

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 _	Методы решения задачи коммивояжёра	
Студе	нт Талышева О.Н.	
Групп	иа <u>ИУ7-55Б</u>	
Препо	одаватели Волкова Л.Л., Строганов Ю.В.	

## СОДЕРЖАНИЕ

BI	ЗЕДЕ	ЕНИЕ	3
1	Ана	литическая часть	4
	1.1	Задача коммивояжёра	4
	1.2	Методы решения задачи коммивояжёра	5
		1.2.1 Метод полного перебора	5
		1.2.2 Муравьиный алгоритм	F
	1.3	Выводы	7
2	Кон	структорская часть	8
	2.1	Описания алгоритмов	8
3	Text	нологическая часть	12
	3.1	Требования к программному обеспечению	12
	3.2	Средства реализации	12
	3.3	Реализации алгоритмов	12
	3.4	Тесты	14
4	Исс.	ледовательская часть	16
	4.1	Пример работы программы	16
	4.2	Результаты параметризации	17
		4.2.1 Класс данных №1	18
		4.2.2 Класс данных №2	19
		4.2.3 Класс данных №3	21
3.4	АКЛІ	ОЧЕНИЕ	25
CI	тисс	ОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	26

#### ВВЕДЕНИЕ

Цель работы: исследовать два подхода к решению задачи коммивояжёра: метод полного перебора и муравьиный алгоритм.

Для достижения этой цели были поставлены следующие задачи:

- 1) рассмотреть метод полного перебора;
- 2) рассмотреть муравьиный алгоритм;
- 3) реализовать оба алгоритма решения задачи коммивояжёра на выбранном языке программирования;
- 4) провести параметризацию муравьиного алгоритма по четырём параметрам и оценить качество решений для серии графов.

#### 1 Аналитическая часть

#### 1.1 Задача коммивояжёра

Задача коммивояжёра — одна из самых известных оптимизационных задач. Её цель заключается в нахождении самого выгодного маршрута (кратчайшего, самого быстрого, наиболее дешёвого), проходящего через все заданные точки (пункты, города) по одному разу.

Условия задачи должны включать критерий выгодности маршрута (например, минимальная длина, время, стоимость или их комбинация), а также исходные данные в виде матрицы затрат (расстояний, стоимости, времени и прочие) при перемещении между пунктами.

Особенность задачи заключается в том, что она просто формулируется, и хорошие приближённые решения можно найти относительно легко. Однако поиск действительно оптимального маршрута для большого набора данных требует значительных вычислительных ресурсов.

#### Математическая модель задачи

Для решения задачи коммивояжёра её можно представить как математическую модель. Исходные условия записываются в формате:

- 1) матрицы таблицы, где строки соответствуют городам отправления, столбцы городам прибытия, а элементы матрицы представляют расстояния, время или стоимость перемещения;
- 2) графа схемы, состоящей из вершин (города) и рёбер (пути между городами), где длина рёбер соответствует расстояниям.

#### Классификация задачи коммивояжёра

По симметричности рёбер графа:

- 1) симметричная задача (Symmetric TSP): длина пути между любыми двумя городами одинакова в обоих направлениях (неориентированный граф);
- 2) асимметричная задача (Asymmetric TSP): длина пути между городами может различаться в зависимости от направления (ориентированный граф).

По замкнутости маршрута:

- 1) замкнутая задача: поиск кратчайшего пути через все города с возвратом в исходную точку;
- 2) незамкнутая задача: поиск кратчайшего пути через все города без обязательного возврата в исходную точку. [2]

#### 1.2 Методы решения задачи коммивояжёра

Существует множество методов, различающихся инструментарием, точностью и вычислительной сложностью. В этом разделе будут подробнее рассмотрены метод полного перебора и муравьиный алгоритм.

#### 1.2.1 Метод полного перебора

Метод полного перебора заключается в построении всех возможных маршрутов для заданного множества городов. Для каждого маршрута вычисляется сумма весов рёбер, составляющих путь, и выбирается маршрут с минимальной общей длиной.

Алгоритм метода полного перебора:

- 1) построить все возможные перестановки городов;
- 2) для каждой перестановки вычислить длину маршрута как сумму весов соответствующих рёбер;
- 3) выбрать маршрут с минимальной длиной.

Преимущества:

- 1) гарантированное нахождение оптимального решения;
- 2) простота реализации.

Недостатки:

- 1) высокая вычислительная сложность O(n!), где n количество городов;
- 2) невозможность применения для больших графов из-за экспоненциального роста числа перестановок.

<u>Пример:</u> Приведён граф с четырьмя городами. Матрица весов рёбер задана следующим образом:

Все возможные маршруты:  $(1 \to 2 \to 3 \to 4 \to 1)$ ,  $(1 \to 3 \to 2 \to 4 \to 1)$  и другие. Для каждого маршрута вычисляем сумму весов рёбер и выбираем минимальную.

#### 1.2.2 Муравьиный алгоритм

Муравьиный алгоритм основан на моделировании поведения реальной муравьиной колонии. Этот метод использует взаимодействие между муравьями через химическое вещество – феромон, который откладывается на пройденных маршрутах. Феромон позволяет учитывать опыт других муравьёв при выборе следующего пути.

Основные элементы алгоритма:

- 1) феромон: определяет предпочтительность выбора маршрута, увеличиваясь на коротких путях;
- 2) видимость: локальная информация, обратная расстоянию между городами  $\eta_{ij} = 1/d_{ij}$ ;
- 3) память: список уже посещённых городов, предотвращающий повторное посещение

#### Этапы алгоритма:

- 1) инициализация параметров алгоритма: количество муравьёв, начальные значения феромона, параметры  $\alpha$  (влияние феромона) и  $\beta$  (влияние видимости);
- 2) для каждого муравья на текущей итерации выполняется выбор следующего города на основе вероятности перехода:

$$P_{ij}^k = rac{( au_{ij})^lpha (oldsymbol{\eta}_{ij})^eta}{\sum_{l \in ext{ iny IIIIIII}} ( au_{il})^lpha (oldsymbol{\eta}_{il})^eta},$$

где  $au_{ij}$  – уровень феромона на ребре,  $au_{ij}$  – видимость, lpha и eta – параметры алгоритма;

3) после завершения всех маршрутов обновляются значения феромона по формуле:

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

где  $\rho$  – коэффициент испарения, а  $\Delta \tau_{ij}$  рассчитывается по формуле:

$$\Delta au_{ij} = \sum_{k=1}^m rac{\mathcal{Q}}{L_k},$$

где  $L_k$  – длина маршрута муравья  $k,\ Q$  – регулируемый параметр;

4) проверяется условие остановки (например, достижение заданного числа итераций), и в случае выполнения выводится лучший маршрут.

#### Преимущества:

- 1) эффективность для больших графов;
- 2) гибкость за счёт параметров  $\alpha$ ,  $\beta$  и  $\rho$ .

#### Недостатки:

- 1) возможность попадания в локальные минимумы;
- 2) сложность настройки параметров. [1]

#### 1.3 Выводы

В данной аналитической части были рассмотрены два подхода к решению задачи коммивояжёра: метод полного перебора и муравьиный алгоритм.

Метод полного перебора, несмотря на его вычислительную сложность, предоставляет эталонное решение задачи, но применим только для небольших графов.

Муравьиный алгоритм, напротив, обеспечивает высокую производительность и подходит для задач с большим числом городов, хотя требует тонкой настройки параметров для достижения оптимальных результатов.

## 2 Конструкторская часть

#### 2.1 Описания алгоритмов

На основании теоретических измышлений были разработаны алгоритмы, реализующие решение задачи коммивояжёра полным перебором и муравьиным алгоритмом. Вариант: ориентированный граф, без элитных муравьёв, карта городов России XVI века. Раз в 60 суток наступает смена летнего сезона на зимний или наоборот. Зимой можно ходить по рекам в обе стороны за равную цену, летом по течению в 2 раза быстрее, против — в 4 раза медленнее. Незамкнутый маршрут. Блок-схемы этих алгоритмов приведены на рисунках 1 (полный перебор), 2 (функция учёта рек и сезона) и 3 (муравьиный алгоритм).

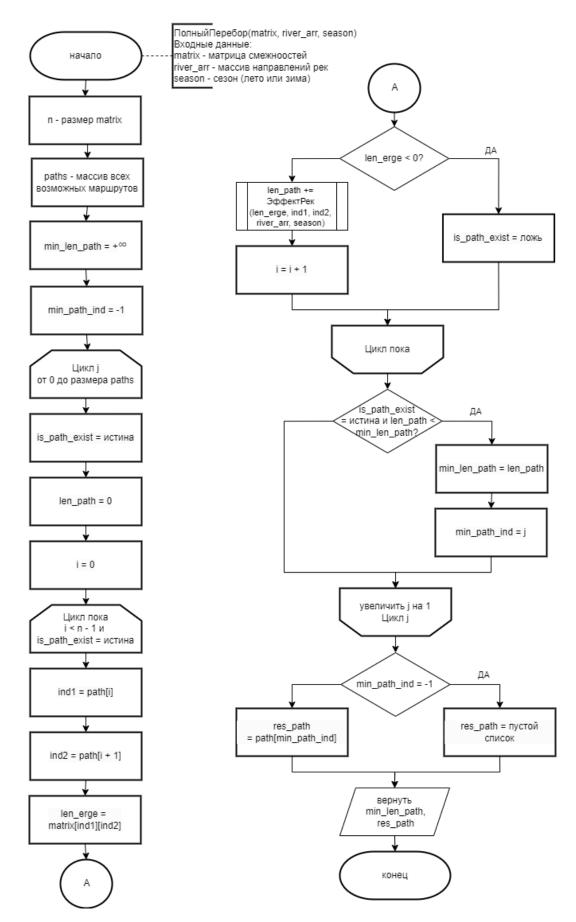


Рисунок 1 – Блок-схема алгоритма полного перебора.

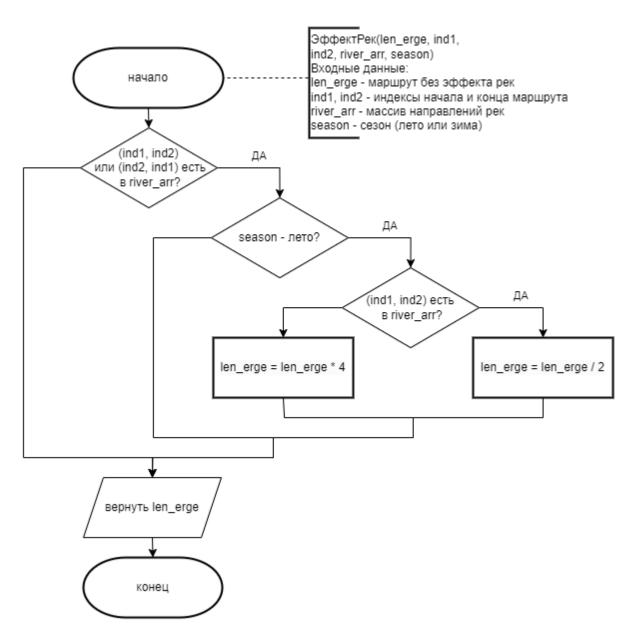


Рисунок 2 – Блок-схема функции учёта рек и сезона при решении задачи коммивояжёра.

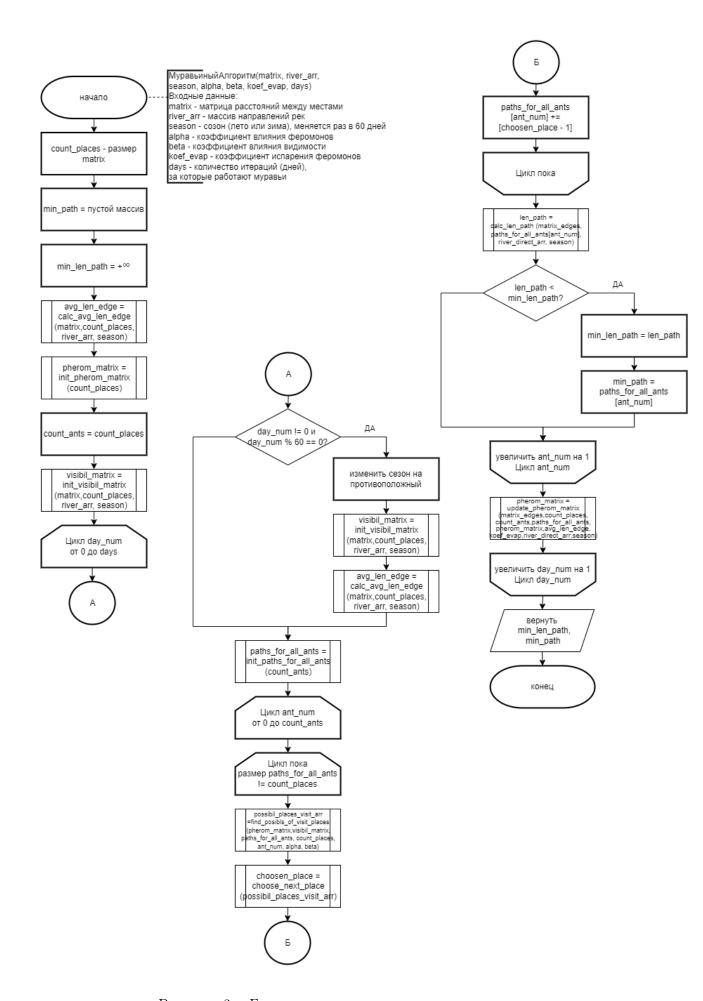


Рисунок 3 – Блок-схема муравьиного алгоритма.

#### 3 Технологическая часть

#### 3.1 Требования к программному обеспечению

#### Входные данные:

Программа принимает файл, содержащий матрицу расстояний между городами, где значение -1 означает отсутствие пути. Также задается список направлений рек и текущий сезон, влияющий на затраты на прохождение рек (зима или лето). Для решения задачи муравьиным алгоритмом дополнительно вводятся следующие параметры:

- 1) alpha параметр влияния длины пути;
- 2) beta параметр влияния феромона;
- 3) koef\_evap коэффициент испарения феромона;
- 4) days количество дней гуляния муравьёв.

#### Выходные данные:

На выходе программа предоставляет кратчайший незамкнутый маршрут обхода всех городов и его длину, вычисленные алгоритмом полного перебора или муравьиным алгоритмом.

#### 3.2 Средства реализации

В связи с опытом реализации подобных задач на Python, программа была реализована на этом языке программирования.

#### 3.3 Реализации алгоритмов

Ниже приведена реализация алгоритма коммивояжёра полным перебором (листинг 1) и муравьиным алгоритмом (листинг 2) на Python.

```
def Algo_full_search_paths(matrix_paths, river_direct_arr, season):
    n = len(matrix_paths)
    varios_paths_id_arr = [i for i in range(n)]
    all_combinations_paths = list()

for path in it.permutations(varios_paths_id_arr):
    all_combinations_paths.append(list(path))

min_len_path = float("+inf")
min_ind_path = -1

for j in range(len(all_combinations_paths)):
    is_path_exist = True
```

```
path = all_combinations_paths[j]
15
           len_path = 0
           i = 0
           while is_path_exist and i < (n - 1):</pre>
               ind_start_city = path[i]
               ind_finish_city = path[i + 1]
20
               len_erge = matrix_paths[ind_start_city][ind_finish_city]
21
               if len_erge < 0:</pre>
                   is_path_exist = False
23
               else:
24
                    len_path += effect_rivers_seasons(len_erge, ind_start_city
                       , ind_finish_city, river_direct_arr, season)
                   i += 1
          if is_path_exist and len_path < min_len_path:</pre>
28
               min_len_path = len_path
29
               min_ind_path = j
3.0
31
      if min_ind_path == -1:
32
          res_path = list()
34
      else:
           res_path = all_combinations_paths[min_ind_path]
36
      return min_len_path, res_path
```

Листинг 1 – реализации алгоритма коммивояжёра полным перебором

```
def Algo_ant_search_paths(matrix_edges, river_direct_arr, season, alpha,
     beta, koef_evap, days):
      count_places = len(matrix_edges)
     min_path = list()
     min_len_path = float("inf")
     avg_len_edge = calc_avg_len_edge(matrix_edges, count_places,
         river_direct_arr, season)
     pherom_matrix = init_pherom_matrix(count_places)
      count_ants = count_places
      visibil_matrix = init_visibil_matrix(matrix_edges, count_places,
         river_direct_arr, season)
     for day_num in range(days):
10
          if day_num != 0 and day_num % CHANGE_SEASON == 0:
              season = (season + 1) \% 2
              visibil_matrix = init_visibil_matrix(matrix_edges,
                 count_places, river_direct_arr, season)
              avg_len_edge = calc_avg_len_edge(matrix_edges, count_places,
                 river_direct_arr, season)
```

```
paths_for_all_ants = init_paths_for_all_ants(count_ants)
16
17
          for ant_num in range(count_ants):
18
              flag_deadlock = False
19
              while len(paths_for_all_ants[ant_num]) != count_places:
                   possibil_places_visit_arr = find_posibls_of_visit_places(
21
                      pherom_matrix, visibil_matrix, paths_for_all_ants,
                      count_places, ant_num, alpha, beta)
                   if all(value == 0 for value in possibil_places_visit_arr):
                       flag_deadlock = True
23
                       break
24
                   chosen_place = choose_next_place(possibil_places_visit_arr
                  paths_for_all_ants[ant_num].append(chosen_place - 1)
26
              if flag_deadlock:
28
                   continue
              len_path = calc_len_path(matrix_edges, paths_for_all_ants[
                  ant_num], river_direct_arr, season)
              if len_path < min_len_path:</pre>
                  min_len_path = len_path
33
                   min_path = paths_for_all_ants[ant_num]
35
          pherom_matrix = update_pherom_matrix(matrix_edges, count_places,
36
              count_ants, paths_for_all_ants, pherom_matrix, avg_len_edge,
             koef_evap, river_direct_arr, season)
37
      return min_len_path, min_path
```

Листинг 2 – реализации алгоритма коммивояжёра муравьиным алгоритмом

#### 3.4 Тесты

Для тестирования муравьиного алгоритма (MA) и алгоритма полного перебора (АПП) были составлены таблицы с входными данными, ожидаемыми результатами и успехом выполнения каждого алгоритма. Входные данные включают матрицу расстояний между точками, направления рек и сезон (0 – лето, 1 – зима).

Таблица 1 – Таблица тестов для алгоритмов поиска кратчайшего пути

Входные данные	Ожидаемый результат	Прохождение АПП	Прохождение МА
0 3 -1 7 3 0 5 6 10 -1 0 5 -1 -1 2 0 - 0 2 1 3 - 0	$\Pi  ext{уть} = [0,1,3,2]$ Длина $= 8.0$	Успех	Успех
0 -1 920 600 500 850 0 500 700 170 -1 500 0 -1 800 600 700 50 0 450 600 100 800 -1 0 — 0 3 — 0	$\Pi$ уть $=[0,3,2,1,4]$ Длина $=1020.0$	Успех	Успех

В ходе проведённого тестирования ошибок в алгоритмах не выявлено.

## 4 Исследовательская часть

## 4.1 Пример работы программы

Работа разработанной программы на тестовых входных данных для решения задачи коммивояжёра полным перебором и муравьиным алгоритмом приведена на рисунке 4.

```
data > = big_data_test.txt
 1
      0 -1 920 600 500
 2 850 0 500 700 170
      -1 500 0 -1 800
    600 700 50 0 450
 5
    600 100 800 -1 0
      0 3
 7
 8
      - - -
PROBLEMS
         OUTPUT DEBUG CONSOLE
                                    TERMINAL
                                               PORTS
Меню
1 - нахождение пути полным перебором;
2 - нахождение пути Муравьиным алгоритмом;
3 - нахождение пути обоими алгоритмами;
4 - параметризация
0. Выход из программы
Выборите пункт меню: 3
Выберите файл из списка:
1 - ancient Rus 1.txt
2 - ancient Rus 2.txt
3 - ancient Rus 3.txt
4 - big data test.txt
5 - little data test.txt
Введите номер файла: 4
Введите параметры для Муравьиного алгоритма.
Введите коэффициент влияния длины пути: 0.5
Введите коэффициент испарения феромона: 0.5
Введите количество дней: 20
Муравьиный алгоритм: кратчайший путь = [0, 3, 2, 1, 4] длиной = 1020.0.
Полный перебор: кратчайший путь = [0, 3, 2, 1, 4] длиной = 1020.0.
```

Рисунок 4 – Пример работы программы.

#### 4.2 Результаты параметризации

Для определения комбинации параметров муравьиного алгоритма, которые позволяют получать ответ, наиболее близкий к результату полного перебора для выбранного класса данных была выполнена автоматическая параметризация на трёх классах данных. Для каждого класса была составлена таблица с входными параметрами и по-

лучившимися результатами, которая включает в себя следующие колонки:

- 1)  $\alpha$  параметр влияния длины пути;
- 2)  $\beta$  параметр влияния феромона;
- 3)  $\rho$  коэффициент испарения феромона;
- 4) дни количество дней жизни колонии муравьёв;
- 5) результат АПП результат, полученный алгоритмом полного перебора;
- 6) результат MA результат, полученный муравьиным алгоритмом на данных параметрах;
- 7) разница разность результата МА с результатом АПП.

#### 4.2.1 Класс данных №1

Класс данных №1 собой представляет собой матрицу смежности размером  $10 \times 10$ , элементы которой являются расстояниями между городами России, 3 реки, сезон зимний. Таблица 2 содержит параметры и результаты для первого класса данных.

Матрица смежностей для класса данных №1:

$$\begin{bmatrix} 0 & 450 & 700 & -1 & 320 & 500 & 650 & -1 & 400 & 580 \\ 450 & 0 & 350 & -1 & -1 & -1 & 300 & 700 & -1 & 450 \\ 700 & 350 & 0 & 200 & 500 & -1 & 450 & -1 & -1 & 600 \\ -1 & -1 & 200 & 0 & 400 & 550 & -1 & 300 & 650 & -1 \\ 320 & -1 & 500 & 400 & 0 & -1 & -1 & -1 & 300 & 100 \\ 500 & -1 & -1 & 550 & -1 & 0 & -1 & 400 & -1 & -1 \\ 650 & 300 & 450 & -1 & -1 & -1 & 0 & 350 & -1 & 200 \\ -1 & 700 & -1 & 300 & -1 & 400 & 350 & 0 & 550 & -1 \\ 400 & -1 & -1 & 650 & 300 & -1 & -1 & 550 & 0 & 250 \\ 580 & 450 & 600 & -1 & 100 & -1 & 200 & -1 & 250 & 0 \end{bmatrix}$$

Направления рек:  $(2 \to 3)$ ,  $(5 \to 8)$  и  $(7 \to 1)$ .

Таблица 2 – Параметры для класса данных №1

α	β	ρ	дни	результат АПП	результат МА	разница
0.10	0.90	0.10	1	2550.00	2900.00	350.00
0.10	0.90	0.10	3	2550.00	3000.00	450.00
0.10	0.90	0.10	5	2550.00	2720.00	170.00
0.10	0.90	0.10	10	2550.00	2770.00	220.00
0.10	0.90	0.10	30	2550.00	2570.00	20.00
0.10	0.90	0.10	60	2550.00	2550.00	0.00
0.10	0.90	0.10	100	2550.00	2580.00	30.00
0.10	0.90	0.10	150	2550.00	2580.00	30.00
0.10	0.90	0.10	200	2550.00	2550.00	0.00
0.10	0.90	0.10	300	2550.00	2630.00	80.00
0.10	0.90	0.20	1	2550.00	3570.00	1020.00
0.10	0.90	0.20	3	2550.00	3150.00	600.00
0.10	0.90	0.20	5	2550.00	2900.00	350.00
0.10	0.90	0.20	10	2550.00	2750.00	200.00
0.10	0.90	0.20	30	2550.00	2750.00	200.00
		•••				•••
0.60	0.40	0.40	1	2550.00	3350.00	800.00
0.60	0.40	0.40	3	2550.00	3070.00	520.00
0.60	0.40	0.40	5	2550.00	3150.00	600.00
0.60	0.40	0.40	10	2550.00	2870.00	320.00
0.60	0.40	0.40	30	2550.00	2850.00	300.00
0.60	0.40	0.40	60	2550.00	2650.00	100.00
0.60	0.40	0.40	100	2550.00	2650.00	100.00
0.60	0.40	0.40	150	2550.00	2600.00	50.00
0.60	0.40	0.40	200	2550.00	2620.00	70.00
0.60	0.40	0.40	300	2550.00	2550.00	0.00
			•••			•••
0.90	0.10	0.80	300	2550.00	2650.00	100.00
0.90	0.10	0.90	1	2550.00	3520.00	970.00
0.90	0.10	0.90	3	2550.00	3170.00	620.00
0.90	0.10	0.90	5	2550.00	3080.00	530.00
0.90	0.10	0.90	10	2550.00	2970.00	420.00
0.90	0.10	0.90	30	2550.00	2900.00	350.00
0.90	0.10	0.90	60	2550.00	2750.00	200.00
0.90	0.10	0.90	100	2550.00	2800.00	250.00
0.90	0.10	0.90	150	2550.00	2600.00	50.00
0.90	0.10	0.90	200	2550.00	2650.00	100.00
0.90	0.10	0.90	300	2550.00	2620.00	70.00

Наилучшими параметрами по результатам данного измерения для класса №1 были выбраны  $\pmb{\alpha}=0.50,\,\pmb{\beta}=0.50,\,\pmb{\rho}=0.90$  и количество дней  $\geq 100.$ 

#### 4.2.2 Класс данных №2

Класс данных N2 собой представляет собой матрицу смежности размером  $10 \times 10$ , элементы которой являются расстояниями между городами России, 2 реки, сезон

летний. Таблица 3 содержит параметры и результаты для второго класса данных. Матрица смежностей для класса данных №2:

$$\begin{bmatrix} 0 & 300 & -1 & 450 & 600 & -1 & 500 & 700 & -1 & -1 \\ 300 & 0 & 350 & -1 & -1 & 200 & -1 & -1 & 650 & -1 \\ -1 & 350 & 0 & 200 & -1 & -1 & -1 & 400 & -1 & 300 \\ 450 & -1 & 200 & 0 & -1 & 550 & -1 & -1 & 600 & -1 \\ 600 & -1 & -1 & -1 & 0 & -1 & 300 & 700 & -1 & -1 \\ -1 & 200 & -1 & 550 & -1 & 0 & 400 & -1 & -1 & -1 \\ 500 & -1 & -1 & -1 & 300 & 400 & 0 & -1 & 250 & -1 \\ 700 & -1 & 400 & -1 & 700 & -1 & -1 & 0 & -1 & 450 \\ -1 & 650 & -1 & 600 & -1 & -1 & 250 & -1 & 0 & 200 \\ -1 & -1 & 300 & -1 & -1 & -1 & -1 & 450 & 200 & 0 \\ \end{bmatrix}$$

Направления рек:  $(4 \to 6)$  и  $(9 \to 7)$ .

Таблица 3 – Параметры для класса данных №2

α	β	ρ	дни	результат АПП	результат МА	разница
0.10	0.90	0.10	1	2375.00	2475.00	100.00
0.10	0.90	0.10	3	2375.00	3075.00	700.00
0.10	0.90	0.10	5	2375.00	2825.00	450.00
0.10	0.90	0.10	10	2375.00	2525.00	150.00
0.10	0.90	0.10	30	2375.00	2475.00	100.00
0.10	0.90	0.10	60	2375.00	2475.00	100.00
0.10	0.90	0.10	100	2375.00	2375.00	0.00
0.10	0.90	0.10	150	2375.00	2375.00	0.00
0.10	0.90	0.10	200	2375.00	2375.00	0.00
0.10	0.90	0.10	300	2375.00	2375.00	0.00
0.10	0.90	0.20	1	2375.00	3475.00	1100.00
0.10	0.90	0.20	3	2375.00	2925.00	550.00
0.10	0.90	0.20	5	2375.00	2775.00	400.00
0.10	0.90	0.20	10	2375.00	2850.00	475.00
0.10	0.90	0.20	30	2375.00	2375.00	0.00
0.10	0.90	0.20	60	2375.00	2375.00	0.00
0.60	0.40	0.40	1	2375.00	2625.00	250.00
0.60	0.40	0.40	3	2375.00	2875.00	500.00
0.60	0.40	0.40	5	2375.00	2775.00	400.00
0.60	0.40	0.40	10	2375.00	2525.00	150.00
0.60	0.40	0.40	30	2375.00	2775.00	400.00
0.60	0.40	0.40	60	2375.00	2375.00	0.00
0.60	0.40	0.40	100	2375.00	2675.00	300.00
0.60	0.40	0.40	150	2375.00	2475.00	100.00
0.60	0.40	0.40	200	2375.00	2375.00	0.00
0.60	0.40	0.40	300	2375.00	2375.00	0.00
		•••				
0.90	0.10	0.90	3	2375.00	3275.00	900.00
0.90	0.10	0.90	5	2375.00	2525.00	150.00
0.90	0.10	0.90	10	2375.00	2525.00	150.00
0.90	0.10	0.90	30	2375.00	2775.00	400.00
0.90	0.10	0.90	60	2375.00	2375.00	0.00
0.90	0.10	0.90	100	2375.00	2475.00	100.00
0.90	0.10	0.90	150	2375.00	2725.00	350.00
0.90	0.10	0.90	200	2375.00	2375.00	0.00
0.90	0.10	0.90	300	2375.00	2475.00	100.00

Наилучшими параметрами по результатам данного измерения для класса №2 были выбраны  $\alpha=0.30,\, \beta=0.70,\, \rho=0.30$  и количество дней  $\geq 30.$ 

#### 4.2.3 Класс данных №3

Класс данных №3 собой представляет собой матрицу смежности размером  $10 \times 10$ , элементы которой являются расстояниями между городами России, 2 реки, сезон летний, в графе малое количество рёбер. Таблица 4 содержит параметры и результаты

для третьего класса данных.

Матрица смежностей для класса данных №3:

Направления рек:  $(1 \to 5)$  и  $(6 \to 10)$ .

Таблица 4 – Параметры для класса данных №3

α	β	ρ	дни	результат АПП	результат МА	разница
0.10	0.90	0.10	1	3350.00	3450.00	100.00
0.10	0.90	0.10	3	3350.00	3500.00	150.00
0.10	0.90	0.10	5	3350.00	3350.00	0.00
0.10	0.90	0.10	10	3350.00	3400.00	50.00
0.10	0.90	0.10	30	3350.00	3350.00	0.00
0.10	0.90	0.10	60	3350.00	3350.00	0.00
0.10	0.90	0.10	100	3350.00	3350.00	0.00
0.10	0.90	0.10	150	3350.00	3350.00	0.00
0.10	0.90	0.10	200	3350.00	3350.00	0.00
0.10	0.90	0.10	300	3350.00	3350.00	0.00
0.10	0.90	0.20	1	3350.00	3500.00	150.00
0.10	0.90	0.20	3	3350.00	3350.00	0.00
0.10	0.90	0.20	5	3350.00	3450.00	100.00
0.10	0.90	0.20	10	3350.00	3400.00	50.00
0.10	0.90	0.20	30	3350.00	3350.00	0.00
0.10	0.90	0.20	60	3350.00	3350.00	0.00
0.10	0.90	0.20	100	3350.00	3350.00	0.00
0.10	0.90	0.20	150	3350.00	3350.00	0.00
0.10	0.90	0.20	200	3350.00	3350.00	0.00
0.10	0.90	0.20	300	3350.00	3350.00	0.00
0.50	0.50	0.80	300	3350.00	3350.00	0.00
0.50	0.50	0.90	1	3350.00	3800.00	450.00
0.50	0.50	0.90	3	3350.00	3350.00	0.00
0.50	0.50	0.90	5	3350.00	3350.00	0.00
0.50	0.50	0.90	10	3350.00	3350.00	0.00
0.50	0.50	0.90	30	3350.00	3350.00	0.00
0.50	0.50	0.90	60	3350.00	3350.00	0.00
0.50	0.50	0.90	100	3350.00	3350.00	0.00
0.50	0.50	0.90	150	3350.00	3350.00	0.00
0.50	0.50	0.90	200	3350.00	3350.00	0.00
0.50	0.50	0.90	300	3350.00	3350.00	0.00
	0.10		····		2250.00	
0.90	0.10	0.90	5	3350.00	3350.00	0.00
0.90	0.10	0.90	$\begin{vmatrix} 10 \\ 20 \end{vmatrix}$	3350.00	3450.00	100.00
$0.90 \\ 0.90$	$\begin{vmatrix} 0.10 \\ 0.10 \end{vmatrix}$	$0.90 \\ 0.90$	$\begin{vmatrix} 30 \\ 60 \end{vmatrix}$	$3350.00 \\ 3350.00$	$3350.00 \\ 3350.00$	$0.00 \\ 0.00$
0.90	$0.10 \ 0.10$	0.90	100	3350.00	3350.00	$0.00 \\ 0.00$
0.90	$0.10 \ 0.10$	$0.90 \\ 0.90$	150	3350.00	3350.00	0.00
0.90	$0.10 \ 0.10$	$0.90 \\ 0.90$	$\frac{150}{200}$	3350.00	3350.00	$0.00 \\ 0.00$
0.90	$0.10 \ 0.10$	0.90	$\frac{200}{300}$	3350.00	3350.00	0.00
0.90	0.10	0.90	500	00.000	00.00.00	0.00

В связи с малым количеством рёбер муравьиный алгоритм практически на всех параметрах выдаёт значения идентичные или близкие к результатам полного перебора. Особо лишь хочется отметить, что расхождения в основном наблюдаются при количество дней меньше 10.

#### Выводы

В данном разделе был приведён пример работы программы, а также проведена автоматическая параметризация для определения наилучших параметров для муравьиного алгоритма на трёх различных входных данных. Была замечена зависимость точности муравьиного алгоритма в зависимости от количества рёбер в графе и числа дней жизни муравьиной колонии.

#### ЗАКЛЮЧЕНИЕ

В результате выполнения лабораторной работы были исследованы два подхода к решению задачи коммивояжёра: метод полного перебора и муравьиный алгоритм.

В частности:

- 1) рассмотрен метод полного перебора;
- 2) рассмотрен муравьиный алгоритм;
- 3) реализованы оба алгоритма решения задачи коммивояжёра на выбранном языке программирования;
- 4) проведена параметризация муравьиного алгоритма по четырём параметрам и оценено качество решений для серии графов.

В ходе лабораторной работы были рассмотрены, спроектированы и запрограммированы алгоритм полного перебора и муравьиный алгоритм решения задачи коммивояжёра.

Были разработаны тесты для всех алгоритмов, ожидаемых результатов которых достигли все реализации.

Проведённые исследования по запуску трёх графов на серии различных комбинаций входных параметров показал уменьшение отличия результата муравьиного алгоритма от эталонного ответа, данного полным перебором, с уменьшение числа рёбер графа и с увеличением числа дней жизни муравьиной колонии.

### СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Синельников Д.А. Муравьиный алгоритм в решении задачи коммивояжёра: Бакалаврская работа. Тольятти: Тольяттинский государственный университет, 2018. 01.03.02. Прикладная математика и информатика. Научный руководитель: к.т.н., доцент А.В. Очеповский.
- [2] Галяутдинов Р.Р. Задача коммивояжёра метод ветвей и границ // Сайт преподавателя экономики. [2023]. URL: https://galyautdinov.ru/post/zadacha-kommivoyazhera (дата обращения: 11.12.2024).