



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»
на тему: «Методы решения задачи коммивояжёра»

Студент ИУ7-55Б
(Группа)

(Подпись, дата)

Половинкин И. Д.
(Фамилия И. О.)

Преподаватель

(Подпись, дата)

Волкова Л. Л.
(Фамилия И. О.)

2025 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Аналитическая часть	4
1.1 Задача коммивояжера	4
1.2 Методы решения задачи коммивояжера	4
2 Конструкторская часть	6
2.1 Схемы алгоритмов	6
2.2 Оценка трудоемкости	8
3 Технологический раздел	9
3.1 Средства реализации	9
3.2 Реализация алгоритмов	9
3.3 Тестирование	9
4 Исследовательская часть	10
4.1 Технические характеристики	10
4.2 Демонстрация работы программы	11
4.3 Параметризация муравьиного алгоритма	12
4.4 Временные характеристики	14
ЗАКЛЮЧЕНИЕ	16
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	17
ПРИЛОЖЕНИЕ А	18

ВВЕДЕНИЕ

Цель данной работы: исследование методов решения задачи коммивояжера полным перебором и с помощью муравьиного алгоритма.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- описать задачу коммивояжера;
- описать метод полного перебора и метод на основе муравьиного алгоритма;
- привести схемы описанных алгоритмов;
- реализовать данные алгоритмы;
- сравнить реализованные алгоритмы по времени выполнения;
- выполнить параметризацию муравьиного алгоритма.

Индивидуальный вариант для выполнения лабораторной работы:

- неориентированный граф;
- с элитными муравьями;
- Гамильтонов цикл;
- карта перемещения по Африке.

1 Аналитическая часть

В этом разделе будет описана задача коммивояжера и методы ее решения: метод полного перебора и на основе муравьиного алгоритма.

1.1 Задача коммивояжера

В этой задаче рассматривается n городов и матрица попарных расстояний между ними. Необходимо найти такую последовательность посещения городов, чтобы пройденное расстояние было как можно меньше. Каждый город должен быть посещен ровно один раз и коммивояжеру необходимо оказаться в городе, из которого он начал свое движение [1].

1.2 Методы решения задачи коммивояжера

Полный перебор. Суть этого решения заключается в переборе всех возможных вариантов замкнутых путей и в выборе кратчайшего из них.

Муравьиный алгоритм. Этот метод основан на принципах, описывающих поведение колонии муравьев. Они используют феромоны для общения друг с другом и поиска пути к пище. Если путь длинный, феромон испаряется, и последующая пара выбирает другой путь с большим количеством феромона, оставляя наибольшее его количество на кратчайшем пути [2].

Введем целевую функцию (1.1), которая описывает привлекательность ребра

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

где D_{ij} — расстояние от текущего пункта i до заданного пункта j .

Вероятность перехода из пункта i в пункт j определяется по формуле (1.2):

$$P_{i,j} = \frac{(\tau_{i,j}^\alpha)(\eta_{i,j}^\beta)}{\sum (\tau_{i,j}^\alpha)(\eta_{i,j}^\beta)}, \quad (1.2)$$

где:

- $\tau_{i,j}$ — количество феромонов на ребре от i до j ;
- $\eta_{i,j}$ — привлекательность пути от i до j ;
- α — параметр влияния расстояния;

— β — параметр влияния феромона.

В случае $\alpha = 0$ выбирается ближайший город и алгоритм становится «жадным», то есть выбираются только оптимальные или самые короткие расстояния.

Если $\beta = 0$, то работает лишь усиление феромонами, что влечет за собой сужение пространства поиска оптимального решения [2].

После происходит обновление феромона на пройденных путях по формуле (1.3), в случае, если p — коэффициент испарения феромона, N — количество феромонов, Q — некоторая константа порядка длины путей, L_k — длина пути муравья с номером k [2].

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

где

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{ребро посещено } k\text{-ым муравьем,} \\ 0, & \text{иначе.} \end{cases} \quad (1.4)$$

Так как вероятность (1.2) перехода в заданную точку не должна быть равна нулю, для этого нужно обеспечить неравенство $\tau_{ij}(t)$ нулю путем введения дополнительного минимально возможного значения феромона τ_{min} . В случае, если $\tau_{ij}(t+1)$ принимает значение, меньшее τ_{min} , откатывать феромон до этой величины.

Одной из модификаций муравьиного алгоритма является элитарная муравьиная система. При таком подходе искусственно вводятся «элитные» муравьи, усиливающие уровень феромонов, оптимального на данный момент маршрута.

Вывод

В этом разделе была рассмотрена задача коммивояжера и способы ее решения — полным перебором и муравьиным алгоритмом.

2 Конструкторская часть

В этом разделе будут рассмотрены схемы алгоритма полного перебора и муравьиного алгоритма.

2.1 Схемы алгоритмов

На рисунке 2.1 представлен алгоритм перебора всех возможных путей.

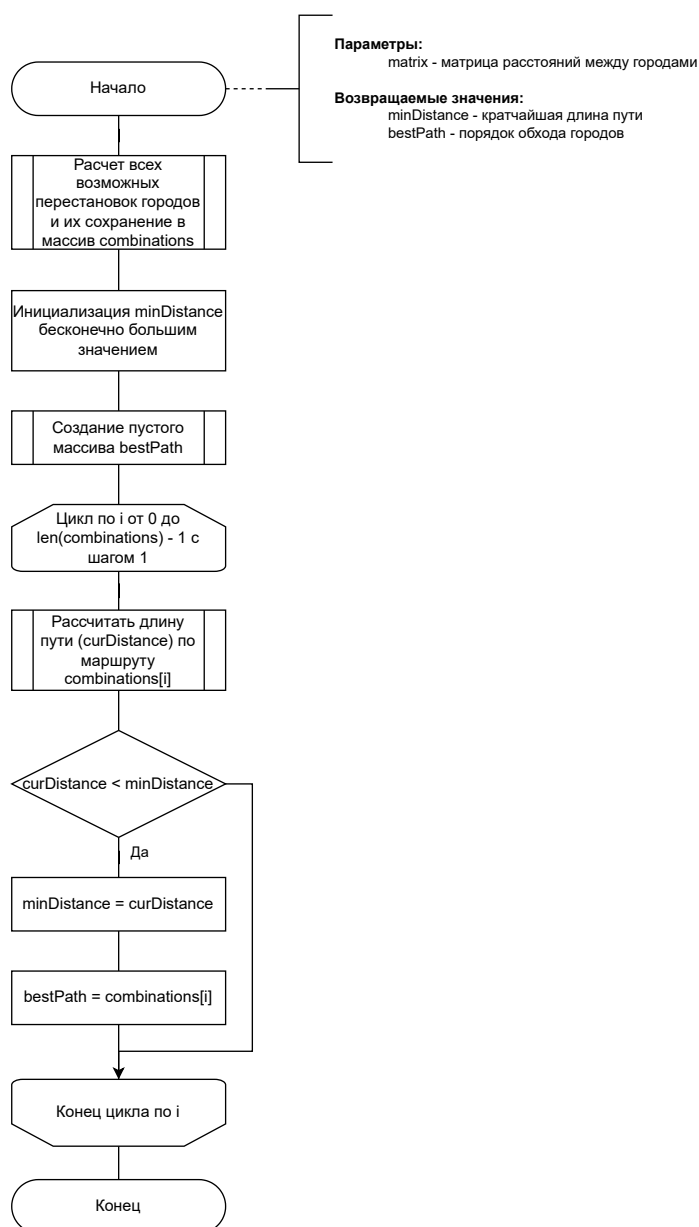


Рисунок 2.1 – Схема алгоритма полного перебора

На рисунке 2.2 представлен муравьиный алгоритм.

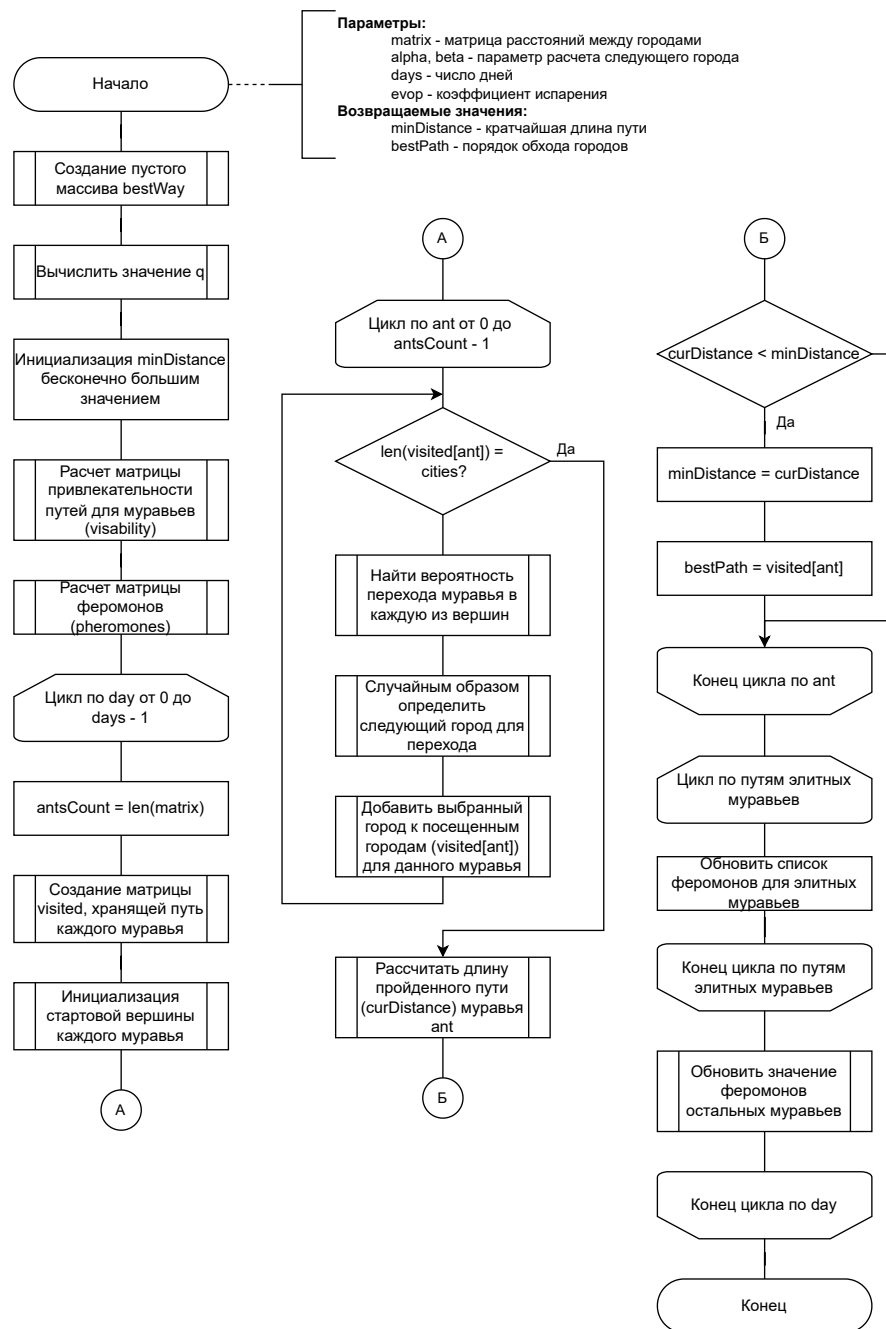


Рисунок 2.2 – Схема муравьиного алгоритма

2.2 Оценка трудоемкости

Задача коммивояжера считается NP - трудной. Сложность алгоритма полного перебора — $O(n!)$ [1].

Сложность муравьиного алгоритма — $O(t_{max} \cdot m \cdot n^2)$. Она зависит от времени жизни колонии, количества городов и количества муравьев в колонии [3].

Поскольку в разработанной реализации количество муравьев и городов одинаково, то трудоемкость муравьиного алгоритма равна $O(t_{max} \cdot n^3)$.

Вывод

В этом разделе были построены схемы описанных алгоритмов.

3 Технологический раздел

В этом разделе будут описаны средства реализации программы, листинги и функциональные тесты.

3.1 Средства реализации

Для реализации программного обеспечения был выбран язык Python [4].

3.2 Реализация алгоритмов

Листинги программ А.1 – А.2 приведены в приложении.

3.3 Тестирование

В таблице 3.1 приведены функциональные тесты программы. Все тесты были успешно пройдены.

Таблица 3.1 – Функциональные тесты

Матрица смежности	Полный перебор	Муравьиный алгоритм
$\begin{pmatrix} 0 & 2 & 6 & 5 \\ 2 & 0 & 7 & 7 \\ 6 & 7 & 0 & 11 \\ 5 & 7 & 11 & 0 \end{pmatrix}$	16, [2, 0, 1, 3]	16, [2, 0, 1, 3]
$\begin{pmatrix} 0 & 2 & 3 \\ 2 & 0 & 5 \\ 3 & 5 & 0 \end{pmatrix}$	5, [1, 0, 2]	5, [1, 0, 2]
$\begin{pmatrix} 0 & 9 \\ 9 & 0 \end{pmatrix}$	9, [0, 1]	9, [0, 1]

Вывод

В этом разделе были описаны средства реализации и представлены листинги реализованных алгоритмов и тесты.

4 Исследовательская часть

4.1 Технические характеристики

Технические характеристики устройства, на котором выполнялось тестирование:

- операционная система: Windows 10 Pro;
- память: 8 Гб;
- процессор: Intel(R) Core(TM) i5-8265U CPU @ 1.60 ГГц 1.80 ГГц;
- 12 логических ядра.

Тестирование проводилось на ноутбуке, который был подключен к сети питания. Во время проведения тестирования ноутбук был нагружен только встроенными приложениями окружения, самим окружением и системой тестирования.

4.2 Демонстрация работы программы

На рисунках 4.1 – 4.3 продемонстрирована работа программы.

```
Меню:
1. Полный перебор
2. Муравьиный алгоритм
3. Загрузить матрицу расстояний
4. Параметризация
5. Замерить время
6. Распечатать матрицу
7. Результат работы двух алгоритмов
8. Создать случайную матрицу
0. Выход
Выбор: 6

Доступные файлы: 2 штук
1. gen.csv
2. real.csv

Выберите файл: 2
0 62 58 47 40 80 82 91 21 41
62 0 62 96 61 94 73 0 56 27
58 62 0 71 15 97 9 63 66 33
47 96 71 0 71 18 94 60 76 16
40 61 15 71 0 1 84 2 57 47
80 94 97 18 1 0 64 0 16 80
82 73 9 94 84 64 0 71 15 61
91 0 63 60 2 0 71 0 71 96
21 56 66 76 57 16 15 71 0 62
41 27 33 16 47 80 61 96 62 0
```

Рисунок 4.1 – Демонстрация работы программы (меню и исходная матрица)

```
Выбор: 1

Доступные файлы: 2 штук
1. gen.csv
2. real.csv

Выберите файл: 2

Минимальная сумма пути = 151.0
Путь: [0, 3, 9, 1, 7, 5, 4, 2, 6, 8]
```

Рисунок 4.2 – Результат работы алгоритма полного перебора

```

Выбор: 2

Доступные файлы: 2 штук
1. gen.csv
2. real.csv

Выберите файл: 2

Введите коэффициент alpha: 0
Введите коэффициент evaporation: 0.5
Введите кол-во дней: 10
Введите количество элитных муравьев: 2
Введите коэффициент усиления феромонов элитных муравьев: 0.5

Минимальная сумма пути = 151.0
Путь: [4, 5, 7, 1, 9, 3, 0, 8, 6, 2]

```

Рисунок 4.3 – Результат работы муравьиного алгоритма

4.3 Параметризация муравьиного алгоритма

Параметризация была проведена на трех классах данных: 4.1, 4.2 и 4.3. Таблицы имеют различный разброс расстояний: 1000, 100, 10 соответственно. Алгоритм был запущен для набора значение $\alpha, eva \in (0, 1)$.

Результирующая таблица значений параметризации будет состоять из следующих столбцов:

- 1) α — параметр α при вычислении вероятности перехода в новый город;
- 2) eva — коэффициент испарения;
- 3) $days$ — количество дней жизни колонии;
- 4) $optim$ — результат решения полным перебором;
- 5) $delta$ — разность между решением полным перебором и решением муравьиного алгоритма.

Замеры проводились 10 раз и выбирался результат с максимальным $delta$ от результата перебора. Для полученных таблиц А.1–А.3 соответственно использовались матрицы расстояний (4.1–4.3).

$$K_1 = \begin{bmatrix} 0 & 2525 & 934 & 3096 & 6574 & 3486 & 1970 & 1328 & 2672 & 2547 \\ 2525 & 0 & 4315 & 4726 & 3236 & 699 & 5121 & 2290 & 6050 & 5950 \\ 934 & 4315 & 0 & 3569 & 5892 & 4410 & 1161 & 2084 & 1997 & 1831 \\ 3096 & 4726 & 3569 & 0 & 7728 & 4259 & 2888 & 3917 & 3239 & 3330 \\ 6574 & 3236 & 5892 & 7728 & 0 & 3804 & 7021 & 4099 & 7889 & 7717 \\ 3486 & 699 & 4410 & 4259 & 3804 & 0 & 5069 & 2539 & 5974 & 5899 \\ 1740 & 5121 & 1161 & 2888 & 7021 & 5069 & 0 & 3061 & 1010 & 831 \\ 1328 & 2290 & 2084 & 3917 & 4099 & 2539 & 3061 & 0 & 3987 & 3851 \\ 2672 & 6050 & 1997 & 3239 & 7889 & 5974 & 1010 & 3987 & 0 & 205 \\ 2547 & 5950 & 1831 & 3330 & 7717 & 5899 & 831 & 3851 & 205 & 0 \end{bmatrix} \quad (4.1)$$

$$K_2 = \begin{bmatrix} 0 & 50 & 72 & 36 & 34 & 89 & 85 & 8 & 22 & 11 \\ 50 & 0 & 33 & 23 & 78 & 61 & 16 & 1 & 92 & 31 \\ 72 & 33 & 0 & 100 & 51 & 9 & 66 & 58 & 36 & 23 \\ 36 & 23 & 100 & 0 & 41 & 43 & 45 & 51 & 31 & 67 \\ 34 & 78 & 51 & 41 & 0 & 33 & 22 & 38 & 32 & 63 \\ 89 & 61 & 9 & 43 & 33 & 0 & 46 & 71 & 9 & 32 \\ 85 & 16 & 66 & 45 & 22 & 46 & 0 & 41 & 51 & 78 \\ 8 & 1 & 58 & 51 & 38 & 71 & 41 & 0 & 100 & 23 \\ 22 & 92 & 36 & 31 & 32 & 9 & 51 & 100 & 0 & 33 \\ 11 & 31 & 23 & 67 & 63 & 32 & 78 & 23 & 33 & 0 \end{bmatrix} \quad (4.2)$$

$$K_3 = \begin{bmatrix} 0 & 1 & 9 & 0 & 9 & 4 & 2 & 10 & 7 & 10 \\ 1 & 0 & 9 & 10 & 9 & 1 & 9 & 1 & 9 & 8 \\ 9 & 9 & 0 & 4 & 8 & 7 & 4 & 6 & 8 & 5 \\ 0 & 10 & 4 & 0 & 9 & 0 & 1 & 7 & 3 & 2 \\ 9 & 9 & 8 & 9 & 0 & 8 & 9 & 8 & 7 & 5 \\ 5 & 1 & 7 & 0 & 8 & 0 & 1 & 4 & 8 & 2 \\ 2 & 9 & 4 & 1 & 9 & 1 & 0 & 9 & 8 & 9 \\ 10 & 1 & 6 & 7 & 8 & 4 & 9 & 0 & 4 & 10 \\ 7 & 9 & 8 & 3 & 7 & 8 & 8 & 4 & 0 & 9 \\ 10 & 8 & 5 & 2 & 4 & 2 & 9 & 10 & 9 & 0 \end{bmatrix} \quad (4.3)$$

4.4 Временные характеристики

Сравнение алгоритмов по времени выполнения производилось при изменении количества городов n от 2 до 10 с шагом 1. В результате замеров времени была получена таблица 4.1. Замеры проводились 10 раз, затем бралось их среднее арифметическое значение. По таблице 4.1 был построен график 4.4. Таблица 4.1 – Результаты измерений реализаций алгоритмов при изменении количества городов

n	Алгоритм полного перебора (с)	Муравьиный алгоритм (с)
2	0.000059	0.002753
3	0.000045	0.012209
4	0.000092	0.027333
5	0.000366	0.060201
6	0.001566	0.101122
7	0.011576	0.189127
8	0.099957	0.325559
9	1.164084	0.576964
10	13.032171	0.655146

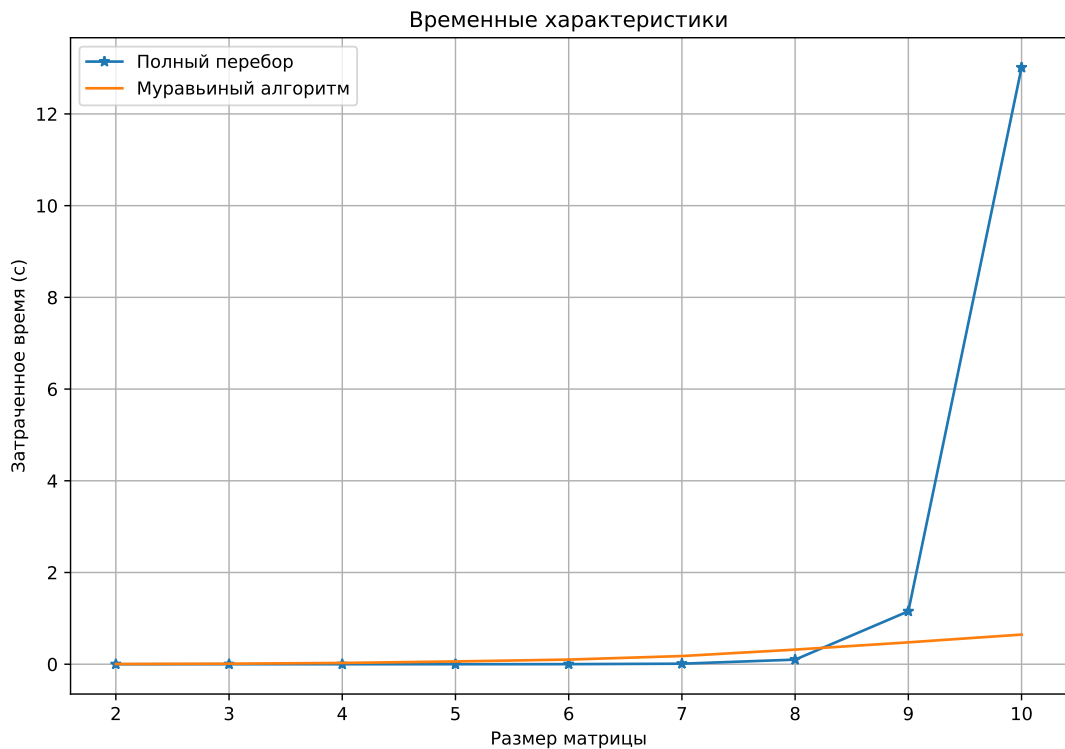


Рисунок 4.4 – Сравнение реализаций алгоритмов по времени работы при изменении размера матрицы смежности

Выводы

С увлечение числа дней жизни колонии *delta* уменьшается, вне зависимости от разброса расстояний. От значений α и *eva* зависимостей не обнаружено, следовательно, значения параметров должны устанавливаться для каждой конкретной задачи отдельно.

Из таблицы 4.1 выяснено: при малых размерах матриц реализация алгоритма полного перебора тратит меньше времени на получение результата.

При размерах матрицы больше 9 реализация алгоритма полного перебора работает в 1.45 раз больше, чем реализация муравьиного алгоритма. Таким образом, выбор решения задачи определяется размером матрицы расстояний между городами.

ЗАКЛЮЧЕНИЕ

Поставленная цель: исследование методов решения задачи коммивояжера двумя способами: полным перебором и с помощью муравьиного алгоритма, была достигнута.

Для поставленной цели были выполнены следующие задачи:

- описана задача коммивояжера;
- описаны методы решения задачи коммивояжера — метод полного перебора и метод на основе муравьиного алгоритма;
- приведены схемы описанных алгоритмов;
- реализованы данные алгоритмы;
- проведено сравнение по времени выполнения;
- выполнена параметризация муравьиного алгоритма.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Задача коммивояжера [Электронный ресурс]. — Режим доступа: <http://old.math.nsc.ru/LBRT/k5/OR-MMF/TSP.r.pdf> (дата обращения: 12.02.2025).
2. Штовба, С. Д. Муравьиные алгоритмы // Exponenta Pro. — 2003. — № 4. — С. 70–75.
3. Ульянов, М. В. Ресурсно-эффективные компьютерные алгоритмы. Разработка и анализ // М.: Наука-Физматлит. — 2007. — С. 201–207.
4. Лутц, Марк. Изучаем Python, том 1, 5-е изд. — Пер. с англ. — СПб.: ООО “Диалектика”, 2019. — 832 с.

ПРИЛОЖЕНИЕ А

Листинг А.1 – Реализация алгоритма полного перебора

```
1 def CombAlg(mtx, size):
2
3     places = np.arange(size)
4     placesCombinations = list()
5
6     for combination in it.permutations(places):
7         combArr = list(combination)
8         placesCombinations.append(combArr)
9
10    minLen = float("inf")
11
12    for i in range(len(placesCombinations)):
13        curLen = 0
14        for j in range(size):
15            startCity = placesCombinations[i][j - 1]
16            endCity = placesCombinations[i][j]
17            curLen += mtx[startCity][endCity]
18
19        if (curLen < minLen):
20            minLen = curLen
21            bestRoute = placesCombinations[i]
22
23    return minLen, bestRoutes
```

Листинг А.2 – Реализация муравьиного алгоритма

```
1 def antAlg(mtx, cities, alpha, beta, k_evaporation, days,
2 eliteAnts, eliteKoeff):
3     q = calculateQ(mtx, cities)
4     bestRoute = []
5     minLen = float("inf")
6     pheromones = calculatePheromones(cities)
7     visibility = calculateVisibility(mtx, cities)
8
9     elite_ant_paths = []
10    elite_paths_updated = False
11
12    for _ in range(days):
```

```

12     route = np.arange(cities)
13     visited = calculateVisitedPlaces(route, cities)
14
15     for ant in range(cities):
16         while (len(visited[ant]) != cities):
17             pk = findProbs(pheromones, visibility, visited,
18                           cities, ant, alpha, beta)
19             chosenPlace = chooseNextCity(pk)
20             visited[ant].append(chosenPlace - 1)
21
22             curLength = calculateLength(mtx, visited[ant])
23
24             if (curLength < minLen):
25                 minLen = curLength
26                 bestRoute = visited[ant]
27
28             if ant < eliteAnts and not elite_paths_updated:
29                 for elitePath in elite_ant_paths:
30                     for i in range(len(elitePath) - 1):
31                         pheromones[elitePath[i]][elitePath[i +
32                                     1]] *= (1 + eliteKoeff)
33                     elite_paths_updated = True
34
35     pheromones = updatePheromone(mtx, cities, visited,
36                                  pheromones, q, k_evaporation, elite_ant_paths,
37                                  eliteKoeff)
38
39     return minLen, bestRoute

```

Таблица А.1 – Таблица параметров для муравьиного алгоритма с разбросом длин путей 1000

α	<i>eva</i>	<i>days</i>	<i>optim</i>	<i>delta</i>
0.2	0.2	1	19991	8619
0.2	0.2	5	19991	4417
0.2	0.2	10	19991	3986
0.2	0.2	50	19991	1921
0.2	0.2	100	19991	1296
0.2	0.3	1	19991	10557
0.2	0.3	5	19991	5179

Таблица А.1 – Таблица параметров для муравьиного алгоритма с разбросом длин путей 1000

α	eva	$days$	$optim$	$delta$
0.2	0.3	10	19991	2771
0.2	0.3	50	19991	2394
0.2	0.3	100	19991	1423
0.2	0.5	1	19991	7648
0.2	0.5	5	19991	4063
0.2	0.5	10	19991	3794
0.2	0.5	50	19991	2428
0.2	0.5	100	19991	1296
0.2	0.7	1	19991	8062
0.2	0.7	5	19991	3946
0.2	0.7	10	19991	3876
0.2	0.7	50	19991	2426
0.2	0.7	100	19991	1035
0.2	0.9	1	19991	7374
0.2	0.9	5	19991	4832
0.2	0.9	10	19991	3959
0.2	0.9	50	19991	1835
0.2	0.9	100	19991	2202
0.3	0.2	1	19991	9208
0.3	0.2	5	19991	5617
0.3	0.2	10	19991	3929
0.3	0.2	50	19991	2036
0.3	0.2	100	19991	1565
0.3	0.3	1	19991	9898
0.3	0.3	5	19991	5509
0.3	0.3	10	19991	5472
0.3	0.3	50	19991	2554
0.3	0.3	100	19991	1781
0.3	0.5	1	19991	8456
0.3	0.5	5	19991	3821
0.3	0.5	10	19991	4617

Таблица А.1 – Таблица параметров для муравьиного алгоритма с разбросом длин путей 1000

α	eva	$days$	$optim$	$delta$
0.3	0.5	50	19991	2767
0.3	0.5	100	19991	2394
0.3	0.7	1	19991	9202
0.3	0.7	5	19991	5547
0.3	0.7	10	19991	4436
0.3	0.7	50	19991	2497
0.3	0.7	100	19991	1565
0.3	0.9	1	19991	10077
0.3	0.9	5	19991	3842
0.3	0.9	10	19991	3939
0.3	0.9	50	19991	2798
0.3	0.9	100	19991	2036
0.5	0.2	1	19991	10583
0.5	0.2	5	19991	6745
0.5	0.2	10	19991	5515
0.5	0.2	50	19991	3593
0.5	0.2	100	19991	3307
0.5	0.3	1	19991	9208
0.5	0.3	5	19991	4635
0.5	0.3	10	19991	4484
0.5	0.3	50	19991	3491
0.5	0.3	100	19991	2301
0.5	0.5	1	19991	10367
0.5	0.5	5	19991	5563
0.5	0.5	10	19991	4292
0.5	0.5	50	19991	3626
0.5	0.5	100	19991	2407
0.5	0.7	1	19991	9829
0.5	0.7	5	19991	7037
0.5	0.7	10	19991	4419
0.5	0.7	50	19991	3900

Таблица А.1 – Таблица параметров для муравьиного алгоритма с разбросом длин путей 1000

α	eva	$days$	$optim$	$delta$
0.5	0.7	100	19991	2439
0.5	0.9	1	19991	10099
0.5	0.9	5	19991	5605
0.5	0.9	10	19991	4359
0.5	0.9	50	19991	3794
0.5	0.9	100	19991	2068
0.7	0.2	1	19991	11161
0.7	0.2	5	19991	6606
0.7	0.2	10	19991	5070
0.7	0.2	50	19991	5265
0.7	0.2	100	19991	3491
0.7	0.3	1	19991	9553
0.7	0.3	5	19991	7171
0.7	0.3	10	19991	5374
0.7	0.3	50	19991	4635
0.7	0.3	100	19991	3724
0.7	0.5	1	19991	12900
0.7	0.5	5	19991	8568
0.7	0.5	10	19991	6465
0.7	0.5	50	19991	4681
0.7	0.5	100	19991	3533
0.7	0.7	1	19991	10023
0.7	0.7	5	19991	7587
0.7	0.7	10	19991	6528
0.7	0.7	50	19991	4816
0.7	0.7	100	19991	3610
0.7	0.9	1	19991	12985
0.7	0.9	5	19991	6649
0.7	0.9	10	19991	7799
0.7	0.9	50	19991	4187
0.7	0.9	100	19991	3549

Таблица А.1 – Таблица параметров для муравьиного алгоритма с разбросом длин путей 1000

α	eva	$days$	$optim$	$delta$
0.9	0.2	1	19991	11798
0.9	0.2	5	19991	8215
0.9	0.2	10	19991	7895
0.9	0.2	50	19991	5769
0.9	0.2	100	19991	3865
0.9	0.3	1	19991	11483
0.9	0.3	5	19991	7317
0.9	0.3	10	19991	8542
0.9	0.3	50	19991	4454
0.9	0.3	100	19991	3641
0.9	0.5	1	19991	11522
0.9	0.5	5	19991	9351
0.9	0.5	10	19991	7813
0.9	0.5	50	19991	5867
0.9	0.5	100	19991	4122
0.9	0.7	1	19991	12543
0.9	0.7	5	19991	9556
0.9	0.7	10	19991	7012
0.9	0.7	50	19991	5861
0.9	0.7	100	19991	4714
0.9	0.9	1	19991	14969
0.9	0.9	5	19991	9260
0.9	0.9	10	19991	6886
0.9	0.9	50	19991	6045
0.9	0.9	100	19991	4171

Таблица А.2 – Таблица параметров для муравьиного алгоритма с разбросом длин путей 100

α	eva	$days$	$optim$	$delta$
0.2	0.2	1	171	121

Таблица А.2 – Таблица параметров для муравьиного алгоритма с разбросом длин путей 100

α	eva	$days$	$optim$	$delta$
0.2	0.2	5	171	65
0.2	0.2	10	171	52
0.2	0.2	50	171	39
0.2	0.2	100	171	24
0.2	0.3	1	171	129
0.2	0.3	5	171	76
0.2	0.3	10	171	52
0.2	0.3	50	171	41
0.2	0.3	100	171	24
0.2	0.5	1	171	121
0.2	0.5	5	171	74
0.2	0.5	10	171	47
0.2	0.5	50	171	25
0.2	0.5	100	171	0
0.2	0.7	1	171	112
0.2	0.7	5	171	65
0.2	0.7	10	171	54
0.2	0.7	50	171	33
0.2	0.7	100	171	0
0.2	0.9	1	171	120
0.2	0.9	5	171	60
0.2	0.9	10	171	68
0.2	0.9	50	171	27
0.2	0.9	100	171	25
0.3	0.2	1	171	155
0.3	0.2	5	171	68
0.3	0.2	10	171	57
0.3	0.2	50	171	36
0.3	0.2	100	171	38
0.3	0.3	1	171	136
0.3	0.3	5	171	72

Таблица А.2 – Таблица параметров для муравьиного алгоритма с разбросом длин путей 100

α	eva	$days$	$optim$	$delta$
0.3	0.3	10	171	54
0.3	0.3	50	171	53
0.3	0.3	100	171	41
0.3	0.5	1	171	126
0.3	0.5	5	171	78
0.3	0.5	10	171	57
0.3	0.5	50	171	45
0.3	0.5	100	171	38
0.3	0.7	1	171	150
0.3	0.7	5	171	74
0.3	0.7	10	171	49
0.3	0.7	50	171	59
0.3	0.7	100	171	27
0.3	0.9	1	171	142
0.3	0.9	5	171	81
0.3	0.9	10	171	66
0.3	0.9	50	171	53
0.3	0.9	100	171	27
0.5	0.2	1	171	174
0.5	0.2	5	171	120
0.5	0.2	10	171	98
0.5	0.2	50	171	80
0.5	0.2	100	171	52
0.5	0.3	1	171	145
0.5	0.3	5	171	112
0.5	0.3	10	171	91
0.5	0.3	50	171	47
0.5	0.3	100	171	50
0.5	0.5	1	171	128
0.5	0.5	5	171	101
0.5	0.5	10	171	87

Таблица А.2 – Таблица параметров для муравьиного алгоритма с разбросом длин путей 100

α	eva	$days$	$optim$	$delta$
0.5	0.5	50	171	59
0.5	0.5	100	171	38
0.5	0.7	1	171	150
0.5	0.7	5	171	124
0.5	0.7	10	171	90
0.5	0.7	50	171	58
0.5	0.7	100	171	40
0.5	0.9	1	171	156
0.5	0.9	5	171	91
0.5	0.9	10	171	82
0.5	0.9	50	171	61
0.5	0.9	100	171	41
0.7	0.2	1	171	169
0.7	0.2	5	171	124
0.7	0.2	10	171	91
0.7	0.2	50	171	73
0.7	0.2	100	171	50
0.7	0.3	1	171	158
0.7	0.3	5	171	121
0.7	0.3	10	171	98
0.7	0.3	50	171	82
0.7	0.3	100	171	65
0.7	0.5	1	171	198
0.7	0.5	5	171	136
0.7	0.5	10	171	99
0.7	0.5	50	171	80
0.7	0.5	100	171	59
0.7	0.7	1	171	175
0.7	0.7	5	171	109
0.7	0.7	10	171	101
0.7	0.7	50	171	89

Таблица А.2 – Таблица параметров для муравьиного алгоритма с разбросом длин путей 100

α	eva	$days$	$optim$	$delta$
0.7	0.7	100	171	52
0.7	0.9	1	171	202
0.7	0.9	5	171	126
0.7	0.9	10	171	93
0.7	0.9	50	171	81
0.7	0.9	100	171	41
0.9	0.2	1	171	233
0.9	0.2	5	171	163
0.9	0.2	10	171	130
0.9	0.2	50	171	90
0.9	0.2	100	171	68
0.9	0.3	1	171	189
0.9	0.3	5	171	127
0.9	0.3	10	171	148
0.9	0.3	50	171	100
0.9	0.3	100	171	72
0.9	0.5	1	171	184
0.9	0.5	5	171	120
0.9	0.5	10	171	114
0.9	0.5	50	171	101
0.9	0.5	100	171	65
0.9	0.7	1	171	199
0.9	0.7	5	171	142
0.9	0.7	10	171	107
0.9	0.7	50	171	84
0.9	0.7	100	171	79
0.9	0.9	1	171	243
0.9	0.9	5	171	166
0.9	0.9	10	171	135
0.9	0.9	50	171	105
0.9	0.9	100	171	77

Таблица А.3 – Таблица параметров для муравьиного алгоритма с разбросом длин путей 10

α	eva	$days$	$optim$	$delta$
0.2	0.2	1	27	11
0.2	0.2	5	27	3
0.2	0.2	10	27	1
0.2	0.2	50	27	1
0.2	0.2	100	27	0
0.2	0.3	1	27	11
0.2	0.3	5	27	6
0.2	0.3	10	27	4
0.2	0.3	50	27	1
0.2	0.3	100	27	0
0.2	0.5	1	27	12
0.2	0.5	5	27	9
0.2	0.5	10	27	3
0.2	0.5	50	27	1
0.2	0.5	100	27	0
0.2	0.7	1	27	13
0.2	0.7	5	27	6
0.2	0.7	10	27	1
0.2	0.7	50	27	1
0.2	0.7	100	27	0
0.2	0.9	1	27	12
0.2	0.9	5	27	4
0.2	0.9	10	27	2
0.2	0.9	50	27	1
0.2	0.9	100	27	0
0.3	0.2	1	27	12
0.3	0.2	5	27	5
0.3	0.2	10	27	6
0.3	0.2	50	27	2
0.3	0.2	100	27	0
0.3	0.3	1	27	12

Таблица А.3 – Таблица параметров для муравьиного алгоритма с разбросом длин путей 10

α	eva	$days$	$optim$	$delta$
0.3	0.3	5	27	8
0.3	0.3	10	27	4
0.3	0.3	50	27	1
0.3	0.3	100	27	0
0.3	0.5	1	27	9
0.3	0.5	5	27	4
0.3	0.5	10	27	6
0.3	0.5	50	27	1
0.3	0.5	100	27	0
0.3	0.7	1	27	11
0.3	0.7	5	27	7
0.3	0.7	10	27	6
0.3	0.7	50	27	1
0.3	0.7	100	27	0
0.3	0.9	1	27	15
0.3	0.9	5	27	6
0.3	0.9	10	27	4
0.3	0.9	50	27	1
0.3	0.9	100	27	1
0.5	0.2	1	27	16
0.5	0.2	5	27	8
0.5	0.2	10	27	5
0.5	0.2	50	27	2
0.5	0.2	100	27	1
0.5	0.3	1	27	12
0.5	0.3	5	27	5
0.5	0.3	10	27	5
0.5	0.3	50	27	2
0.5	0.3	100	27	1
0.5	0.5	1	27	13
0.5	0.5	5	27	8

Таблица А.3 – Таблица параметров для муравьиного алгоритма с разбросом длин путей 10

α	eva	$days$	$optim$	$delta$
0.5	0.5	10	27	6
0.5	0.5	50	27	2
0.5	0.5	100	27	1
0.5	0.7	1	27	14
0.5	0.7	5	27	8
0.5	0.7	10	27	6
0.5	0.7	50	27	2
0.5	0.7	100	27	0
0.5	0.9	1	27	12
0.5	0.9	5	27	9
0.5	0.9	10	27	7
0.5	0.9	50	27	2
0.5	0.9	100	27	0
0.7	0.2	1	27	16
0.7	0.2	5	27	8
0.7	0.2	10	27	8
0.7	0.2	50	27	4
0.7	0.2	100	27	2
0.7	0.3	1	27	11
0.7	0.3	5	27	6
0.7	0.3	10	27	7
0.7	0.3	50	27	3
0.7	0.3	100	27	1
0.7	0.5	1	27	12
0.7	0.5	5	27	8
0.7	0.5	10	27	8
0.7	0.5	50	27	3
0.7	0.5	100	27	1
0.7	0.7	1	27	13
0.7	0.7	5	27	9
0.7	0.7	10	27	6

Таблица А.3 – Таблица параметров для муравьиного алгоритма с разбросом длин путей 10

α	eva	$days$	$optim$	$delta$
0.7	0.7	50	27	5
0.7	0.7	100	27	2
0.7	0.9	1	27	15
0.7	0.9	5	27	8
0.7	0.9	10	27	7
0.7	0.9	50	27	3
0.7	0.9	100	27	1
0.9	0.2	1	27	12
0.9	0.2	5	27	9
0.9	0.2	10	27	7
0.9	0.2	50	27	6
0.9	0.2	100	27	3
0.9	0.3	1	27	14
0.9	0.3	5	27	9
0.9	0.3	10	27	7
0.9	0.3	50	27	5
0.9	0.3	100	27	3
0.9	0.5	1	27	15
0.9	0.5	5	27	9
0.9	0.5	10	27	8
0.9	0.5	50	27	4
0.9	0.5	100	27	2
0.9	0.7	1	27	16
0.9	0.7	5	27	9
0.9	0.7	10	27	6
0.9	0.7	50	27	3
0.9	0.7	100	27	1
0.9	0.9	1	27	16
0.9	0.9	5	27	8
0.9	0.9	10	27	8
0.9	0.9	50	27	4

Таблица А.3 – Таблица параметров для муравьиного алгоритма с разбросом длин путей 10

α	eva	$days$	$optim$	$delta$
0.9	0.9	100	27	3