**Міністерство освіти і науки України**

**Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформатики та програмної інженерії**

**Звіт**

з лабораторної роботи № 5 з дисципліни

«Проєктування алгоритмів»

„**Проєктування і аналіз алгоритмів для вирішення NP-складних задач ч.2**”

**Виконав(ла)**

(шифр, прізвище, ім'я, по батькові)

*ІП-32 Юрченко Богдан Валерійович*

**Перевірив**

(прізвище, ім'я, по батькові)

*Соколовський В.В.*

Київ 2024

Зміст

[1 Мета лабораторної роботи 3](#_Toc52291748)

[2 Завдання 4](#_Toc52291749)

[3 Виконання 10](#_Toc52291750)

[3.1 Покроковий алгоритм 10](#_Toc52291751)

[3.2 Програмна реалізація алгоритму 10](#_Toc52291752)

[3.2.1 Вихідний код 10](#_Toc52291753)

[3.2.2 Приклади роботи 10](#_Toc52291754)

[3.3 Тестування алгоритму 11](#_Toc52291755)

[Висновок 12](#_Toc52291756)

[Критерії оцінювання 13](#_Toc52291757)

# Мета лабораторної роботи

Мета роботи – вивчити основні підходи розробки метаеврестичних алгоритмів для типових прикладних задач. Опрацювати методологію підбору прийнятних параметрів алгоритму.

# Завдання

Згідно з варіантом, формалізувати алгоритм вирішення задачі відповідно до загальної методології.

Записати розроблений алгоритм у покроковому вигляді. З достатнім степенем деталізації.

Виконати його програмну реалізацію на будь-якій мові програмування.

Перелік задач наведено у таблиці 2.1.

Перелік алгоритмів і досліджуваних параметрів у таблиці 2.2.

Задача і алгоритм наведені в таблиці 2.3.

Змінюючи параметри алгоритму, визначити кращі вхідні параметри алгоритму. Для цього необхідно:

* обрати критерій зупинки алгоритму (кількість ітерацій або значення ЦФ);
* зафіксувати усі параметри крім одного і змінювати цей параметр, поки не буде досягнуто пікової ефективності;
* після цього параметр фіксується і змінюються інші параметри;
* далі повторюємо процедуру спочатку, з першого зафіксованого параметру;
* зупиняємось, коли будуть знайдені кращі значення вхідних параметрів для даної задачі або встановлена залежність одних параметрів від інших.

Зробити узагальнений висновок в якому обов’язково описати залежність якості розв’язку від вхідних параметрів.

Таблиця 2.1 – Прикладні задачі

|  |  |
| --- | --- |
| **№** | **Задача** |
| 2 | **Задача комівояжера** (300 вершин, відстань між вершинами випадкова від 5 до 150) полягає у знаходженні найвигіднішого маршруту, що проходить через вказані міста хоча б по одному разу. В умовах завдання вказуються критерій вигідності маршруту (найкоротший, найдешевший, сукупний критерій тощо) і відповідні матриці відстаней, вартості тощо. Зазвичай задано, що маршрут повинен проходити через кожне місто тільки один раз, в такому випадку розв'язок знаходиться серед гамільтонових циклів.  **Розглядається симетричний, асиметричний та змішаний варіанти.**  В загальному випадку, асиметрична задача комівояжера відрізняється тим, що ребра між вершинами можуть мати різну вагу в залежності від напряму, тобто, задача моделюється орієнтованим графом. Таким чином, окрім ваги ребер графа, слід також зважати і на те, в якому напрямку знаходяться ребра.  У випадку симетричної задачі всі пари ребер між одними й тими самими вершинами мають однакову вагу.  У випадку реальних міст може бути як симетричною, так і асиметричною в залежності від тривалості або довжини маршрутів і напряму руху.  Застосування:   * доставка товарів (в цьому випадку може бути більш доречна постановка транспортної задачі - доставка в кілька магазинів з декількох складів); * доставка води; * моніторинг об'єктів; * поповнення банкоматів готівкою; * збір співробітників для доставки вахтовим методом. |

Таблиця 2.2 – Варіанти алгоритмів і досліджувані параметри

|  |  |
| --- | --- |
| **№** | **Алгоритми і досліджувані параметри** |
| 2 | **Мурашиний алгоритм**:   * α; * β; * ρ; * Lmin; * кількість мурах М і їх типи (елітні, тощо…); * маршрути з однієї чи різних вершин. |

Таблиця 2.3 – Варіанти задач і алгоритмів

|  |  |
| --- | --- |
| **№** | **Задачі і алгоритми** |
| 28 | Задача комівояжера (асиметрична мережа) + Мурашиний алгоритм |

# Виконання

## Покроковий алгоритм

1. Створення мурах
2. Пошук рішення усіма мурахами
3. Перевірка умови виходу, продовження алгоритму якщо вона не виконана
4. Оновлення феромону
5. Наступна ітерація

## Програмна реалізація алгоритму

### Вихідний код

### Приклади роботи

На рисунках 3.1 і 3.2 показані приклади роботи програми.

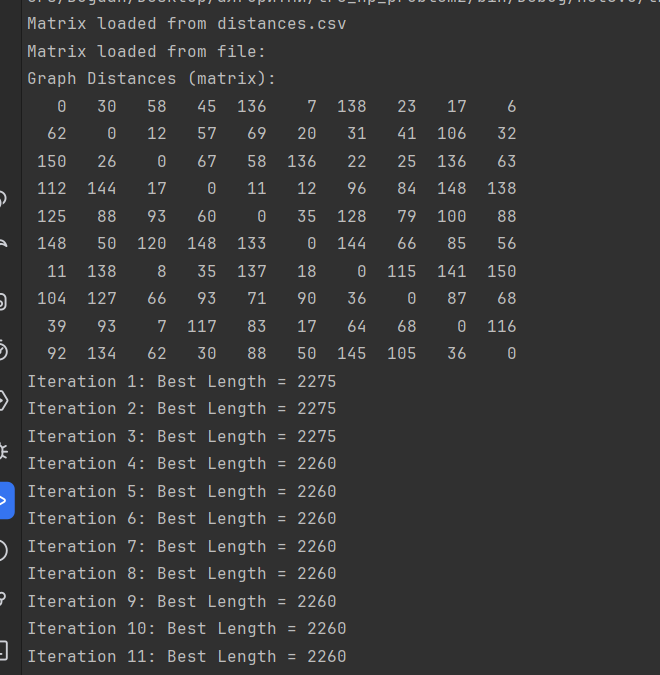


Рисунок 3.1

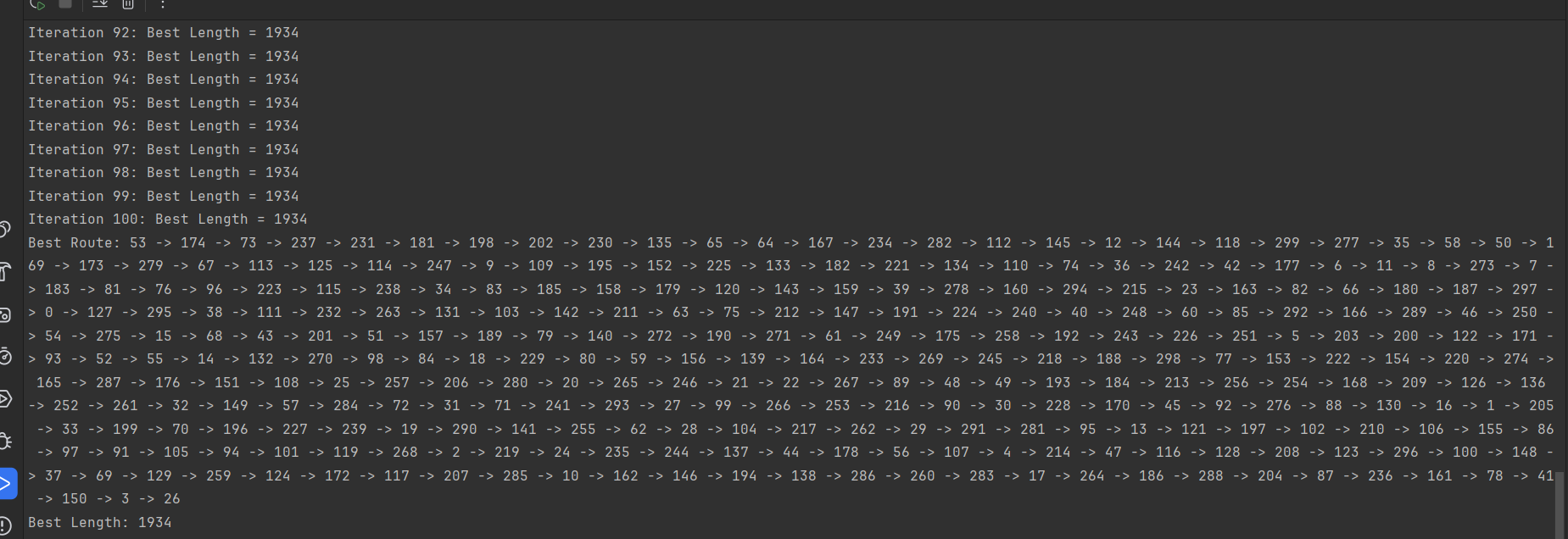


Рисунок 3.2

## Тестування алгоритму

Дослідимо якість розв’язку в залежності від кількості ітерацій на одній і тій самій матриці відстаней на 300 вершин:

Таблиця 3.3.1 – Залежність якості розв’язку від кількості ітерацій

|  |  |
| --- | --- |
| Кількість ітерацій | Довжина маршруту |
| 10 | 2275 |
| 20 | 2214 |
| 50 | 1977 |
| 100 | 1890 |
| 150 | 1875 |
| 200 | 1848 |
| 500 | 1843 |

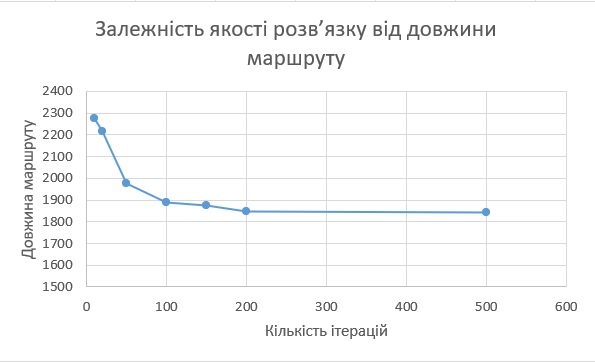


Рисунок 3.3 – Графік залежності якості розв’язку від довжини маршруту

Тепер Дослідимо якість розв’язку в залежності від кількості мурах на одній і тій самій матриці відстаней на 300 вершин на 100 ітерацій:

Таблиця 3.3.1 – Залежність якості розв’язку від кількості мурах

|  |  |
| --- | --- |
| Кількість ітерацій | Довжина маршруту |
| 1 | 2016 |
| 2 | 1979 |
| 5 | 1902 |
| 10 | 1897 |
| 20 | 1877 |
| 50 | 1851 |
| 100 | 1821 |

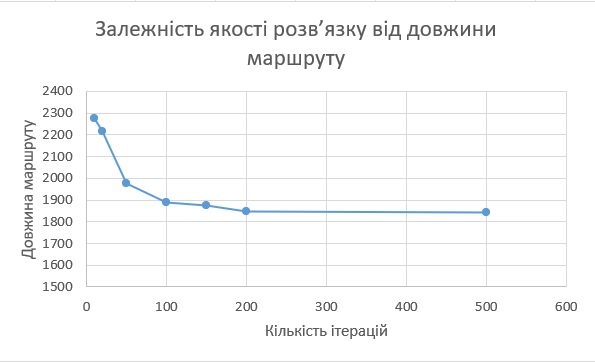


Рисунок 3.3 – Графік залежності якості розв’язку від кількості мурах

Висновок

В рамках даної лабораторної роботи було досліджено розв’язання задачі комівояжера в асиметричній мережі на 300 вершин за допомогою мурашиного алгоритму. Було досліджено параметри даного алгоритму, в особливості кількість ітерацій алгоритму та кількість мурах.

Критерії оцінювання

Критерії оцінювання у відсотках від максимального балу:

* покроковий алгоритм – 15%;
* програмна реалізація алгоритму – 50%;
* тестування алгоритму– 30%;
* висновок – 5%.