



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»

Отчет по лабораторной работе №6 по курсу «Анализ алгоритмов»

Тема Муравьиный алгоритм

Студент Динь Вьет Ань

Группа ИУ7И-54Б

Оценка (баллы) _____

Преподаватель Волкова Л.Л., Строганов Ю.В.

Оглавление

Введение	3
1 Аналитическая часть	4
1.1 Задача коммивояжера	4
1.2 Алгоритм полного перебора для решения задачи коммивояжера	4
1.3 Муравьиный алгоритм для решения задачи коммивояжера .	5
2 Конструкторская часть	7
2.1 Требования к вводу	7
2.2 Требования к программе	7
2.3 Разработка алгоритмов	8
3 Технологическая часть	12
3.1 Выбор языка программирования и среды разработки	12
3.2 Реализация алгоритмов	12
3.3 Тестирование	16
4 Исследовательская часть	17
4.1 Технические характеристики	17
4.2 Примеры работы программы	17
4.3 Время выполнения реализаций алгоритмов	20
4.4 Автоматическая параметризация	20
4.4.1 Класс данных 1	21
4.4.2 Класс данных 2	21
4.5 Вывод	22
Заключение	24
Список использованных источников	25
ПРИЛОЖЕНИЕ А Таблица параметризации муравьиного алгоритма	26

Введение

Задача поиска оптимальных маршрутов является одной из важных. Муравьиный алгоритм – один из эффективных полиномиальных алгоритмов для нахождения приближённых решений задачи коммивояжёра, а также решения аналогичных задач поиска маршрутов на графах. Суть подхода заключается в анализе и использовании модели поведения муравьёв, ищущих пути от колонии к источнику питания, и представляет собой метаэвристическую оптимизацию.

Целью данной лабораторной работы является реализация муравьиного алгоритма и приобретение навыков параметризации методов на примере реализованного алгоритма, примененного к задаче коммивояжера.

Для достижения данной цели необходимо решить следующие задачи.

1. Изучить алгоритм полного перебора для решения задачи коммивояжера.
2. Реализовать алгоритм полного перебора для решения задачи коммивояжера.
3. Изучить муравьиный алгоритм для решения задачи коммивояжера.
4. Реализовать муравьиный алгоритм для решения задачи коммивояжера.
5. Провести параметризацию муравьиного алгоритма на трех классах данных.
6. Провести сравнительный анализ скорости работы реализованных алгоритмов.
7. Подготовить отчет о выполненной лабораторной работе.

1 Аналитическая часть

В данном разделе была рассмотрена задача коммивояжера и были описаны алгоритмы её решения.

1.1 Задача коммивояжера

Цель задачи коммивояжера [1] заключается в нахождении самого выгодного маршрута (кратчайшего, самого быстрого, наиболее дешевого), проходящего через все заданные точки (пункты, города) по одному разу.

Условия задачи должны содержать критерий выгодности маршрута (должен ли он быть максимально коротким, быстрым, дешевым или все вместе), а также исходные данные в виде матрицы затрат (расстояния, стоимости, времени) при перемещении между рассматриваемыми пунктами.

1.2 Алгоритм полного перебора для решения задачи коммивояжера

Рассмотрим n городов и матрицу расстояний между ними. Найдем самый короткий маршрут посещения всех городов ровно по одному разу, без возвращения в первый город.

Алгоритм полного перебора для решения задачи коммивояжера предполагает рассмотрение всех возможных путей в графе и выбор наименьшего из них. Смысл перебора состоит в том, что мы перебираем все варианты объезда городов и выбираем оптимальный. Однако, при таком подходе количество возможных маршрутов очень быстро возрастает с ростом n (сложность алгоритма равна $n!$).

Алгоритм полного перебора гарантирует точное решение задачи, однако, уже при небольшом числе городов будут большие затраты по времени выполнения.

1.3 Муравьиный алгоритм для решения задачи коммивояжера

Идея муравьиного алгоритма [2] — моделирование поведения муравьев, связанное с их способностью быстро находить кратчайший путь и адаптироваться к изменяющимся условиям, находя новый кратчайший путь.

Муравьи действуют согласно следующим правилам.

1. Муравей запоминает посещенные города, причем каждый город может быть посещен только один раз. Обозначим через $J_{i,k}$ список городов, которые посетил муравей k , находящийся в городе i .
2. Муравей обладает видимостью η_{ij} - эвристическим желанием посетить город j , если муравей находится в городе i , причем

$$\eta_{ij} = 1/D_{ij}, \quad (1.1)$$

где D_{ij} — стоимость пути из города i в город j .

3. Муравей может улавливать след феромона - специального химического вещества. Число феромона на пути из города i в город j - τ_{ij} .

Муравей выполняет следующую последовательность действий, пока не посетит все города.

1. Муравей выбирает следующий город назначения, основываясь на вероятностно - пропорциональном правиле (1.2), в котором учитываются видимость и число феромона.

$$P_{ij,k} = \begin{cases} \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{l=1}^m \tau_{il}^\alpha \eta_{il}^\beta}, & \text{если город } j \text{ необходимо посетить;} \\ 0, & \text{иначе,} \end{cases} \quad (1.2)$$

где α - параметр влияния феромона, β - параметр влияния видимости пути, τ_{ij} - число феромона на ребре (ij) , η_{ij} - эвристическое желание посетить город j , если муравей находится в городе i . Выбор города является вероятностным, данное правило определяет ширину зоны

города j , в общую зону всех городов $J_{i,k}$ бросается случайное число, которое и определяет выбор муравья.

2. Муравей проходит путь (ij) и оставляет на нем феромон.

Информация о числе феромона на пути используется другими муравьями для выбора пути. Те муравьи, которые случайно выберут кратчайший путь, будут быстрее его проходить, и за несколько передвижений он будет более обогащен феромоном. Следующие муравьи будут предпочитать именно этот путь, продолжая обогащать его феромоном.

После прохождения маршрутов всеми муравьями значение феромона на путях обновляется в соответствии со следующим правилом (1.3).

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}, \quad (1.3)$$

где ρ - коэффициент испарения. Чтобы найденное локальное решение не было единственным, моделируется испарение феромона.

При этом

$$\Delta\tau_{ij} = \sum_{k=1}^m \tau_{ij,k}, \quad (1.4)$$

где m - число муравьев,

$$\Delta\tau_{ij,k} = \begin{cases} Q/L_k, & \text{если } k\text{-ый муравей прошел путь } (i,j); \\ 0, & \text{иначе.} \end{cases} \quad (1.5)$$

Вывод

Была изучена задача поиска оптимального маршрута, проходящего через все заданные вершины по одному разу. Были рассмотрены подходы к решению замкнутой симметричной задачи коммивояжера.

2 Конструкторская часть

В этом разделе были представлены требования к вводу и программе, а также схемы алгоритма полного перебора и муравьиного алгоритма.

2.1 Требования к вводу

На вход программе должна подаваться матрица стоимостей, которая задает взвешенный неориентированный граф.

2.2 Требования к программе

Выходные данные программы — оптимальный маршрут, проходящий через все заданные вершины по одному разу с последующим возвратом в исходную точку, и его стоимость. Программа должна работать в рамках следующих ограничений:

- стоимости путей должны быть целыми числами;
- число городов должно быть больше 1;
- число дней должно быть больше 0;
- параметры муравьиного алгоритма должны быть вещественными числами, большими 0;
- матрица должна задавать неориентированный граф;
- должно быть выдано сообщение об ошибке при некорректном вводе параметров.

Пользователь должен иметь возможность выбора метода решения — полным перебором или муравьиным алгоритмом, и вывода результата на экран. Кроме того должна быть возможность проведения параметризации

муравьиного алгоритма. Также должны быть реализованы сравнение алгоритмов по времени работы с выводом результатов на экран и получение графического представления результатов сравнения. Данные действия пользователь должен выполнять при помощи меню.

2.3 Разработка алгоритмов

На рисунке 2.1 представлена схема алгоритма полного перебора путей, а на рисунках 2.2 — схема муравьиного алгоритма поиска путей.

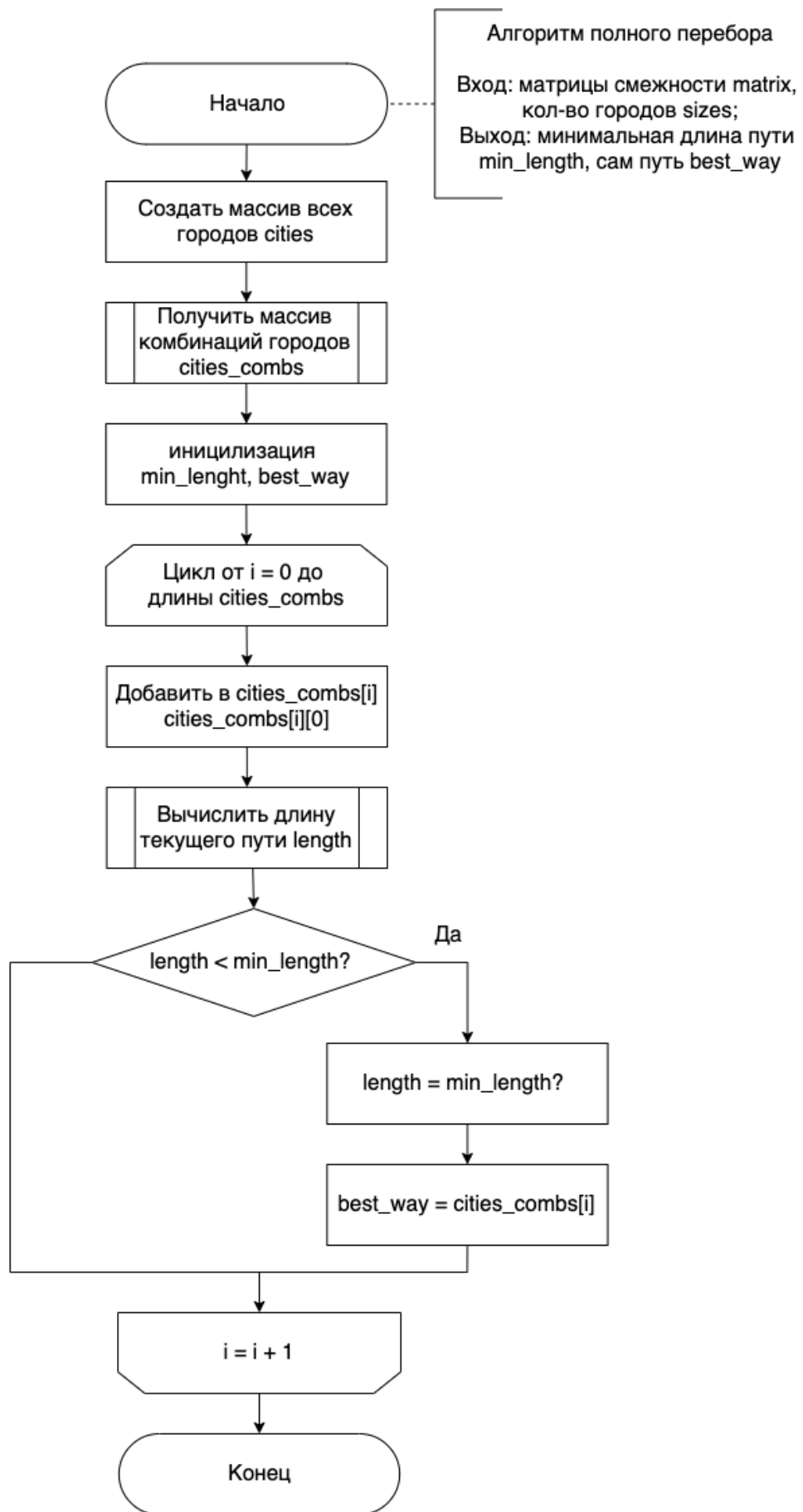


Рисунок 2.1 – Схема алгоритма полного перебора путей

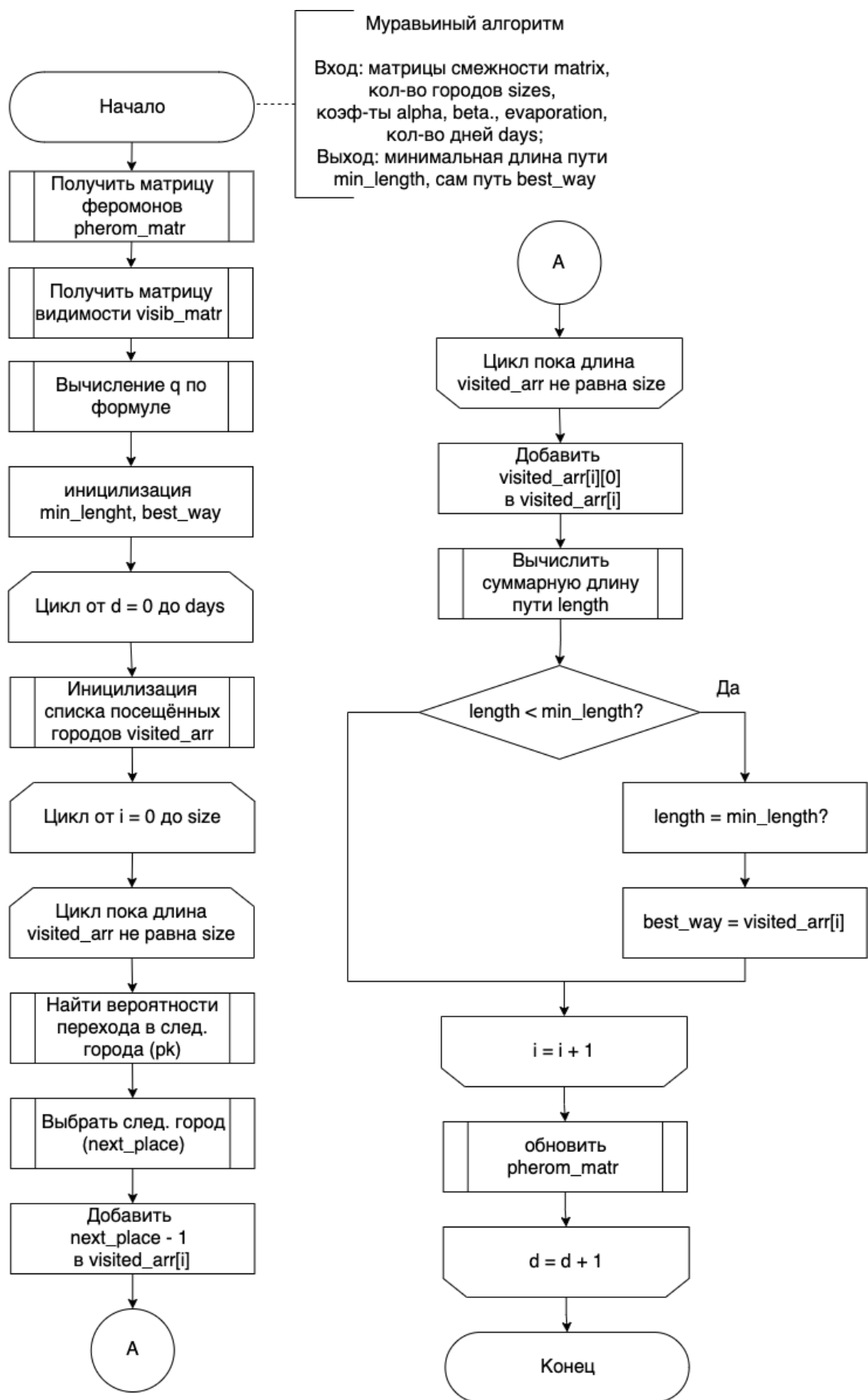


Рисунок 2.2 – Схема муравьиного алгоритма

Вывод

В данном разделе были построены схемы алгоритмов, рассматриваемых в лабораторной работе, были описаны требования ко вводу и программе.

3 Технологическая часть

В данном разделе будут рассмотрены средства реализации лабораторной работы, представлены листинги алгоритмов сортировки, а также тестовые данные, которые использовались для проверки корректности работы алгоритмов.

3.1 Выбор языка программирования и среды разработки

Для реализации этих алгоритмов был выбран язык программирования *Python*, так как он предоставляет необходимую функциональность, позволяющую решить поставленные задачи на реализацию алгоритмов и выполнение замеров процессорного времени работы их реализаций. Для замеров процессорного времени использовалась функция *process_time()* [3] из библиотеки *time*, а для построения графиков – библиотека *matplotlib* [4].

3.2 Реализация алгоритмов

В листингах 3.2 - 3.2 представлены функции для конвейерного и линейного алгоритмов обработки матриц.

Листинг 3.1 – Алгоритм полного перебора

```
1 def full_combinations_alg(matrix, size):
2
3     cities = np.arange(size)
4     cities_combs = []
5
6     for combination in itertools.permutations(cities):
7         cities_combs.append(list(combination))
8
9     best_way = []
10    min_length = float("inf")
11
12    for i in range(len(cities_combs)):
```

```

13         cities_combs[i].append(cities_combs[i][0])
14
15         length = calc_length(matrix, size, cities_combs[i])
16
17         if length < min_length:
18             min_length = length
19             best_way = cities_combs[i]
20
21     return min_length, best_way

```

Листинг 3.2 – Муравьиный алгоритм

```

1 def ant_alg(matrix, size, alpha, beta, evaporation, days):
2
3     pherom_matr = get_pherom_matr(size)
4     visib_matr = get_visib_matr(matrix, size)
5
6     q = calc_q(matrix, size)
7
8     best_way = []
9     min_length = float("inf")
10
11     for _ in range(days):
12         visited_arr = get_visited_places(np.arange(size), size)
13
14         for i in range(size):
15             while (len(visited_arr[i]) != size):
16                 pk = search_probability(pherom_matr, visib_matr,
17                                         visited_arr, size, i, alpha, beta)
18                 next_place = choose_next_place(pk)
19
20                 visited_arr[i].append(next_place - 1)
21
22             visited_arr[i].append(visited_arr[i][0])
23
24             length = calc_length(matrix, size, visited_arr[i])
25
26             if length < min_length:
27                 min_length = length
28                 best_way = visited_arr[i]

```

```

29         pherom_matr = update_pherom_matr(matrix, size,
30             visited_arr, pherom_matr, q, evaporation)
31     return min_length, best_way

```

Листинг 3.3 – Функция для обновления феромонов

```

1 def update_pherom_matr(matrix, size, visited_arr, pherom_matr, q,
2     evaporation):
3     ants = size
4
5     for i in range(size):
6         for j in range(size):
7             delta_pherom_matr = 0
8
9             for ant in range(ants):
10                length = calc_length(matrix, size,
11                    visited_arr[ant])
12                delta_pherom_matr += q / length
13
14                pherom_matr[i][j] *= (1 - evaporation)
15                pherom_matr[i][j] += delta_pherom_matr
16
17                if pherom_matr[i][j] < MIN_PHEROMONE:
18                    pherom_matr[i][j] = MIN_PHEROMONE
19
20    return pherom_matr

```

Листинг 3.4 – Функция для нахождения вероятней перехода в каждый из городов

```

1 def search_probability(pherom_matr, visib_matr, visited_arr,
2     size, ant, alpha, beta):
3     pk = [0] * size
4
5     for i in range(size):
6         if i not in visited_arr[ant]:
7             ant_i = visited_arr[ant][-1]
8
9             pk[i] = pow(pherom_matr[ant_i][i], alpha) * \

```

```

9             pow(visib_matr[ant_i][i], beta)
10         else:
11             pk[i] = 0
12
13     pk_sum = sum(pk)
14
15     for i in range(size):
16         pk[i] /= pk_sum
17
18     return pk

```

Листинг 3.5 – Функция выбора следующего города

```

1 def choose_next_place(pk):
2
3     size = len(pk)
4     numb = 0
5     i = 0
6
7     probability = random()
8
9     while numb < probability and i < size:
10         numb += pk[i]
11         i += 1
12
13     return i

```

Листинг 3.6 – Функция нахождения длины пути

```

1 def calc_length(matrix, size, way):
2
3     length = 0
4
5     for i in range(size):
6         beg_city = way[i]
7         end_city = way[i + 1]
8
9         length += matrix[beg_city][end_city]
10
11     return length

```

3.3 Тестирование

В таблице 3.1 приведены тесты для алгоритмов решения задачи коммивояжера (муравьиного алгоритма и алгоритма полного перебора). Все тесты пройдены успешно.

Таблица 3.1 – Функциональные тесты

Входные данные	Выходные данные	
Матрица смежности	Результат	Ожидаемый результат
$\begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 3 \\ 2 & 3 & 0 \end{pmatrix}$	6, [0, 1, 2, 0]	6, [0, 1, 2, 0]
$\begin{pmatrix} 0 & 3 & 4 & 1 \\ 3 & 0 & 1 & 1 \\ 4 & 1 & 0 & 2 \\ 1 & 1 & 2 & 0 \end{pmatrix}$	7, [0, 1, 2, 3, 0]	7, [0, 1, 2, 3, 0]
$\begin{pmatrix} 0 & 11 & 12 & 14 & 13 \\ 11 & 0 & 15 & 10 & 10 \\ 12 & 15 & 0 & 14 & 13 \\ 14 & 10 & 14 & 0 & 10 \\ 13 & 10 & 13 & 10 & 0 \end{pmatrix}$	56, [0, 1, 3, 4, 2, 0]	56, [0, 1, 3, 4, 2, 0]

Вывод

В данном разделе были рассмотрены средства реализации лабораторной работы, представлены листинги алгоритмов сортировки, а также тестовые данные, которые использовались для проверки корректности работы алгоритмов.

4 Исследовательская часть

В данном разделе будут предоставлена информация о технических характеристиках устройства, представлены примеры работы программы и проведены замеры процессорного времени.

4.1 Технические характеристики

Технические характеристики устройства, на котором выполнялось тестирование:

- операционная система Window 10 Home Single Language;
- память 8 Гб;
- процессор 11th Gen Intel(R) Core(TM) i5-1135G7 2.42 ГГц, 4 ядра.

Во время замера устройство было подключено к сети электропитания, нагружено приложениями окружения и самой системой замера.

4.2 Примеры работы программы

На рисунке 4.1–4.2 представлен результат работы программы.

Меню

1. Полный перебор
2. Муравьиный алгоритм
3. Параметризация
4. Замеры времени
5. Распечатать матрицу
0. Выход

Выбор: 5

0	4	1	8	9
4	0	4	8	6
1	4	0	7	5
8	8	7	0	8
9	6	5	8	0

Меню

1. Полный перебор
2. Муравьиный алгоритм
3. Параметризация
4. Замеры времени
5. Распечатать матрицу
0. Выход

Выбор: 1

Минимальная сумма пути: 26

Минимальный путь: [0, 1, 3, 4, 2, 0]

Рисунок 4.1 – Пример работы программы с выбором 1

0	4	1	8	9
4	0	4	8	6
1	4	0	7	5
8	8	7	0	8
9	6	5	8	0

Меню

1. Полный перебор
2. Муравьиный алгоритм
3. Параметризация
4. Замеры времени
5. Распечатать матрицу
0. Выход

Выбор: 2

Коэффициент α = 0.6

Коэффициент β = 0.4

Коэффициент evaporation = 0.6

Введите кол-во дней: 20

Минимальная сумма пути: 26

Минимальный путь: [2, 3, 4, 1, 0, 2]

Рисунок 4.2 – Пример работы программы с выбором 2

4.3 Время выполнения реализаций алгоритмов

Результаты замеров времени работы алгоритмов решения задачи коммивояжера представлены на рисунках 4.3. Замеры времени проводились в секундах и усреднялись для каждого набора одинаковых экспериментов.

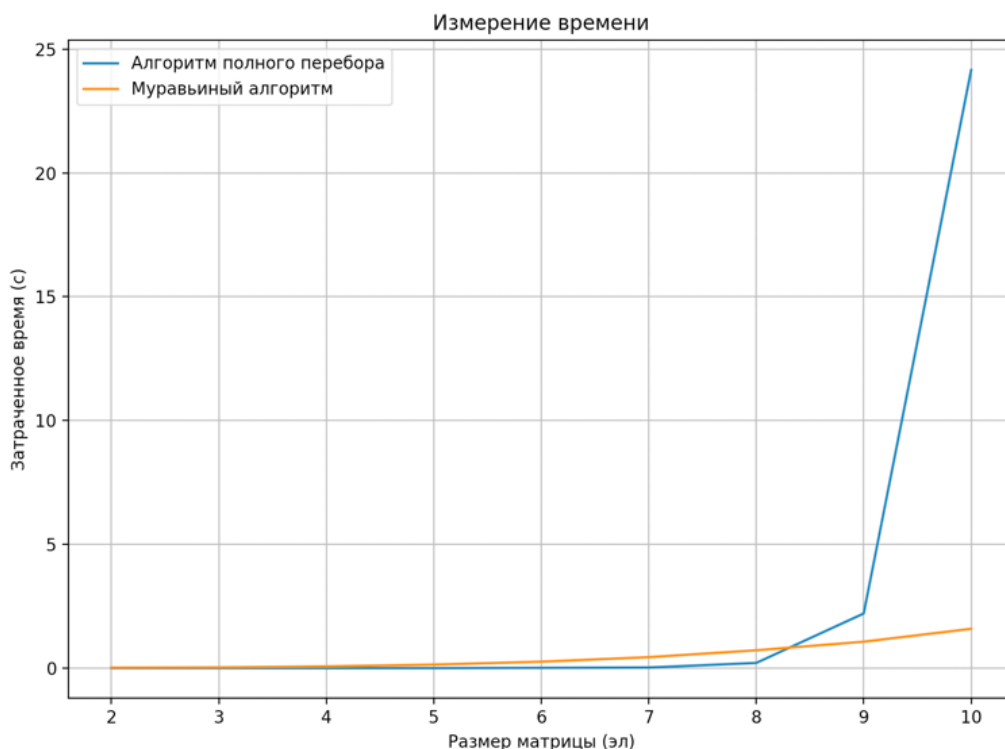


Рисунок 4.3 – Зависимость времени работы реализаций алгоритмов от размера матрицы

4.4 Автоматическая параметризация

Автоматическая параметризация была проведена на двух классах данных. Для проведения эксперимента были взяты матрицы размером 10x10. Муравьиный алгоритм был запущен для всех значений $\alpha, \rho \in (0, 1)$ с шагом 0.1.

В качестве эталонного значения был взят результат работы алгоритма полного перебора.

Далее будут представлены матрицы смежности (матрица 4.1 для первого класса данных и 4.2 для второго), на которых происходила параметризация и таблицы с результатами её выполнения.

4.4.1 Класс данных 1

В качестве первого класса данных была взята матрица смежности, в которой все значения незначительно отличаются друг от друга, находятся в диапазоне $[1, 3]$. Таблица с результатами параметризации представлена в приложении А.

$$M_1 = \begin{pmatrix} 0 & 2 & 1 & 1 & 3 & 3 & 2 & 2 & 3 & 3 \\ 2 & 0 & 3 & 3 & 3 & 1 & 3 & 2 & 2 & 3 \\ 1 & 3 & 0 & 2 & 3 & 3 & 1 & 1 & 2 & 3 \\ 1 & 3 & 2 & 0 & 2 & 3 & 2 & 3 & 1 & 3 \\ 3 & 3 & 3 & 2 & 0 & 3 & 2 & 3 & 1 & 1 \\ 3 & 1 & 3 & 3 & 3 & 0 & 1 & 1 & 2 & 1 \\ 2 & 3 & 1 & 2 & 2 & 1 & 0 & 2 & 1 & 2 \\ 2 & 2 & 1 & 3 & 3 & 1 & 2 & 0 & 2 & 3 \\ 3 & 2 & 2 & 1 & 1 & 2 & 1 & 2 & 0 & 2 \\ 3 & 3 & 3 & 3 & 1 & 1 & 2 & 3 & 2 & 0 \end{pmatrix} \quad (4.1)$$

4.4.2 Класс данных 2

В качестве второго класса данных была взята матрица смежности, в которой все значения отличаются на большое значение друг от друга, находятся в диапазоне $[1, 1000]$. Таблица с результатами параметризации представлена в приложении А.

$$M_2 = \begin{pmatrix} 0 & 157 & 611 & 117 & 341 & 452 & 579 & 773 & 370 & 343 \\ 157 & 0 & 170 & 696 & 238 & 669 & 302 & 633 & 111 & 427 \\ 611 & 170 & 0 & 95 & 829 & 14 & 661 & 118 & 871 & 754 \\ 117 & 696 & 95 & 0 & 37 & 535 & 308 & 996 & 419 & 456 \\ 341 & 238 & 829 & 37 & 0 & 854 & 454 & 908 & 806 & 455 \\ 452 & 669 & 14 & 535 & 854 & 0 & 646 & 262 & 400 & 799 \\ 579 & 302 & 661 & 308 & 454 & 646 & 0 & 139 & 982 & 423 \\ 773 & 633 & 118 & 996 & 908 & 262 & 139 & 0 & 887 & 59 \\ 370 & 111 & 871 & 419 & 806 & 400 & 982 & 887 & 0 & 578 \\ 343 & 427 & 754 & 456 & 455 & 799 & 423 & 59 & 578 & 0 \end{pmatrix} \quad (4.2)$$

4.5 Вывод

В этом разделе были указаны технические характеристики машины, на которой происходило сравнение времени работы алгоритмов (муравьиного алгоритма и алгоритма полного перебора) решения задачи коммивояжера, также была рассмотрена автоматическая параметризация.

В результате замеров времени было установлено, что муравьиный алгоритм работает хуже алгоритма полного перебора на матрицах, размер которых меньше 9 (при размере 8 он работает хуже в 3.7 раза). Но при больших размерах муравьиный алгоритм существенно превосходит алгоритм полного перебора (на матрицах 9x9 он лучше в 2.1 раз, а на матрицах 10x10 уже в 15.4 раза). Во время проведения замеров времени кол-во дней для муравьиного алгоритма было взято 300.

На основе проведённой параметризации по двум классам данных можно сделать следующие выводы.

Для первого класса данных 4.1 лучше всего подходят следующие параметры:

- $\alpha = 0.1, \beta = 0.9, \rho = 0.2, 0.5;$
- $\alpha = 0.2, \beta = 0.8, \rho = 0.3, 0.4, 0.6;$
- $\alpha = 0.3, \beta = 0.7, \rho = 0.2;$

Для второго класса данных 4.2 лучше всего подходят следующие коэффициенты:

- $\alpha = 0.1, \beta = 0.9, \rho = 0.1, 0.6, 0.8;$
- $\alpha = 0.2, \beta = 0.8, \rho = 0.5;$
- $\alpha = 0.3, \beta = 0.7, \rho = 0.4;$

Таким образом, можно сделать вывод о том, что для лучшей работы муравьиного алгоритма на используемых классах данных (4.1 и 4.2) нужно использовать полученные коэффициенты.

Заключение

Было экспериментально подтверждено различие во временной эффективности муравьиного алгоритма и алгоритма полного перебора решения задачи коммивояжера. В результате исследований можно сделать вывод о том, что при матрицах большого размера (больше 9) стоит использовать муравьиный алгоритм решения задачи коммивояжера, а не алгоритм полного перебора (на матрице размером 10x10 он работает в 15.4 раза быстрее). Также было установлено по результатам параметризации на экспериментальных класса данных, что при коэффициенте $\alpha = 0.1, 0.2, 0.3$ муравьиный алгоритм работает наилучшим образом.

Цель, поставленная перед началом работы, была достигнута. В ходе лабораторной работы были решены все задачи:

- изучены основы алгоритма полного перебора;
- применены изученные основы алгоритма полного перебора для реализации;
- изучены основы муравьиного алгоритма;
- применены изученные основы муравьиного алгоритма для реализации;
- проведена параметризация муравьиного алгоритма;
- проведен сравнительный анализ времени работы реализованных алгоритмов;
- подготовлен отчет о лабораторной работе.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Задача коммивояжера - метод ветвей и границ. Режим доступа: <http://galyautdinov.ru/post/zadacha-kommivoyazhera> (дата обращения: 22.04.2023).
2. М.В. Ульянов Ресурсно-эффективные компьютерные алгоритмы. Разработка и анализ. 2007.
3. Time access and conversions [Электронный ресурс]. Режим доступа: https://docs.python.org/3/library/time.html#time.process_time (дата обращения: 22.04.2023).
4. Matplotlib documentation [Электронный ресурс]. Режим доступа: <https://matplotlib.org/stable/index.html> (дата обращения: 22.04.2023).

ПРИЛОЖЕНИЕ А

В таблицах А.1 – А.2 представлены результаты параметризации муравьиного алгоритма.

Таблица А.1 – Параметры для класса данных 1

α	β	ρ	Дней	Результат	Ошибка
0.1	0.9	0.1	50	12	1
0.1	0.9	0.1	100	12	1
0.1	0.9	0.1	200	12	0
0.1	0.9	0.2	50	12	0
0.1	0.9	0.2	100	12	0
0.1	0.9	0.2	200	12	0
0.1	0.9	0.3	50	12	1
0.1	0.9	0.3	100	12	1
0.1	0.9	0.3	200	12	0
0.1	0.9	0.4	50	12	1
0.1	0.9	0.4	100	12	0
0.1	0.9	0.4	200	12	0
0.1	0.9	0.5	50	12	0
0.1	0.9	0.5	100	12	0
0.1	0.9	0.5	200	12	0
0.1	0.9	0.6	50	12	1
0.1	0.9	0.6	100	12	1
0.1	0.9	0.6	200	12	0
0.1	0.9	0.7	50	12	1
0.1	0.9	0.7	100	12	0
0.1	0.9	0.7	200	12	0
0.1	0.9	0.8	50	12	1
0.1	0.9	0.8	100	12	0
0.1	0.9	0.8	200	12	0
0.2	0.8	0.1	50	12	0
0.2	0.8	0.1	100	12	1
0.2	0.8	0.1	200	12	1

Продолжение таблицы А.1

α	β	ρ	Дней	Результат	Ошибка
0.2	0.8	0.2	50	12	1
0.2	0.8	0.2	100	12	0
0.2	0.8	0.2	200	12	0
0.2	0.8	0.3	50	12	0
0.2	0.8	0.3	100	12	0
0.2	0.8	0.3	200	12	0
0.2	0.8	0.4	50	12	0
0.2	0.8	0.4	100	12	0
0.2	0.8	0.4	200	12	0
0.2	0.8	0.5	50	12	1
0.2	0.8	0.5	100	12	0
0.2	0.8	0.5	200	12	1
0.2	0.8	0.6	50	12	0
0.2	0.8	0.6	100	12	0
0.2	0.8	0.6	200	12	0
0.2	0.8	0.7	50	12	1
0.2	0.8	0.7	100	12	1
0.2	0.8	0.7	200	12	0
0.2	0.8	0.8	50	12	1
0.2	0.8	0.8	100	12	0
0.2	0.8	0.8	200	12	0
0.3	0.7	0.1	50	12	0
0.3	0.7	0.1	100	12	1
0.3	0.7	0.1	200	12	1
0.3	0.7	0.2	50	12	0
0.3	0.7	0.2	100	12	0
0.3	0.7	0.2	200	12	0
0.3	0.7	0.3	50	12	1
0.3	0.7	0.3	100	12	1
0.3	0.7	0.3	200	12	0
0.3	0.7	0.4	50	12	1
0.3	0.7	0.4	100	12	0
0.3	0.7	0.4	200	12	1
0.3	0.7	0.5	50	12	0
0.3	0.7	0.5	100	12	1
0.3	0.7	0.5	200	12	0
0.3	0.7	0.6	50	12	2
0.3	0.7	0.6	100	12	1
0.3	0.7	0.6	200	12	1

Продолжение таблицы А.1

α	β	ρ	Дней	Результат	Ошибка
0.3	0.7	0.7	50	12	1
0.3	0.7	0.7	100	12	1
0.3	0.7	0.7	200	12	0
0.3	0.7	0.8	50	12	1
0.3	0.7	0.8	100	12	1
0.3	0.7	0.8	200	12	1
0.4	0.6	0.1	50	12	1
0.4	0.6	0.1	100	12	0
0.4	0.6	0.1	200	12	0
0.4	0.6	0.2	50	12	1
0.4	0.6	0.2	100	12	0
0.4	0.6	0.2	200	12	1
0.4	0.6	0.3	50	12	0
0.4	0.6	0.3	100	12	0
0.4	0.6	0.3	200	12	1
0.4	0.6	0.4	50	12	1
0.4	0.6	0.4	100	12	0
0.4	0.6	0.4	200	12	0
0.4	0.6	0.5	50	12	1
0.4	0.6	0.5	100	12	1
0.4	0.6	0.5	200	12	0
0.4	0.6	0.6	50	12	0
0.4	0.6	0.6	100	12	1
0.4	0.6	0.6	200	12	0
0.4	0.6	0.7	50	12	2
0.4	0.6	0.7	100	12	1
0.4	0.6	0.7	200	12	0
0.4	0.6	0.8	50	12	1
0.4	0.6	0.8	100	12	1
0.4	0.6	0.8	200	12	0
0.5	0.5	0.1	50	12	2
0.5	0.5	0.1	100	12	0
0.5	0.5	0.1	200	12	1
0.5	0.5	0.2	50	12	1
0.5	0.5	0.2	100	12	0
0.5	0.5	0.2	200	12	1
0.5	0.5	0.3	50	12	2
0.5	0.5	0.3	100	12	1
0.5	0.5	0.3	200	12	1

Продолжение таблицы А.1

α	β	ρ	Дней	Результат	Ошибка
0.5	0.5	0.4	50	12	2
0.5	0.5	0.4	100	12	1
0.5	0.5	0.4	200	12	0
0.5	0.5	0.5	50	12	2
0.5	0.5	0.5	100	12	1
0.5	0.5	0.5	200	12	1
0.5	0.5	0.6	50	12	2
0.5	0.5	0.6	100	12	1
0.5	0.5	0.6	200	12	1
0.5	0.5	0.7	50	12	2
0.5	0.5	0.7	100	12	0
0.5	0.5	0.7	200	12	0
0.5	0.5	0.8	50	12	1
0.5	0.5	0.8	100	12	1
0.5	0.5	0.8	200	12	1
0.6	0.4	0.1	50	12	1
0.6	0.4	0.1	100	12	0
0.6	0.4	0.1	200	12	1
0.6	0.4	0.2	50	12	2
0.6	0.4	0.2	100	12	1
0.6	0.4	0.2	200	12	1
0.6	0.4	0.3	50	12	1
0.6	0.4	0.3	100	12	2
0.6	0.4	0.3	200	12	1
0.6	0.4	0.4	50	12	1
0.6	0.4	0.4	100	12	0
0.6	0.4	0.4	200	12	0
0.6	0.4	0.5	50	12	1
0.6	0.4	0.5	100	12	1
0.6	0.4	0.5	200	12	1
0.6	0.4	0.6	50	12	1
0.6	0.4	0.6	100	12	1
0.6	0.4	0.6	200	12	1
0.6	0.4	0.7	50	12	1
0.6	0.4	0.7	100	12	1
0.6	0.4	0.7	200	12	1
0.6	0.4	0.8	50	12	1
0.6	0.4	0.8	100	12	2
0.6	0.4	0.8	200	12	1

Продолжение таблицы А.1

α	β	ρ	Дней	Результат	Ошибка
0.7	0.3	0.1	50	12	2
0.7	0.3	0.1	100	12	1
0.7	0.3	0.1	200	12	1
0.7	0.3	0.2	50	12	1
0.7	0.3	0.2	100	12	1
0.7	0.3	0.2	200	12	1
0.7	0.3	0.3	50	12	2
0.7	0.3	0.3	100	12	1
0.7	0.3	0.3	200	12	1
0.7	0.3	0.4	50	12	2
0.7	0.3	0.4	100	12	1
0.7	0.3	0.4	200	12	0
0.7	0.3	0.5	50	12	2
0.7	0.3	0.5	100	12	1
0.7	0.3	0.5	200	12	1
0.7	0.3	0.6	50	12	3
0.7	0.3	0.6	100	12	2
0.7	0.3	0.6	200	12	1
0.7	0.3	0.7	50	12	2
0.7	0.3	0.7	100	12	2
0.7	0.3	0.7	200	12	1
0.7	0.3	0.8	50	12	2
0.7	0.3	0.8	100	12	1
0.7	0.3	0.8	200	12	0
0.8	0.2	0.1	50	12	1
0.8	0.2	0.1	100	12	2
0.8	0.2	0.1	200	12	1
0.8	0.2	0.2	50	12	2
0.8	0.2	0.2	100	12	2
0.8	0.2	0.2	200	12	1
0.8	0.2	0.3	50	12	1
0.8	0.2	0.3	100	12	2
0.8	0.2	0.3	200	12	1
0.8	0.2	0.4	50	12	1
0.8	0.2	0.4	100	12	2
0.8	0.2	0.4	200	12	1
0.8	0.2	0.5	50	12	2
0.8	0.2	0.5	100	12	1
0.8	0.2	0.5	200	12	0

Продолжение таблицы А.1

α	β	ρ	Дней	Результат	Ошибка
0.8	0.2	0.6	50	12	2
0.8	0.2	0.6	100	12	1
0.8	0.2	0.6	200	12	0
0.8	0.2	0.7	50	12	2
0.8	0.2	0.7	100	12	2
0.8	0.2	0.7	200	12	2
0.8	0.2	0.8	50	12	0
0.8	0.2	0.8	100	12	1
0.8	0.2	0.8	200	12	1
0.9	0.1	0.1	50	12	2
0.9	0.1	0.1	100	12	1
0.9	0.1	0.1	200	12	2
0.9	0.1	0.2	50	12	2
0.9	0.1	0.2	100	12	1
0.9	0.1	0.2	200	12	0
0.9	0.1	0.3	50	12	3
0.9	0.1	0.3	100	12	1
0.9	0.1	0.3	200	12	1
0.9	0.1	0.4	50	12	3
0.9	0.1	0.4	100	12	1
0.9	0.1	0.4	200	12	2
0.9	0.1	0.5	50	12	2
0.9	0.1	0.5	100	12	2
0.9	0.1	0.5	200	12	2
0.9	0.1	0.6	50	12	2
0.9	0.1	0.6	100	12	2
0.9	0.1	0.6	200	12	1
0.9	0.1	0.7	50	12	2
0.9	0.1	0.7	100	12	1
0.9	0.1	0.7	200	12	1
0.9	0.1	0.8	50	12	2
0.9	0.1	0.8	100	12	2
0.9	0.1	0.8	200	12	1

Таблица А.2 – Параметры для класса данных 2

α	β	ρ	Дней	Результат	Ошибка
0.1	0.9	0.1	50	1809	0
0.1	0.9	0.1	100	1809	0
0.1	0.9	0.1	200	1809	0
0.1	0.9	0.2	50	1809	0
0.1	0.9	0.2	100	1809	32
0.1	0.9	0.2	200	1809	32
0.1	0.9	0.3	50	1809	0
0.1	0.9	0.3	100	1809	0
0.1	0.9	0.3	200	1809	81
0.1	0.9	0.4	50	1809	32
0.1	0.9	0.4	100	1809	0
0.1	0.9	0.4	200	1809	0
0.1	0.9	0.5	50	1809	32
0.1	0.9	0.5	100	1809	0
0.1	0.9	0.5	200	1809	0
0.1	0.9	0.6	50	1809	0
0.1	0.9	0.6	100	1809	0
0.1	0.9	0.6	200	1809	0
0.1	0.9	0.7	50	1809	205
0.1	0.9	0.7	100	1809	32
0.1	0.9	0.7	200	1809	0
0.1	0.9	0.8	50	1809	0
0.1	0.9	0.8	100	1809	0
0.1	0.9	0.8	200	1809	0
0.2	0.8	0.1	50	1809	0
0.2	0.8	0.1	100	1809	32
0.2	0.8	0.1	200	1809	0
0.2	0.8	0.2	50	1809	146
0.2	0.8	0.2	100	1809	0
0.2	0.8	0.2	200	1809	0
0.2	0.8	0.3	50	1809	32
0.2	0.8	0.3	100	1809	0
0.2	0.8	0.3	200	1809	0
0.2	0.8	0.4	50	1809	81
0.2	0.8	0.4	100	1809	32
0.2	0.8	0.4	200	1809	0
0.2	0.8	0.5	50	1809	0
0.2	0.8	0.5	100	1809	0
0.2	0.8	0.5	200	1809	0

Продолжение таблицы А.2

α	β	ρ	Дней	Результат	Ошибка
0.2	0.8	0.6	50	1809	32
0.2	0.8	0.6	100	1809	32
0.2	0.8	0.6	200	1809	0
0.2	0.8	0.7	50	1809	32
0.2	0.8	0.7	100	1809	0
0.2	0.8	0.7	200	1809	0
0.2	0.8	0.8	50	1809	162
0.2	0.8	0.8	100	1809	0
0.2	0.8	0.8	200	1809	0
0.3	0.7	0.1	50	1809	81
0.3	0.7	0.1	100	1809	0
0.3	0.7	0.1	200	1809	32
0.3	0.7	0.2	50	1809	131
0.3	0.7	0.2	100	1809	0
0.3	0.7	0.2	200	1809	0
0.3	0.7	0.3	50	1809	32
0.3	0.7	0.3	100	1809	0
0.3	0.7	0.3	200	1809	0
0.3	0.7	0.4	50	1809	0
0.3	0.7	0.4	100	1809	0
0.3	0.7	0.4	200	1809	0
0.3	0.7	0.5	50	1809	208
0.3	0.7	0.5	100	1809	32
0.3	0.7	0.5	200	1809	0
0.3	0.7	0.6	50	1809	131
0.3	0.7	0.6	100	1809	32
0.3	0.7	0.6	200	1809	0
0.3	0.7	0.7	50	1809	32
0.3	0.7	0.7	100	1809	0
0.3	0.7	0.7	200	1809	0
0.3	0.7	0.8	50	1809	131
0.3	0.7	0.8	100	1809	0
0.3	0.7	0.8	200	1809	32
0.4	0.6	0.1	50	1809	81
0.4	0.6	0.1	100	1809	0
0.4	0.6	0.1	200	1809	0
0.4	0.6	0.2	50	1809	81
0.4	0.6	0.2	100	1809	159
0.4	0.6	0.2	200	1809	0

Продолжение таблицы А.2

α	β	ρ	Дней	Результат	Ошибка
0.4	0.6	0.3	50	1809	0
0.4	0.6	0.3	100	1809	161
0.4	0.6	0.3	200	1809	0
0.4	0.6	0.4	50	1809	32
0.4	0.6	0.4	100	1809	270
0.4	0.6	0.4	200	1809	0
0.4	0.6	0.5	50	1809	188
0.4	0.6	0.5	100	1809	32
0.4	0.6	0.5	200	1809	0
0.4	0.6	0.6	50	1809	81
0.4	0.6	0.6	100	1809	81
0.4	0.6	0.6	200	1809	32
0.4	0.6	0.7	50	1809	32
0.4	0.6	0.7	100	1809	188
0.4	0.6	0.7	200	1809	32
0.4	0.6	0.8	50	1809	378
0.4	0.6	0.8	100	1809	161
0.4	0.6	0.8	200	1809	32
0.5	0.5	0.1	50	1809	146
0.5	0.5	0.1	100	1809	0
0.5	0.5	0.1	200	1809	32
0.5	0.5	0.2	50	1809	273
0.5	0.5	0.2	100	1809	0
0.5	0.5	0.2	200	1809	0
0.5	0.5	0.3	50	1809	131
0.5	0.5	0.3	100	1809	32
0.5	0.5	0.3	200	1809	0
0.5	0.5	0.4	50	1809	205
0.5	0.5	0.4	100	1809	0
0.5	0.5	0.4	200	1809	32
0.5	0.5	0.5	50	1809	223
0.5	0.5	0.5	100	1809	159
0.5	0.5	0.5	200	1809	0
0.5	0.5	0.6	50	1809	224
0.5	0.5	0.6	100	1809	81
0.5	0.5	0.6	200	1809	0
0.5	0.5	0.7	50	1809	289
0.5	0.5	0.7	100	1809	0
0.5	0.5	0.7	200	1809	0

Продолжение таблицы А.2

α	β	ρ	Дней	Результат	Ошибка
0.5	0.5	0.8	50	1809	205
0.5	0.5	0.8	100	1809	32
0.5	0.5	0.8	200	1809	81
0.6	0.4	0.1	50	1809	219
0.6	0.4	0.1	100	1809	131
0.6	0.4	0.1	200	1809	0
0.6	0.4	0.2	50	1809	162
0.6	0.4	0.2	100	1809	237
0.6	0.4	0.2	200	1809	146
0.6	0.4	0.3	50	1809	81
0.6	0.4	0.3	100	1809	161
0.6	0.4	0.3	200	1809	0
0.6	0.4	0.4	50	1809	269
0.6	0.4	0.4	100	1809	81
0.6	0.4	0.4	200	1809	131
0.6	0.4	0.5	50	1809	349
0.6	0.4	0.5	100	1809	219
0.6	0.4	0.5	200	1809	0
0.6	0.4	0.6	50	1809	32
0.6	0.4	0.6	100	1809	0
0.6	0.4	0.6	200	1809	81
0.6	0.4	0.7	50	1809	0
0.6	0.4	0.7	100	1809	0
0.6	0.4	0.7	200	1809	32
0.6	0.4	0.8	50	1809	224
0.6	0.4	0.8	100	1809	224
0.6	0.4	0.8	200	1809	131
0.7	0.3	0.1	50	1809	384
0.7	0.3	0.1	100	1809	146
0.7	0.3	0.1	200	1809	287
0.7	0.3	0.2	50	1809	0
0.7	0.3	0.2	100	1809	287
0.7	0.3	0.2	200	1809	0
0.7	0.3	0.3	50	1809	341
0.7	0.3	0.3	100	1809	237
0.7	0.3	0.3	200	1809	341
0.7	0.3	0.4	50	1809	239
0.7	0.3	0.4	100	1809	389
0.7	0.3	0.4	200	1809	81

Продолжение таблицы А.2

α	β	ρ	Дней	Результат	Ошибка
0.7	0.3	0.5	50	1809	310
0.7	0.3	0.5	100	1809	353
0.7	0.3	0.5	200	1809	0
0.7	0.3	0.6	50	1809	313
0.7	0.3	0.6	100	1809	269
0.7	0.3	0.6	200	1809	0
0.7	0.3	0.7	50	1809	555
0.7	0.3	0.7	100	1809	225
0.7	0.3	0.7	200	1809	0
0.7	0.3	0.8	50	1809	470
0.7	0.3	0.8	100	1809	316
0.7	0.3	0.8	200	1809	243
0.8	0.2	0.1	50	1809	81
0.8	0.2	0.1	100	1809	569
0.8	0.2	0.1	200	1809	389
0.8	0.2	0.2	50	1809	162
0.8	0.2	0.2	100	1809	225
0.8	0.2	0.2	200	1809	348
0.8	0.2	0.3	50	1809	411
0.8	0.2	0.3	100	1809	450
0.8	0.2	0.3	200	1809	287
0.8	0.2	0.4	50	1809	539
0.8	0.2	0.4	100	1809	32
0.8	0.2	0.4	200	1809	237
0.8	0.2	0.5	50	1809	489
0.8	0.2	0.5	100	1809	32
0.8	0.2	0.5	200	1809	161
0.8	0.2	0.6	50	1809	208
0.8	0.2	0.6	100	1809	254
0.8	0.2	0.6	200	1809	146
0.8	0.2	0.7	50	1809	559
0.8	0.2	0.7	100	1809	161
0.8	0.2	0.7	200	1809	237
0.8	0.2	0.8	50	1809	161
0.8	0.2	0.8	100	1809	487
0.8	0.2	0.8	200	1809	188
0.9	0.1	0.1	50	1809	223
0.9	0.1	0.1	100	1809	146
0.9	0.1	0.1	200	1809	254

Продолжение таблицы А.2

α	β	ρ	Дней	Результат	Ошибка
0.9	0.1	0.2	50	1809	433
0.9	0.1	0.2	100	1809	529
0.9	0.1	0.2	200	1809	413
0.9	0.1	0.3	50	1809	423
0.9	0.1	0.3	100	1809	458
0.9	0.1	0.3	200	1809	161
0.9	0.1	0.4	50	1809	482
0.9	0.1	0.4	100	1809	420
0.9	0.1	0.4	200	1809	387
0.9	0.1	0.5	50	1809	672
0.9	0.1	0.5	100	1809	319
0.9	0.1	0.5	200	1809	81
0.9	0.1	0.6	50	1809	322
0.9	0.1	0.6	100	1809	612
0.9	0.1	0.6	200	1809	146
0.9	0.1	0.7	50	1809	349
0.9	0.1	0.7	100	1809	467
0.9	0.1	0.7	200	1809	310
0.9	0.1	0.8	50	1809	237
0.9	0.1	0.8	100	1809	703
0.9	0.1	0.8	200	1809	387