
0.6	0.4	0.7	1	2754	0
0.6	0.4	0.7	2	2754	0
0.6	0.4	0.7	3	2754	0
0.6	0.4	0.8	1	2754	0
0.6	0.4	0.8	2	2754	0
0.6	0.4	0.8	3	2754	0
0.6	0.4	0.9	1	2754	373
0.6	0.4	0.9	2	2754	0
0.6	0.4	0.9	3	2754	0
0.6	0.4	1.0	1	2754	0
0.6	0.4	1.0	2	2754	0
0.6	0.4	1.0	3	2754	0
0.7	0.3	0.1	1	2754	0
0.7	0.3	0.1	2	2754	0
0.7	0.3	0.1	3	2754	0
0.7	0.3	0.2	1	2754	0
0.7	0.3	0.2	2	2754	0
0.7	0.3	0.2	3	2754	0
0.7	0.3	0.3	1	2754	0
0.7	0.3	0.3	2	2754	0
0.7	0.3	0.3	3	2754	0

Продолжение таблицы 5.5

0.7	0.3	0.4	1	2754	373
0.7	0.3	0.4	2	2754	0
0.7	0.3	0.4	3	2754	0
0.7	0.3	0.5	1	2754	0
0.7	0.3	0.5	2	2754	0
0.7	0.3	0.5	3	2754	0
0.7	0.3	0.6	1	2754	0
0.7	0.3	0.6	2	2754	0
0.7	0.3	0.6	3	2754	0
0.7	0.3	0.7	1	2754	0
0.7	0.3	0.7	2	2754	0
0.7	0.3	0.7	3	2754	0
0.7	0.3	0.8	1	2754	0
0.7	0.3	0.8	2	2754	0
0.7	0.3	0.8	3	2754	0
0.7	0.3	0.9	1	2754	0
0.7	0.3	0.9	2	2754	0
0.7	0.3	0.9	3	2754	0
0.7	0.3	1.0	1	2754	0
0.7	0.3	1.0	2	2754	0
0.7	0.3	1.0	3	2754	0
0.8	0.2	0.1	1	2754	0
0.8	0.2	0.1	2	2754	373
0.8	0.2	0.1	3	2754	0
0.8	0.2	0.2	1	2754	373
0.8	0.2	0.2	2	2754	0
0.8	0.2	0.2	3	2754	0
0.8	0.2	0.3	1	2754	373
0.8	0.2	0.3	2	2754	0
0.8	0.2	0.3	3	2754	0
0.8	0.2	0.4	1	2754	0
0.8	0.2	0.4	2	2754	0
0.8	0.2	0.4	3	2754	0

Продолжение таблицы 5.5

0.8	0.2	0.5	1	2754	0
0.8	0.2	0.5	2	2754	0
0.8	0.2	0.5	3	2754	0
0.8	0.2	0.6	1	2754	373
0.8	0.2	0.6	2	2754	0
0.8	0.2	0.6	3	2754	0
0.8	0.2	0.7	1	2754	0
0.8	0.2	0.7	2	2754	0
0.8	0.2	0.7	3	2754	0
0.8	0.2	0.8	1	2754	373
0.8	0.2	0.8	2	2754	0
0.8	0.2	0.8	3	2754	0
0.8	0.2	0.9	1	2754	0
0.8	0.2	0.9	2	2754	0
0.8	0.2	0.9	3	2754	0
0.8	0.2	1.0	1	2754	0
0.8	0.2	1.0	2	2754	0
0.8	0.2	1.0	3	2754	0
0.9	0.1	0.1	1	2754	0
0.9	0.1	0.1	2	2754	0
0.9	0.1	0.1	3	2754	0
0.9	0.1	0.2	1	2754	0
0.9	0.1	0.2	2	2754	0
0.9	0.1	0.2	3	2754	0
0.9	0.1	0.3	1	2754	0
0.9	0.1	0.3	2	2754	0
0.9	0.1	0.3	3	2754	0
0.9	0.1	0.4	1	2754	0
0.9	0.1	0.4	2	2754	0
0.9	0.1	0.4	3	2754	0
0.9	0.1	0.5	1	2754	0
0.9	0.1	0.5	2	2754	0
0.9	0.1	0.5	3	2754	0



Продолжение таблицы 5.5

0.9	0.1	0.6	1	2754	373
0.9	0.1	0.6	2	2754	0
0.9	0.1	0.6	3	2754	0
0.9	0.1	0.7	1	2754	0
0.9	0.1	0.7	2	2754	0
0.9	0.1	0.7	3	2754	0
0.9	0.1	0.8	1	2754	0
0.9	0.1	0.8	2	2754	0
0.9	0.1	0.8	3	2754	0
0.9	0.1	0.9	1	2754	0
0.9	0.1	0.9	2	2754	0
0.9	0.1	0.9	3	2754	0
0.9	0.1	1.0	1	2754	0
0.9	0.1	1.0	2	2754	0
0.9	0.1	1.0	3	2754	0
1.0	0.0	0.1	1	2754	0
1.0	0.0	0.1	2	2754	0
1.0	0.0	0.1	3	2754	0
1.0	0.0	0.2	1	2754	373
1.0	0.0	0.2	2	2754	0
1.0	0.0	0.2	3	2754	0
1.0	0.0	0.3	1	2754	0
1.0	0.0	0.3	2	2754	0
1.0	0.0	0.3	3	2754	0
1.0	0.0	0.4	1	2754	0
1.0	0.0	0.4	2	2754	0
1.0	0.0	0.4	3	2754	0
1.0	0.0	0.5	1	2754	0
1.0	0.0	0.5	2	2754	0
1.0	0.0	0.5	3	2754	0
1.0	0.0	0.6	1	2754	0
1.0	0.0	0.6	2	2754	0
1.0	0.0	0.6	3	2754	0

Продолжение таблицы 5.5

1.0	0.0	0.7	1	2754	0
1.0	0.0	0.7	2	2754	0
1.0	0.0	0.7	3	2754	0
1.0	0.0	0.8	1	2754	0
1.0	0.0	0.8	2	2754	0
1.0	0.0	0.8	3	2754	0
1.0	0.0	0.9	1	2754	0
1.0	0.0	0.9	2	2754	0
1.0	0.0	0.9	3	2754	0
1.0	0.0	1.0	1	2754	0
1.0	0.0	1.0	2	2754	0
1.0	0.0	1.0	3	2754	0

## Параметризация для графа №2

Таблица 5.6 — Результаты параметризующего запуска для графа №2 из 8 вершин

$\alpha$	$\beta$	$\rho$	$time$	$ideal$	$diff$
0.1	0.9	0.1	2	10425	0
0.1	0.9	0.1	3	10425	0
0.1	0.9	0.2	1	10425	2647
0.1	0.9	0.2	2	10425	0
0.1	0.9	0.2	3	10425	0
0.1	0.9	0.3	1	10425	3006
0.1	0.9	0.3	2	10425	0
0.1	0.9	0.3	3	10425	0
0.1	0.9	0.4	1	10425	0
0.1	0.9	0.4	2	10425	0
0.1	0.9	0.4	3	10425	0
0.1	0.9	0.5	1	10425	1772
0.1	0.9	0.5	2	10425	1863
0.1	0.9	0.5	3	10425	0
0.1	0.9	0.6	1	10425	0
0.1	0.9	0.6	2	10425	0
0.1	0.9	0.6	3	10425	0
0.1	0.9	0.7	1	10425	0
0.1	0.9	0.7	2	10425	2557
0.1	0.9	0.7	3	10425	0
0.1	0.9	0.8	1	10425	763
0.1	0.9	0.8	2	10425	0
0.1	0.9	0.8	3	10425	0
0.1	0.9	0.9	1	10425	3344
0.1	0.9	0.9	2	10425	1472
0.1	0.9	0.9	3	10425	0
0.1	0.9	1.0	1	10425	2899
0.1	0.9	1.0	2	10425	0
0.1	0.9	1.0	3	10425	2186
0.2	0.8	0.1	1	10425	5109

Продолжение таблицы 5.6

0.2	0.8	0.1	2	10425	1863
0.2	0.8	0.1	3	10425	0
0.2	0.8	0.2	1	10425	3268
0.2	0.8	0.2	2	10425	763
0.2	0.8	0.2	3	10425	0
0.2	0.8	0.3	1	10425	3674
0.2	0.8	0.3	2	10425	0
0.2	0.8	0.3	3	10425	0
0.2	0.8	0.4	1	10425	0
0.2	0.8	0.4	2	10425	1772
0.2	0.8	0.4	3	10425	0
0.2	0.8	0.5	1	10425	763
0.2	0.8	0.5	2	10425	0
0.2	0.8	0.5	3	10425	0
0.2	0.8	0.6	1	10425	3701
0.2	0.8	0.6	2	10425	0
0.2	0.8	0.6	3	10425	0
0.2	0.8	0.7	1	10425	0
0.2	0.8	0.7	2	10425	0
0.2	0.8	0.7	3	10425	0
0.2	0.8	0.8	1	10425	0
0.2	0.8	0.8	2	10425	763
0.2	0.8	0.8	3	10425	0
0.2	0.8	0.9	1	10425	3570
0.2	0.8	0.9	2	10425	2230
0.2	0.8	0.9	3	10425	763
0.2	0.8	1.0	1	10425	1472
0.2	0.8	1.0	2	10425	763
0.2	0.8	1.0	3	10425	0
0.3	0.7	0.1	1	10425	763
0.3	0.7	0.1	2	10425	1472
0.3	0.7	0.1	3	10425	0
0.3	0.7	0.2	1	10425	2647

Продолжение таблицы 5.6

0.3	0.7	0.2	2	10425	2962
0.3	0.7	0.2	3	10425	0
0.3	0.7	0.3	1	10425	0
0.3	0.7	0.3	2	10425	0
0.3	0.7	0.3	3	10425	0
0.3	0.7	0.4	1	10425	1472
0.3	0.7	0.4	2	10425	3786
0.3	0.7	0.4	3	10425	1472
0.3	0.7	0.5	1	10425	763
0.3	0.7	0.5	2	10425	2988
0.3	0.7	0.5	3	10425	763
0.3	0.7	0.6	1	10425	0
0.3	0.7	0.6	2	10425	0
0.3	0.7	0.6	3	10425	0
0.3	0.7	0.7	1	10425	2125
0.3	0.7	0.7	2	10425	763
0.3	0.7	0.7	3	10425	0
0.3	0.7	0.8	1	10425	0
0.3	0.7	0.8	2	10425	763
0.3	0.7	0.8	3	10425	763
0.3	0.7	0.9	1	10425	0
0.3	0.7	0.9	2	10425	3884
0.3	0.7	0.9	3	10425	763
0.3	0.7	1.0	1	10425	763
0.3	0.7	1.0	2	10425	0
0.3	0.7	1.0	3	10425	763
0.4	0.6	0.1	1	10425	2647
0.4	0.6	0.1	2	10425	0
0.4	0.6	0.1	3	10425	0
0.4	0.6	0.2	1	10425	0
0.4	0.6	0.2	2	10425	763
0.4	0.6	0.2	3	10425	0
0.4	0.6	0.3	1	10425	0

Продолжение таблицы 5.6

0.4	0.6	0.3	2	10425	0
0.4	0.6	0.3	3	10425	0
0.4	0.6	0.4	1	10425	5170
0.4	0.6	0.4	2	10425	0
0.4	0.6	0.4	3	10425	0
0.4	0.6	0.5	1	10425	3786
0.4	0.6	0.5	2	10425	0
0.4	0.6	0.5	3	10425	0
0.4	0.6	0.6	1	10425	0
0.4	0.6	0.6	2	10425	0
0.4	0.6	0.6	3	10425	2338
0.4	0.6	0.7	1	10425	0
0.4	0.6	0.7	2	10425	763
0.4	0.6	0.7	3	10425	0
0.4	0.6	0.8	1	10425	763
0.4	0.6	0.8	2	10425	0
0.4	0.6	0.8	3	10425	0
0.4	0.6	0.9	1	10425	3786
0.4	0.6	0.9	2	10425	763
0.4	0.6	0.9	3	10425	1472
0.4	0.6	1.0	1	10425	1472
0.4	0.6	1.0	2	10425	1472
0.4	0.6	1.0	3	10425	0
0.5	0.5	0.1	1	10425	0
0.5	0.5	0.1	2	10425	0
0.5	0.5	0.1	3	10425	0
0.5	0.5	0.2	1	10425	763
0.5	0.5	0.2	2	10425	0
0.5	0.5	0.2	3	10425	0
0.5	0.5	0.3	1	10425	0
0.5	0.5	0.3	2	10425	1472
0.5	0.5	0.3	3	10425	0
0.5	0.5	0.4	1	10425	763

Продолжение таблицы 5.6

0.5	0.5	0.4	2	10425	2408
0.5	0.5	0.4	3	10425	0
0.5	0.5	0.5	1	10425	763
0.5	0.5	0.5	2	10425	1472
0.5	0.5	0.5	3	10425	0
0.5	0.5	0.6	1	10425	763
0.5	0.5	0.6	2	10425	0
0.5	0.5	0.6	3	10425	0
0.5	0.5	0.7	1	10425	2646
0.5	0.5	0.7	2	10425	763
0.5	0.5	0.7	3	10425	0
0.5	0.5	0.8	1	10425	0
0.5	0.5	0.8	2	10425	0
0.5	0.5	0.8	3	10425	0
0.5	0.5	0.9	1	10425	0
0.5	0.5	0.9	2	10425	0
0.5	0.5	0.9	3	10425	0
0.5	0.5	1.0	1	10425	1472
0.5	0.5	1.0	2	10425	763
0.5	0.5	1.0	3	10425	0
0.6	0.4	0.1	1	10425	763
0.6	0.4	0.1	2	10425	0
0.6	0.4	0.1	3	10425	0
0.6	0.4	0.2	1	10425	1472
0.6	0.4	0.2	2	10425	3674
0.6	0.4	0.2	3	10425	0
0.6	0.4	0.3	1	10425	5448
0.6	0.4	0.3	2	10425	0
0.6	0.4	0.3	3	10425	0
0.6	0.4	0.4	1	10425	5906
0.6	0.4	0.4	2	10425	3674
0.6	0.4	0.4	3	10425	0
0.6	0.4	0.5	1	10425	0

Продолжение таблицы 5.6

0.6	0.4	0.5	2	10425	763
0.6	0.4	0.5	3	10425	763
0.6	0.4	0.6	1	10425	0
0.6	0.4	0.6	2	10425	0
0.6	0.4	0.6	3	10425	0
0.6	0.4	0.7	1	10425	0
0.6	0.4	0.7	2	10425	0
0.6	0.4	0.7	3	10425	0
0.6	0.4	0.8	1	10425	0
0.6	0.4	0.8	2	10425	763
0.6	0.4	0.8	3	10425	0
0.6	0.4	0.9	1	10425	5349
0.6	0.4	0.9	2	10425	0
0.6	0.4	0.9	3	10425	0
0.6	0.4	1.0	1	10425	3786
0.6	0.4	1.0	2	10425	0
0.6	0.4	1.0	3	10425	0
0.7	0.3	0.1	1	10425	0
0.7	0.3	0.1	2	10425	0
0.7	0.3	0.1	3	10425	763
0.7	0.3	0.2	1	10425	1472
0.7	0.3	0.2	2	10425	0
0.7	0.3	0.2	3	10425	0
0.7	0.3	0.3	1	10425	1472
0.7	0.3	0.3	2	10425	0
0.7	0.3	0.3	3	10425	0
0.7	0.3	0.4	1	10425	1472
0.7	0.3	0.4	2	10425	0
0.7	0.3	0.4	3	10425	0
0.7	0.3	0.5	1	10425	8845
0.7	0.3	0.5	2	10425	0
0.7	0.3	0.5	3	10425	0
0.7	0.3	0.6	1	10425	0



Продолжение таблицы 5.6

0.7	0.3	0.6	2	10425	763
0.7	0.3	0.6	3	10425	0
0.7	0.3	0.7	1	10425	7187
0.7	0.3	0.7	2	10425	763
0.7	0.3	0.7	3	10425	2647
0.7	0.3	0.8	1	10425	0
0.7	0.3	0.8	2	10425	0
0.7	0.3	0.8	3	10425	0
0.7	0.3	0.9	1	10425	0
0.7	0.3	0.9	2	10425	763
0.7	0.3	0.9	3	10425	0
0.7	0.3	1.0	1	10425	0
0.7	0.3	1.0	2	10425	0
0.7	0.3	1.0	3	10425	763
0.8	0.2	0.1	1	10425	0
0.8	0.2	0.1	2	10425	0
0.8	0.2	0.1	3	10425	0
0.8	0.2	0.2	1	10425	763
0.8	0.2	0.2	2	10425	0
0.8	0.2	0.2	3	10425	0
0.8	0.2	0.3	1	10425	763
0.8	0.2	0.3	2	10425	1472
0.8	0.2	0.3	3	10425	0
0.8	0.2	0.4	1	10425	0
0.8	0.2	0.4	2	10425	763
0.8	0.2	0.4	3	10425	0
0.8	0.2	0.5	1	10425	763
0.8	0.2	0.5	2	10425	0
0.8	0.2	0.5	3	10425	1472
0.8	0.2	0.6	1	10425	763
0.8	0.2	0.6	2	10425	0
0.8	0.2	0.6	3	10425	1472
0.8	0.2	0.7	1	10425	763

Продолжение таблицы 5.6

0.8	0.2	0.7	2	10425	0
0.8	0.2	0.7	3	10425	0
0.8	0.2	0.8	1	10425	9550
0.8	0.2	0.8	2	10425	0
0.8	0.2	0.8	3	10425	0
0.8	0.2	0.9	1	10425	3674
0.8	0.2	0.9	2	10425	0
0.8	0.2	0.9	3	10425	0
0.8	0.2	1.0	1	10425	6383
0.8	0.2	1.0	2	10425	0
0.8	0.2	1.0	3	10425	1472
0.9	0.1	0.1	1	10425	3786
0.9	0.1	0.1	2	10425	763
0.9	0.1	0.1	3	10425	763
0.9	0.1	0.2	1	10425	0
0.9	0.1	0.2	2	10425	763
0.9	0.1	0.2	3	10425	0
0.9	0.1	0.3	1	10425	4209
0.9	0.1	0.3	2	10425	0
0.9	0.1	0.3	3	10425	0
0.9	0.1	0.4	1	10425	1472
0.9	0.1	0.4	2	10425	763
0.9	0.1	0.4	3	10425	763
0.9	0.1	0.5	1	10425	3674
0.9	0.1	0.5	2	10425	0
0.9	0.1	0.5	3	10425	0
0.9	0.1	0.6	1	10425	0
0.9	0.1	0.6	2	10425	1472
0.9	0.1	0.6	3	10425	0
0.9	0.1	0.7	1	10425	0
0.9	0.1	0.7	2	10425	0
0.9	0.1	0.7	3	10425	0
0.9	0.1	0.8	1	10425	0

Продолжение таблицы 5.6

0.9	0.1	0.8	2	10425	763
0.9	0.1	0.8	3	10425	0
0.9	0.1	0.9	1	10425	763
0.9	0.1	0.9	2	10425	763
0.9	0.1	0.9	3	10425	1472
0.9	0.1	1.0	1	10425	763
0.9	0.1	1.0	2	10425	0
0.9	0.1	1.0	3	10425	763
1.0	0.0	0.1	1	10425	0
1.0	0.0	0.1	2	10425	0
1.0	0.0	0.1	3	10425	0
1.0	0.0	0.2	1	10425	0
1.0	0.0	0.2	2	10425	0
1.0	0.0	0.2	3	10425	2340
1.0	0.0	0.3	1	10425	2411
1.0	0.0	0.3	2	10425	2547
1.0	0.0	0.3	3	10425	726
1.0	0.0	0.4	1	10425	726
1.0	0.0	0.4	2	10425	1613
1.0	0.0	0.4	3	10425	2498
1.0	0.0	0.5	1	10425	2547
1.0	0.0	0.5	2	10425	2547
1.0	0.0	0.5	3	10425	0
1.0	0.0	0.6	1	10425	0
1.0	0.0	0.6	2	10425	2547
1.0	0.0	0.6	3	10425	2525
1.0	0.0	0.7	1	10425	726
1.0	0.0	0.7	2	10425	3703
1.0	0.0	0.7	3	10425	0
1.0	0.0	0.8	1	10425	0
1.0	0.0	0.8	2	10425	0
1.0	0.0	0.8	3	10425	0
1.0	0.0	0.9	1	10425	0

Продолжение таблицы 5.6

1.0	0.0	0.9	2	10425	0
1.0	0.0	0.9	3	10425	0
1.0	0.0	1.0	1	10425	0
1.0	0.0	1.0	2	10425	0
1.0	0.0	1.0	3	10425	0

# Параметризация для графа №3

Таблица 5.7 — Результаты параметризующего запуска для графа №3 из 10 вершин

$\alpha$	$\beta$	$\rho$	$time$	$ideal$	$diff$
0.1	0.9	0.1	2	10755	0
0.1	0.9	0.1	3	10755	601
0.1	0.9	0.2	1	10755	0
0.1	0.9	0.2	2	10755	601
0.1	0.9	0.2	3	10755	601
0.1	0.9	0.3	1	10755	601
0.1	0.9	0.3	2	10755	601
0.1	0.9	0.3	3	10755	601
0.1	0.9	0.4	1	10755	601
0.1	0.9	0.4	2	10755	601
0.1	0.9	0.4	3	10755	810
0.1	0.9	0.5	1	10755	601
0.1	0.9	0.5	2	10755	0
0.1	0.9	0.5	3	10755	810
0.1	0.9	0.6	1	10755	0
0.1	0.9	0.6	2	10755	0
0.1	0.9	0.6	3	10755	601
0.1	0.9	0.7	1	10755	601
0.1	0.9	0.7	2	10755	601
0.1	0.9	0.7	3	10755	0
0.1	0.9	0.8	1	10755	601
0.1	0.9	0.8	2	10755	601
0.1	0.9	0.8	3	10755	601
0.1	0.9	0.9	1	10755	601
0.1	0.9	0.9	2	10755	601
0.1	0.9	0.9	3	10755	601
0.1	0.9	1.0	1	10755	0
0.1	0.9	1.0	2	10755	0
0.1	0.9	1.0	3	10755	601
0.2	0.8	0.1	1	10755	601

Продолжение таблицы 5.7

0.2	0.8	0.1	2	10755	601
0.2	0.8	0.1	3	10755	601
0.2	0.8	0.2	1	10755	601
0.2	0.8	0.2	2	10755	601
0.2	0.8	0.2	3	10755	810
0.2	0.8	0.3	1	10755	601
0.2	0.8	0.3	2	10755	601
0.2	0.8	0.3	3	10755	0
0.2	0.8	0.4	1	10755	4001
0.2	0.8	0.4	2	10755	0
0.2	0.8	0.4	3	10755	601
0.2	0.8	0.5	1	10755	601
0.2	0.8	0.5	2	10755	0
0.2	0.8	0.5	3	10755	601
0.2	0.8	0.6	1	10755	4082
0.2	0.8	0.6	2	10755	0
0.2	0.8	0.6	3	10755	0
0.2	0.8	0.7	1	10755	0
0.2	0.8	0.7	2	10755	810
0.2	0.8	0.7	3	10755	810
0.2	0.8	0.8	1	10755	810
0.2	0.8	0.8	2	10755	601
0.2	0.8	0.8	3	10755	0
0.2	0.8	0.9	1	10755	3453
0.2	0.8	0.9	2	10755	0
0.2	0.8	0.9	3	10755	601
0.2	0.8	1.0	1	10755	4002
0.2	0.8	1.0	2	10755	810
0.2	0.8	1.0	3	10755	0
0.3	0.7	0.1	1	10755	1768
0.3	0.7	0.1	2	10755	601
0.3	0.7	0.1	3	10755	0
0.3	0.7	0.2	1	10755	3970

Продолжение таблицы 5.7

0.3	0.7	0.2	2	10755	810
0.3	0.7	0.2	3	10755	0
0.3	0.7	0.3	1	10755	601
0.3	0.7	0.3	2	10755	0
0.3	0.7	0.3	3	10755	2004
0.3	0.7	0.4	1	10755	2004
0.3	0.7	0.4	2	10755	601
0.3	0.7	0.4	3	10755	0
0.3	0.7	0.5	1	10755	601
0.3	0.7	0.5	2	10755	0
0.3	0.7	0.5	3	10755	810
0.3	0.7	0.6	1	10755	601
0.3	0.7	0.6	2	10755	0
0.3	0.7	0.6	3	10755	0
0.3	0.7	0.7	1	10755	0
0.3	0.7	0.7	2	10755	810
0.3	0.7	0.7	3	10755	601
0.3	0.7	0.8	1	10755	3970
0.3	0.7	0.8	2	10755	601
0.3	0.7	0.8	3	10755	0
0.3	0.7	0.9	1	10755	601
0.3	0.7	0.9	2	10755	810
0.3	0.7	0.9	3	10755	0
0.3	0.7	1.0	1	10755	0
0.3	0.7	1.0	2	10755	0
0.3	0.7	1.0	3	10755	0
0.4	0.6	0.1	1	10755	601
0.4	0.6	0.1	2	10755	601
0.4	0.6	0.1	3	10755	0
0.4	0.6	0.2	1	10755	2004
0.4	0.6	0.2	2	10755	1768
0.4	0.6	0.2	3	10755	601
0.4	0.6	0.3	1	10755	1768

Продолжение таблицы 5.7

0.4	0.6	0.3	2	10755	0
0.4	0.6	0.3	3	10755	601
0.4	0.6	0.4	1	10755	601
0.4	0.6	0.4	2	10755	601
0.4	0.6	0.4	3	10755	601
0.4	0.6	0.5	1	10755	0
0.4	0.6	0.5	2	10755	0
0.4	0.6	0.5	3	10755	0
0.4	0.6	0.6	1	10755	601
0.4	0.6	0.6	2	10755	601
0.4	0.6	0.6	3	10755	0
0.4	0.6	0.7	1	10755	810
0.4	0.6	0.7	2	10755	601
0.4	0.6	0.7	3	10755	601
0.4	0.6	0.8	1	10755	810
0.4	0.6	0.8	2	10755	0
0.4	0.6	0.8	3	10755	601
0.4	0.6	0.9	1	10755	3970
0.4	0.6	0.9	2	10755	601
0.4	0.6	0.9	3	10755	0
0.4	0.6	1.0	1	10755	601
0.4	0.6	1.0	2	10755	601
0.4	0.6	1.0	3	10755	0
0.5	0.5	0.1	1	10755	601
0.5	0.5	0.1	2	10755	0
0.5	0.5	0.1	3	10755	0
0.5	0.5	0.2	1	10755	0
0.5	0.5	0.2	2	10755	0
0.5	0.5	0.2	3	10755	601
0.5	0.5	0.3	1	10755	601
0.5	0.5	0.3	2	10755	1768
0.5	0.5	0.3	3	10755	0
0.5	0.5	0.4	1	10755	601



Продолжение таблицы 5.7

0.5	0.5	0.4	2	10755	0
0.5	0.5	0.4	3	10755	0
0.5	0.5	0.5	1	10755	2665
0.5	0.5	0.5	2	10755	0
0.5	0.5	0.5	3	10755	601
0.5	0.5	0.6	1	10755	601
0.5	0.5	0.6	2	10755	601
0.5	0.5	0.6	3	10755	601
0.5	0.5	0.7	1	10755	3471
0.5	0.5	0.7	2	10755	601
0.5	0.5	0.7	3	10755	601
0.5	0.5	0.8	1	10755	0
0.5	0.5	0.8	2	10755	810
0.5	0.5	0.8	3	10755	0
0.5	0.5	0.9	1	10755	601
0.5	0.5	0.9	2	10755	2984
0.5	0.5	0.9	3	10755	0
0.5	0.5	1.0	1	10755	3970
0.5	0.5	1.0	2	10755	0
0.5	0.5	1.0	3	10755	810
0.6	0.4	0.1	1	10755	1768
0.6	0.4	0.1	2	10755	0
0.6	0.4	0.1	3	10755	0
0.6	0.4	0.2	1	10755	810
0.6	0.4	0.2	2	10755	601
0.6	0.4	0.2	3	10755	601
0.6	0.4	0.3	1	10755	601
0.6	0.4	0.3	2	10755	0
0.6	0.4	0.3	3	10755	0
0.6	0.4	0.4	1	10755	810
0.6	0.4	0.4	2	10755	0
0.6	0.4	0.4	3	10755	601
0.6	0.4	0.5	1	10755	601

Продолжение таблицы 5.7

0.6	0.4	0.5	2	10755	810
0.6	0.4	0.5	3	10755	601
0.6	0.4	0.6	1	10755	810
0.6	0.4	0.6	2	10755	0
0.6	0.4	0.6	3	10755	0
0.6	0.4	0.7	1	10755	601
0.6	0.4	0.7	2	10755	0
0.6	0.4	0.7	3	10755	0
0.6	0.4	0.8	1	10755	601
0.6	0.4	0.8	2	10755	601
0.6	0.4	0.8	3	10755	601
0.6	0.4	0.9	1	10755	3970
0.6	0.4	0.9	2	10755	0
0.6	0.4	0.9	3	10755	0
0.6	0.4	1.0	1	10755	601
0.6	0.4	1.0	2	10755	601
0.6	0.4	1.0	3	10755	601
0.7	0.3	0.1	1	10755	601
0.7	0.3	0.1	2	10755	0
0.7	0.3	0.1	3	10755	0
0.7	0.3	0.2	1	10755	810
0.7	0.3	0.2	2	10755	601
0.7	0.3	0.2	3	10755	0
0.7	0.3	0.3	1	10755	601
0.7	0.3	0.3	2	10755	601
0.7	0.3	0.3	3	10755	0
0.7	0.3	0.4	1	10755	0
0.7	0.3	0.4	2	10755	810
0.7	0.3	0.4	3	10755	601
0.7	0.3	0.5	1	10755	0
0.7	0.3	0.5	2	10755	601
0.7	0.3	0.5	3	10755	0
0.7	0.3	0.6	1	10755	601

Продолжение таблицы 5.7

0.7	0.3	0.6	2	10755	0
0.7	0.3	0.6	3	10755	601
0.7	0.3	0.7	1	10755	0
0.7	0.3	0.7	2	10755	601
0.7	0.3	0.7	3	10755	2004
0.7	0.3	0.8	1	10755	0
0.7	0.3	0.8	2	10755	601
0.7	0.3	0.8	3	10755	601
0.7	0.3	0.9	1	10755	3970
0.7	0.3	0.9	2	10755	601
0.7	0.3	0.9	3	10755	0
0.7	0.3	1.0	1	10755	0
0.7	0.3	1.0	2	10755	1768
0.7	0.3	1.0	3	10755	0
0.8	0.2	0.1	1	10755	3970
0.8	0.2	0.1	2	10755	601
0.8	0.2	0.1	3	10755	601
0.8	0.2	0.2	1	10755	601
0.8	0.2	0.2	2	10755	601
0.8	0.2	0.2	3	10755	810
0.8	0.2	0.3	1	10755	0
0.8	0.2	0.3	2	10755	0
0.8	0.2	0.3	3	10755	810
0.8	0.2	0.4	1	10755	2004
0.8	0.2	0.4	2	10755	0
0.8	0.2	0.4	3	10755	0
0.8	0.2	0.5	1	10755	5694
0.8	0.2	0.5	2	10755	0
0.8	0.2	0.5	3	10755	601
0.8	0.2	0.6	1	10755	601
0.8	0.2	0.6	2	10755	0
0.8	0.2	0.6	3	10755	0
0.8	0.2	0.7	1	10755	601

Продолжение таблицы 5.7

0.8	0.2	0.7	2	10755	601
0.8	0.2	0.7	3	10755	0
0.8	0.2	0.8	1	10755	1768
0.8	0.2	0.8	2	10755	1768
0.8	0.2	0.8	3	10755	601
0.8	0.2	0.9	1	10755	1768
0.8	0.2	0.9	2	10755	601
0.8	0.2	0.9	3	10755	0
0.8	0.2	1.0	1	10755	601
0.8	0.2	1.0	2	10755	0
0.8	0.2	1.0	3	10755	601
0.9	0.1	0.1	1	10755	0
0.9	0.1	0.1	2	10755	0
0.9	0.1	0.1	3	10755	601
0.9	0.1	0.2	1	10755	0
0.9	0.1	0.2	2	10755	601
0.9	0.1	0.2	3	10755	0
0.9	0.1	0.3	1	10755	0
0.9	0.1	0.3	2	10755	0
0.9	0.1	0.3	3	10755	601
0.9	0.1	0.4	1	10755	0
0.9	0.1	0.4	2	10755	0
0.9	0.1	0.4	3	10755	0
0.9	0.1	0.5	1	10755	601
0.9	0.1	0.5	2	10755	601
0.9	0.1	0.5	3	10755	810
0.9	0.1	0.6	1	10755	810
0.9	0.1	0.6	2	10755	0
0.9	0.1	0.6	3	10755	601
0.9	0.1	0.7	1	10755	601
0.9	0.1	0.7	2	10755	810
0.9	0.1	0.7	3	10755	0
0.9	0.1	0.8	1	10755	3970

Продолжение таблицы 5.7

0.9	0.1	0.8	2	10755	2004
0.9	0.1	0.8	3	10755	601
0.9	0.1	0.9	1	10755	601
0.9	0.1	0.9	2	10755	0
0.9	0.1	0.9	3	10755	810
0.9	0.1	1.0	1	10755	601
0.9	0.1	1.0	2	10755	601
0.9	0.1	1.0	3	10755	601
1.0	0.0	0.1	1	10755	2518
1.0	0.0	0.1	2	10755	2504
1.0	0.0	0.1	3	10755	1450
1.0	0.0	0.2	1	10755	2747
1.0	0.0	0.2	2	10755	2504
1.0	0.0	0.2	3	10755	1450
1.0	0.0	0.3	1	10755	2504
1.0	0.0	0.3	2	10755	1450
1.0	0.0	0.3	3	10755	1450
1.0	0.0	0.4	1	10755	2677
1.0	0.0	0.4	2	10755	2747
1.0	0.0	0.4	3	10755	1450
1.0	0.0	0.5	1	10755	2677
1.0	0.0	0.5	2	10755	1450
1.0	0.0	0.5	3	10755	1450
1.0	0.0	0.6	1	10755	2504
1.0	0.0	0.6	2	10755	2518
1.0	0.0	0.6	3	10755	1450
1.0	0.0	0.7	1	10755	1450
1.0	0.0	0.7	2	10755	2518
1.0	0.0	0.7	3	10755	2504
1.0	0.0	0.8	1	10755	1450
1.0	0.0	0.8	2	10755	2504
1.0	0.0	0.8	3	10755	1450
1.0	0.0	0.9	1	10755	2504

Продолжение таблицы 5.7

1.0	0.0	0.9	2	10755	2504
1.0	0.0	0.9	3	10755	2504
1.0	0.0	1.0	1	10755	2747
1.0	0.0	1.0	2	10755	2518
1.0	0.0	1.0	3	10755	2504