# ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

### Кафедра АСУ

### 

Отчёт

о лабораторной работе №7

по курсу: «Решение задачи оптимизации муравьиным алгоритмом»

# Вариант «2»

Выполнил:

ст. гр. ИСм-22

Мариничев И.И

Проверила:

Васяева Т.А

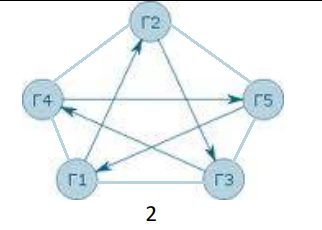
# 

# Донецк 2022

**Ход работы:**

1. Разработать МА для решение задачи поиска гамильтонова пути. Индивидуальное заданию выбирается по таблице E.1 в приложении E согласно номеру варианта.
2. Выполнить программную реализацию разработанного алгоритма на языке высокого уровня. Рекомендованный язык Python с применением DEAP. Предусмотреть возможность просмотра процесса поиска решения.
3. Представить графически найденное решение. Предусмотреть возможность пошагового просмотра процесса поиска решения.
4. Сравнить найденное решение с представленным в условии задачи оптимальным решением.

*Задание:*



В качестве точки остановки был выбран следующий критерий: должен быть найден такой путь, длина которого не превышает количество вершин и который обязательно проходит через каждую вершину.

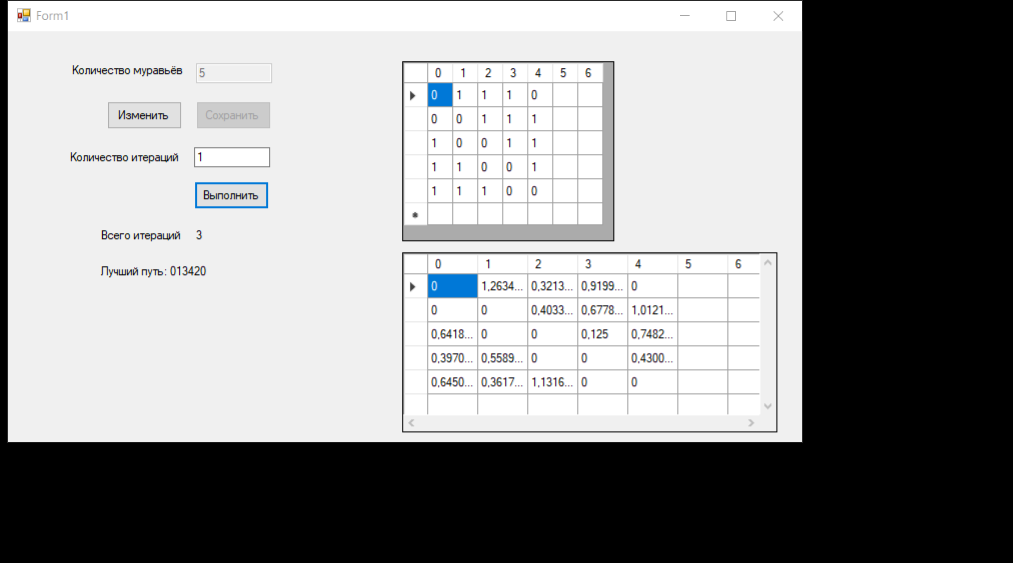


Рис. 1.1 – Результат работы

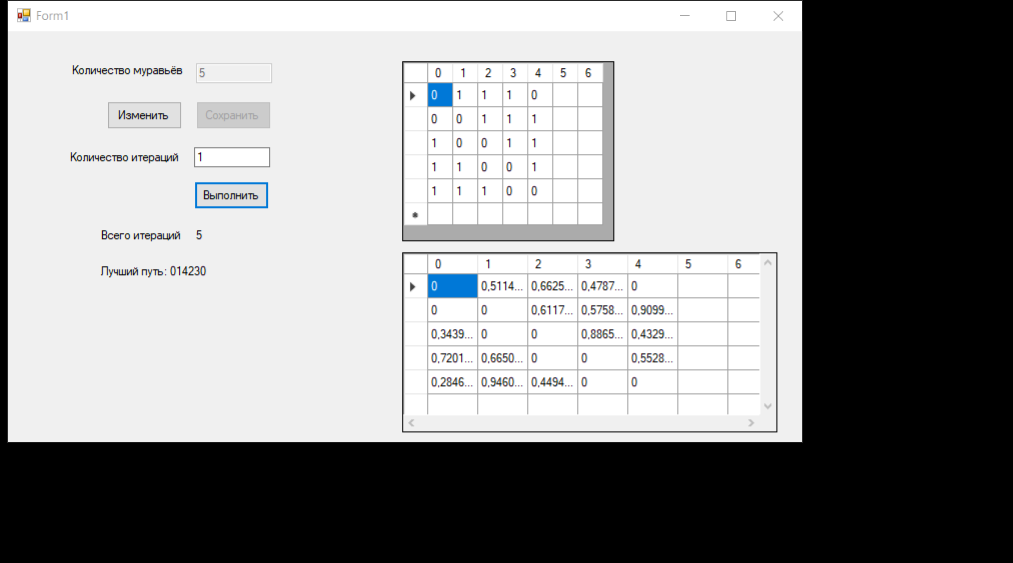


Рис. 1.2 – Результат работы

Уже примерно к 3-10 итерации алгоритм находит Гамильтонов путь.

Алгоритм работы программы:

for(количество итераций либо критерий остановки)

{

foreach(муравей в колонии)

{

do{

Вычисление шанса на переход для каждого следующего узла;

Запись узла в путь;

Выбор узла и переход на него;

if(пройдены все узлы) {окончание пути; переход к следующему муравью;}

else if(Проверка: является ли узел тупиковым){если является – путь не засчитывается; переход к следующему муравью}

}while(не построен путь)

}

//начисление феромона

foreach(каждый муравей в колонии)

{

foreach(каждый узел в пути) феромон узла+=1/количество узлов в пути;

}

foreach(каждый узел в графе) foreach(каждый переход в узле) испарение феромона;

}

Листинг программного кода:

Главная форма:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ЭМО\_лабораторная\_7

{

class Ant

{

public List<int> daway;

public bool colch;

public bool[] vertexes;

public Ant()

{

daway = new List<int>();

colch = true;

}

public void setvert(int colv)

{

vertexes = new bool[colv];

for (int i = 0; i < colv; i++)

{

vertexes[i] = false;

}

}

public string toString()

{

string str = "";

foreach(var v in daway)

{

str += Convert.ToString(v);

}

return str;

}

}

class AntAlgorithm

{

private double isparenie;

public double[,] feromons;

public List<Ant> ants;

public Random rand;

public AntAlgorithm()

{

isparenie = 0.25;

ants = new List<Ant>();

rand = new Random();

}

public void algorithminit(int colf)

{

feromons = new double[colf, colf];

for (int i = 0; i < colf; i++)

{

for (int j = 0; j < colf; j++)

{

if (Form1.Graph[i, j] == 1)

{

feromons[i, j] = 1;

}

}

}

for (int i = 0; i < colf; i++)

{

ants.Add(new Ant());

}

}

public void maincikl(int coliter)

{

bool t1;

double sum;

List<double> chance = new List<double>();

List<int> chancenum = new List<int>();

int tmpv = 0;

for (int i = 0; i < coliter; i++)

{

foreach (var ant in ants)

{

ant.daway.Clear();

t1 = false;

ant.daway.Add(0);

ant.colch = true;

ant.setvert(Convert.ToInt32(Math.Sqrt(feromons.Length)));

do

{

sum = 0;

chance.Clear();

chancenum.Clear();

for (int j = 0; j < Math.Sqrt(feromons.Length); j++)

{

sum += feromons[ant.daway.Last(), j];//Form1.Graph[ant.daway.Last(), j];

if (Form1.Graph[ant.daway.Last(), j] > 0)

{

chancenum.Add(j);

}

}

if (chancenum.Count > 0)

{

if (chancenum.Count > 1)

{

chance.Add(feromons[ant.daway.Last(), chancenum[0]] / sum);

for (int j = 1; j < chancenum.Count; j++)

{

chance.Add(feromons[ant.daway.Last(), chancenum[j]] / sum);

chance[j] += chance[j - 1];

}

double tmp = rand.NextDouble();

for (int j = 0; j < chance.Count; j++)

{

if (tmp <= chance[j])

{ ant.daway.Add(chancenum[j]);

break;

}

}

tmpv = ant.daway.Last();

ant.vertexes[tmpv] = true;

}

else

{

ant.daway.Add(chancenum.Last());

tmpv = ant.daway.Last();

ant.vertexes[tmpv] = true;

}

}

else

{

ant.colch = false;

break;

}

//tmpv = ant.daway.Last();

//ant.vertexes[tmpv] = true;

//tmpv = 0;

}

while (ant.vertexes.Contains(false));

}

addferomons();

}

}

private void addferomons()

{

for(int i =0; i<Math.Sqrt(feromons.Length); i++)

{

for (int j = 0; j < Math.Sqrt(feromons.Length); j++)

{

if (feromons[i, j] > 0) feromons[i, j] \*= 0.5;

}

}

double timeferomon;

foreach (var element in ants)

{

double asd = element.daway.Count;

timeferomon = (1.0/asd);

for (int i = 1; i < element.daway.Count; i++)

{

feromons[element.daway[i - 1], element.daway[i]] += timeferomon;

}

}

}

}

}

Класс алгоритм:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ЭМО\_лабораторная\_7

{

public partial class Form1 : Form

{

public static int[,] Graph = new int[5, 5];

int kol\_ant;

AntAlgorithm alg = new AntAlgorithm();

public Form1()

{

InitializeComponent();

kol\_ant = Convert.ToInt32(this.textBox1.Text);

using (StreamReader reader = new StreamReader("Graph.txt"))

{

string line;

string[] coll;

int counter = 0;

while ((line = reader.ReadLine()) != null)

{

coll = line.Split(' ');

this.dataGridView1.Rows.Add();

//this.dataGridView1.Rows[counter].HeaderCell.Value = counter;

//Debug.Print(line);

for (int i = 0; i < coll.Length; i++)

{

if (int.Parse(coll[i]) > 0)

Graph[counter, i] = 1;

else

Graph[counter, i] = 0;

this.dataGridView1[i, counter].Value = Graph[counter, i];

}

counter++;

}

alg.algorithminit(counter);

}

}

private void button1\_Click(object sender, EventArgs e)

{

this.button2.Enabled = true;

this.textBox1.Enabled = true;

}

private void button2\_Click(object sender, EventArgs e)

{

kol\_ant = Convert.ToInt32(this.textBox1.Text);

this.button2.Enabled = false;

this.textBox1.Enabled = false;

}

private void button3\_Click(object sender, EventArgs e)

{

alg.maincikl(Convert.ToInt32(textBox2.Text));

int best = 0;

int tmpbestcol = alg.ants[0].daway.Count;

for (int i = 1; i < alg.ants.Count; i++)

{

if (alg.ants[i].daway.Count < tmpbestcol)

{

best = i;

tmpbestcol = alg.ants[i].daway.Count;

}

}

//string str = alg.ants[best].daway.ToString();

label5.Text = "Лучший путь: " + alg.ants[best].toString();

for(int i=0; i<Math.Sqrt(alg.feromons.Length); i++)

{

this.dataGridView2.Rows.Add();

for (int j = 0; j < Math.Sqrt(alg.feromons.Length); j++)

{

this.dataGridView2[j, i].Value = alg.feromons[i, j];

}

}

label4.Text = Convert.ToString(Convert.ToInt32(label4.Text) + 1);

}

}

}