**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

отчет

**по лабораторной работе №1**

**по дисциплине «Основы искусственного интеллекта»**

Тема: Муравьиный алгоритм

|  |  |  |
| --- | --- | --- |
| Студенты гр. 9302 |  | Ширнин К.В. |
|  |  | Квитко Д.В. |
| Преподаватель |  | Новакова Н.Е. |

Санкт-Петербург

2022

**Оглавление**

[Цель работы 3](#_Toc115821449)

[Реализация алгоритма 3](#_Toc115821450)

[Ход работы 5](#_Toc115821451)

[Пример работы программы 6](#_Toc115821452)

[Вывод 8](#_Toc115821453)

[Список используемой литературы 8](#_Toc115821454)

# Цель работы

Изучить принцип работы муравьиного алгоритма. Написать ПО с пользовательским графическим интерфейсом, находящее кратчайший маршрут среди всех узлов данного графа, с использованием муравьиного алгоритма

**Теория**

Вероятность посещения муравьем смежной точки определяется формулой:

,

где – коэффициент привлекательности феромона,

- коэффициент привлекательности кратчайшего пути,

- количество феромона на пути к вершине,

– обратная величина пути к вершине,

– любая смежная доступная вершина,

i, j – номера вершин, которые соединяет ребро.

Испарение старого феромона с заданным коэффициентом определяется по формуле:

,

где – количество феромона на дорожке,

t – номер итерации,

i, j – номера вершин, которые соединяет ребро,

p – коэффициент испарения феромона.

Количество феромона, которое добавляется на каждый пройдённый муравьями маршрут, рассчитывается по следующей формуле:

Где Q – заданный пользователем коэффициент,

– пройденный муравьем путь.

# Особенности реализации алгоритма

Для реализации муравьиного алгоритма ПО было разделено на две составляющие, а именно на GUI и сам программный код алгоритма. Программный комплекс был написан в рамках правил ООП и без особых трудностей может использоваться как отдельный виджет в других ПО или как библиотека.

Ход алгоритма делится на несколько этапов – он делится на итерации. После каждой итерации производится обновление количества феромона на дорожках – рёбрах графа – с учетом скорости их испарения и других заданных пользователем параметров. Одна итерация, в свою очередь, делится на проход пачки муравьев. Их количество задается пользователем. Каждый муравей из пачки может пойти в абсолютно любую другую смежную вершину, даже если на этой дорожке большое количество феромона. Это происходит из-за того, что выбор муравья определяется случайной составляющей. С учетом удаленности вершины и рассыпанного по пути к ней феромона муравью более привлекательна та или иная вершина, но даже у самой непривлекательной точки будет иметься вероятность посещения её муравьем. Это позволяет изменить даже, самый лучший маршрут на другой и найти еще более оптимальный путь. После того как все муравьи прошли все вершины - производится обновление количества феромона на дорожках. Сначала испаряется старый феромон с заданным коэффициентом. После, ко всем пройденным маршрутам муравьями на данной итерации добавляется некоторое количество феромона.

# UML-диаграмма

На рисунке 1 представлена UML-диаграмма структуры программы

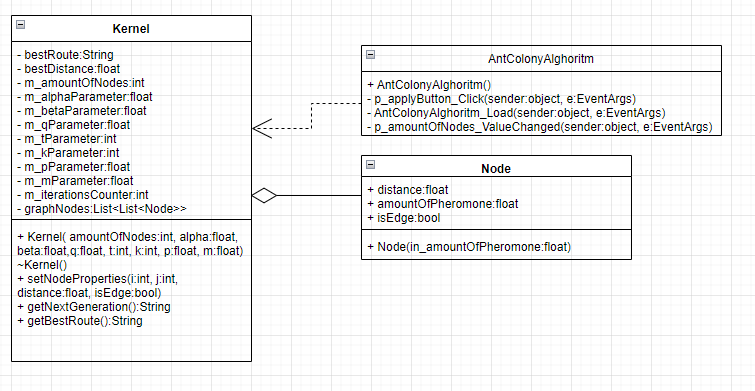


Рис. 1. UML-диаграмма

Таблица 1. Методы и их описание

|  |  |
| --- | --- |
| Метод | Описание |
| public Kernel(int amountOfNodes,float alpha,float beta,float q,int t,int k,float p,float m) | Конструктор класса Kernel |
| ~Kernel() | Деструктор класса Kernel |
| public void setNodeProperties(int i, int j, float distance, bool isEdge) | Устанавливает параметры Node’а |
| public String getNextGeneration() | Вызывает срабатывание одного такта алгоритма |
| public String getBestRoute() | Возвращает лучший маршрут |
| public Node(float in\_amountOfPheromone) | Конструктор класса Node |
| private void p\_applyButton\_Click(object sender, EventArgs e) | Запускает работу алгоритма при нажатии пользователем кнопки |
| private void AntColonyAlghoritm\_Load(object sender, EventArgs e) | Вызывается при загрузки формы запускает алгоритм построения динамических ячеек |
| private void p\_amountOfNodes\_ValueChanged(object sender, EventArgs e) | Алгоритм построения динамических ячеек |

# 

# Ход работы

Для выполнения данной лабораторной работы был выбран язык C#.

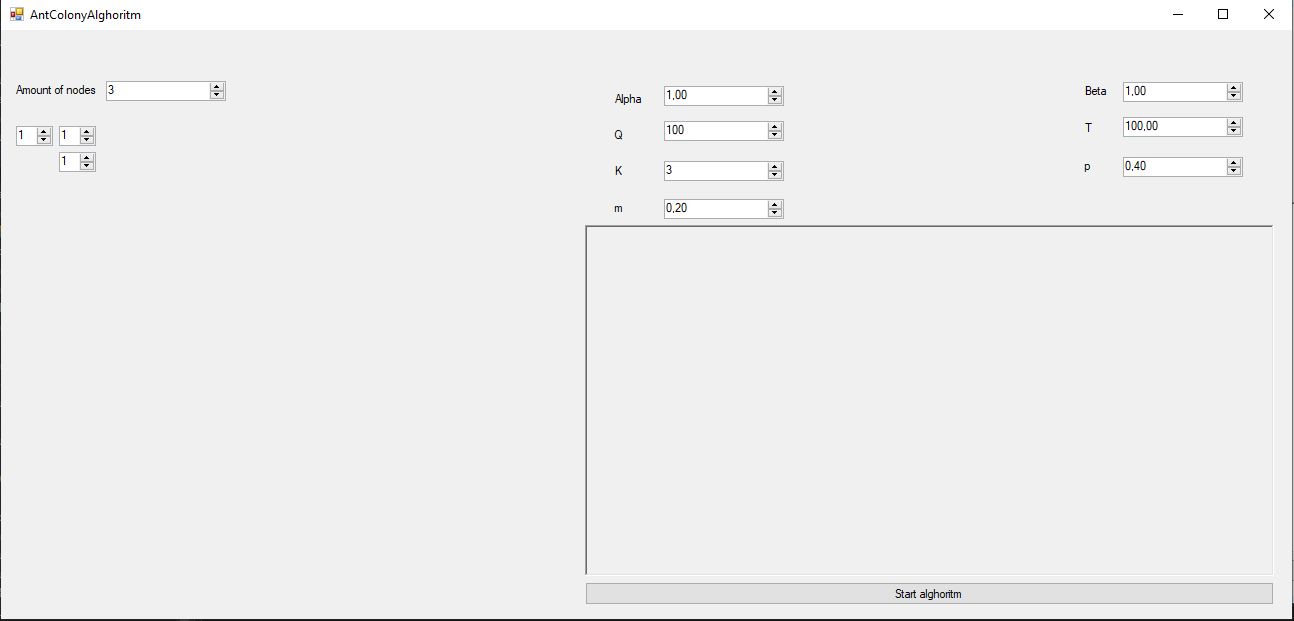


Рис.2. Главное окно приложения

При выполнении работы, были взяты основные используемые параметры:

* Альфа – коэффициент привлекательности феромона на дорожке
* Бета – коэффициент привлекательности более короткой дорожки
* Q – Константа, определяющая сколько феромона оставит за собой муравей
* T – Количество итераций алгоритма
* K – Количество муравьев (кол-во итераций без обновления количества феромонов на дорожках)
* p – Скорость испарения феромона с дорожек
* m – Начальное количество феромона на дорожка

Для задачи графа была реализована динамическая составляющая пользовательского интерфейса. Пользователю достаточно выставить в поле amount of nodes количество вершин в графе, после чего программа сама отобразит матрицу смежности нужного размера. Она представлена в виде треугольника, т.к. в муравьином алгоритме будет производиться расчет на неориентированном графе (что указывает на симметричность матрицу относительно главной диагонали). Это позволит сэкономить время при вводе данных, а также в два раза сократит расходы на выделяемую память, так как хранение внутри программы производится тем же самым образом.

При выполнении работы также были учтены крайние значения, вводимые пользователем в программу.

# Пример работы программы

Пример работы программы представлен на рисунках ниже.

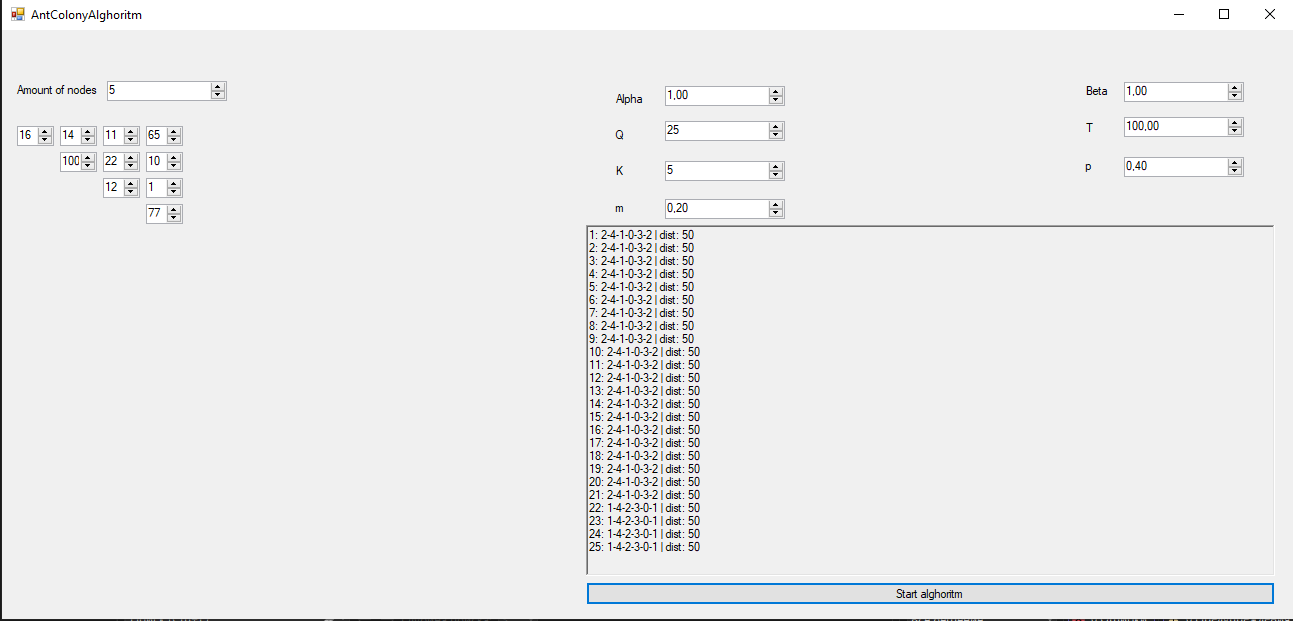


Рис. 3. Пример работы программы для 5 вершин

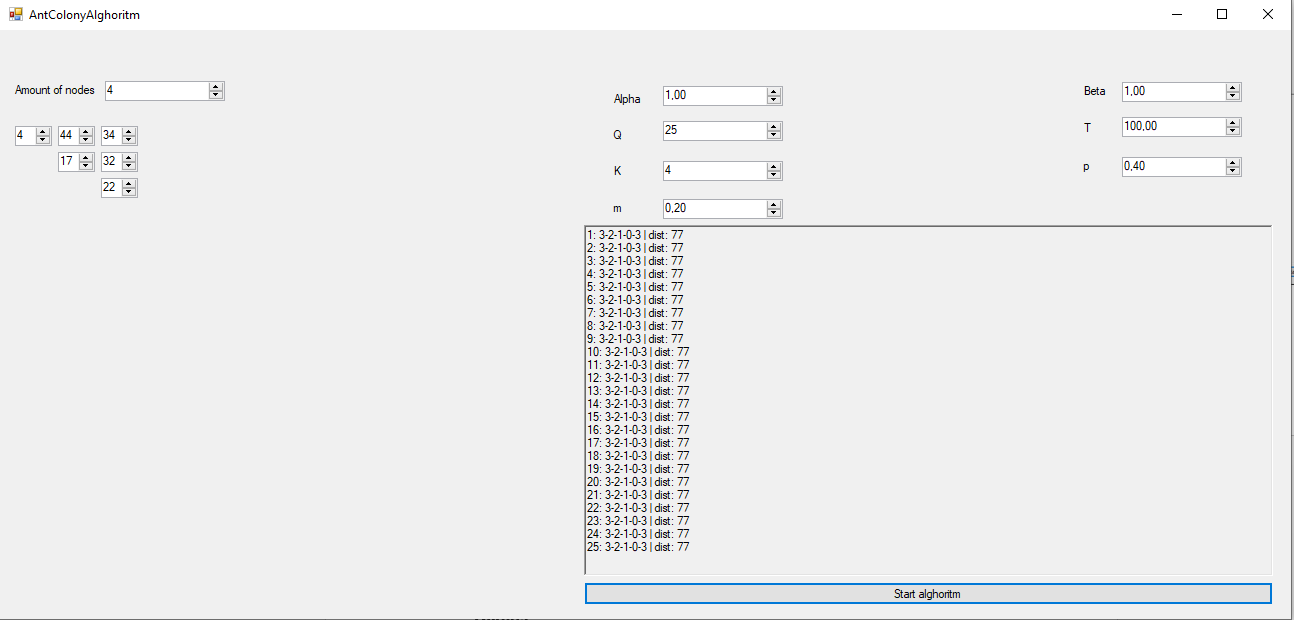


Рис.4. Пример работы программы для 4 вершин

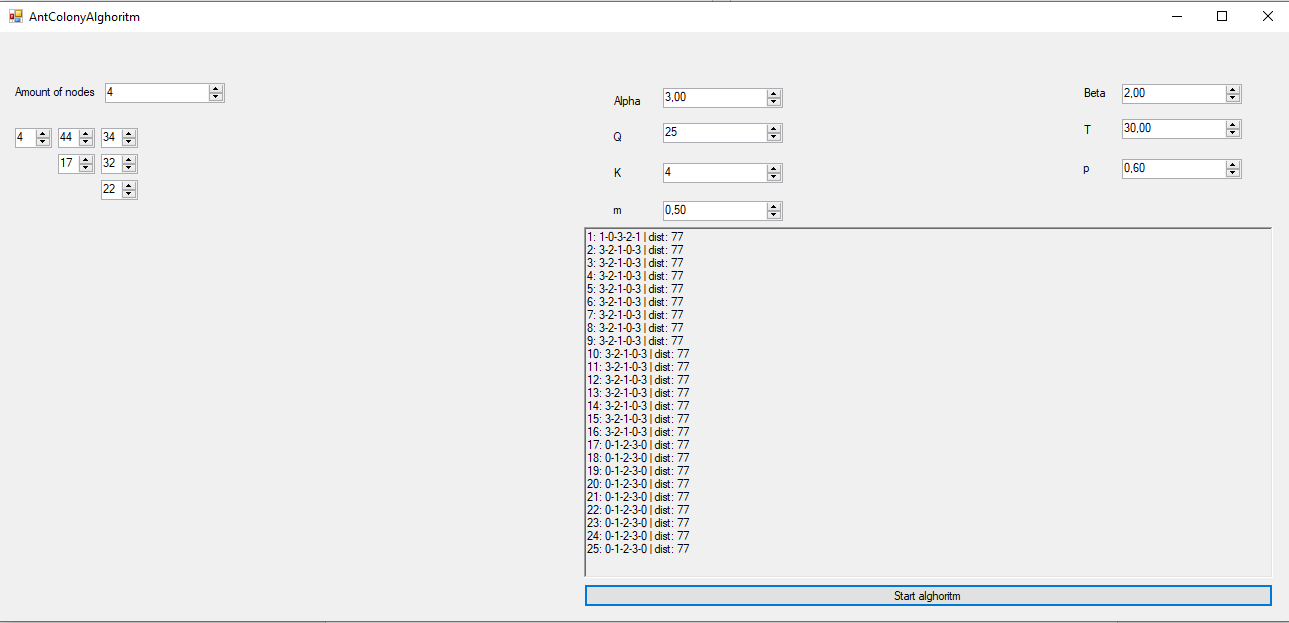


Рис.5. Пример работы программы для 4 вершин

# Вывод

В ходе выполнения работы стало ясно, что муравьиный алгоритм очень сильно зависим от заданных констант и параметров. Абсолютно к каждой новой задаваемой матрице смежности требуется подбирать свои индивидуальные параметры. Если подобрать конфигурацию неверно – алгоритм может проложить сильно неверный путь или зайти в тупик, что не имеет никакой практической ценности для решения поставленных задач. Алгоритм был успешно реализован на языке C#. В лабораторной работе были достигнуты все цели.

# Список используемой литературы

1. Муравьиные алгоритмы // Хабр URL: https://habr.com/ru/post/105302/ (дата обращения: 21.09.2022).