Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Кафедра ИС

Лабораторная работа №3

ИССЛЕДОВАНИЕ ПРИМЕНЕНИЯ ТЕОРИИ ВАЖНОСТИ КРИТЕРИЕВ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ВЫБОРА АЛЬТЕРНАТИВ

Выполнил:

ст. гр. И-32-д

Серба А.В.

Проверил:

Токарев А.И.

Севастополь

2014

1. ЦЕЛЬ РАБОТЫ

Исследовать применение аппарата теории важности критериев при принятии решений по выбору альтернатив.

1. ЗАДАНИЕ НА РАБОТУ

Для третьего варианта задания предусматривается следующий порядок действий по выполнению лабораторной работы:

* 1. на основе задаваемых в качестве входных данных значений  и , соответствующих уступкам и приращениям для критериев  и , вычислить значения коэффициентов относительной важности критериев ;
  2. разработать процедуру определения доминируемых решений, выполняющую для каждого решения  сравнение его значений скалярных оценок  вектора  с такими же скалярными оценками  решений  ( и ); тем самым должны быть определены решения , доминируемые текущим рассматриваемым решением  (при  и ); результатом выполнения процедуры является множество  не сравнимых между собой с использованием отношения предпочтения решений;
  3. разработать процедуру определения значений векторных оценок  для всех *n* решений () c учетом вычисленных значений ; при этом учесть, что при < критерий является менее важным, чем критерий ; в этом случае пересчитываются скалярные оценки , соответствующие этому *j-*му критерию (в этом случае определяется скалярная оценка , входящая в модифицируемый вектор );
  4. для модифицированных векторных оценок  каждого решения  () проконтролировать выполнение условия доминирования им других решений  для их векторных оценок  (при  и ) (т.е. выполняется поэлементное сравнение оценок  и  из соответствующих векторов  и ); при выполнении условия , процедура реализует исключение решения  из множества : ;
  5. результатом выполнения разрабатываемой программы является определение множества не сравнимых решений , сформированного на основе информации об относительной важности критериев;
  6. выполнить вывод множества , полученного в результате исключения из него доминируемых решений  при учете информации об относительной важности критериев.

В качестве исходных данных для выполнения задания по лабораторной работе (для всех вариантов) заданы: множество решений вида , оценки пяти критериев сведены в Таблицу 4.

Таблица 1. Скалярные оценки  критериев  для решений  ()

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Варианты | Критерии | | | | |
|  |  |  |  |  |
|  | 3 | 5 | 5 | 4 | 4 |
|  | 4 | 4 | 4 | 5 | 4 |
|  | 5 | 4 | 3 | 3 | 5 |
|  | 3 | 5 | 3 | 5 | 3 |
|  | 4 | 2 | 4 | 5 | 5 |
|  | 3 | 5 | 3 | 5 | 3 |
|  | 5 | 3 | 4 | 3 | 4 |
|  | 4 | 5 | 3 | 4 | 3 |

1. ВАРИАНТ ЗАДАНИЯ

**Вариант 3.** Определить множество несравнимых решений , используя информацию об относительной важности критериев в следующем виде:

, ;

, .

1. ТЕКСТ ПРОГРАММЫ

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace TPR3

{

public partial class FormТеорияВажности : Form

{

public FormТеорияВажности()

{

InitializeComponent();

dataGridViewСкалярныеОценки.Columns.Add("K1", "K1");

dataGridViewСкалярныеОценки.Columns.Add("K2", "K2");

dataGridViewСкалярныеОценки.Columns.Add("K3", "K3");

dataGridViewСкалярныеОценки.Columns.Add("K4", "K4");

dataGridViewСкалярныеОценки.Columns.Add("K5", "K5");

dataGridViewСкалярныеОценки.Rows.Add();

dataGridViewСкалярныеОценки.Rows[0].Cells[0].Value = "x1";

dataGridViewСкалярныеОценки.Rows[0].Cells[1].Value = 3;

dataGridViewСкалярныеОценки.Rows[0].Cells[2].Value = 5;

dataGridViewСкалярныеОценки.Rows[0].Cells[3].Value = 5;

dataGridViewСкалярныеОценки.Rows[0].Cells[4].Value = 4;

dataGridViewСкалярныеОценки.Rows[0].Cells[5].Value = 4;

dataGridViewСкалярныеОценки.Rows.Add();

dataGridViewСкалярныеОценки.Rows[1].Cells[0].Value = "x2";

dataGridViewСкалярныеОценки.Rows[1].Cells[1].Value = 4;

dataGridViewСкалярныеОценки.Rows[1].Cells[2].Value = 4;

dataGridViewСкалярныеОценки.Rows[1].Cells[3].Value = 4;

dataGridViewСкалярныеОценки.Rows[1].Cells[4].Value = 5;

dataGridViewСкалярныеОценки.Rows[1].Cells[5].Value = 4;

dataGridViewСкалярныеОценки.Rows.Add();

dataGridViewСкалярныеОценки.Rows[2].Cells[0].Value = "x3";

dataGridViewСкалярныеОценки.Rows[2].Cells[1].Value = 5;

dataGridViewСкалярныеОценки.Rows[2].Cells[2].Value = 4;

dataGridViewСкалярныеОценки.Rows[2].Cells[3].Value = 3;

dataGridViewСкалярныеОценки.Rows[2].Cells[4].Value = 3;

dataGridViewСкалярныеОценки.Rows[2].Cells[5].Value = 5;

dataGridViewСкалярныеОценки.Rows.Add();

dataGridViewСкалярныеОценки.Rows[3].Cells[0].Value = "x4";

dataGridViewСкалярныеОценки.Rows[3].Cells[1].Value = 3;

dataGridViewСкалярныеОценки.Rows[3].Cells[2].Value = 5;

dataGridViewСкалярныеОценки.Rows[3].Cells[3].Value = 3;

dataGridViewСкалярныеОценки.Rows[3].Cells[4].Value = 5;

dataGridViewСкалярныеОценки.Rows[3].Cells[5].Value = 3;

dataGridViewСкалярныеОценки.Rows.Add();

dataGridViewСкалярныеОценки.Rows[4].Cells[0].Value = "x5";

dataGridViewСкалярныеОценки.Rows[4].Cells[1].Value = 4;

dataGridViewСкалярныеОценки.Rows[4].Cells[2].Value = 2;

dataGridViewСкалярныеОценки.Rows[4].Cells[3].Value = 4;

dataGridViewСкалярныеОценки.Rows[4].Cells[4].Value = 5;

dataGridViewСкалярныеОценки.Rows[4].Cells[5].Value = 5;

dataGridViewСкалярныеОценки.Rows.Add();

dataGridViewСкалярныеОценки.Rows[5].Cells[0].Value = "x6";

dataGridViewСкалярныеОценки.Rows[5].Cells[1].Value = 3;

dataGridViewСкалярныеОценки.Rows[5].Cells[2].Value = 5;

dataGridViewСкалярныеОценки.Rows[5].Cells[3].Value = 3;

dataGridViewСкалярныеОценки.Rows[5].Cells[4].Value = 5;

dataGridViewСкалярныеОценки.Rows[5].Cells[5].Value = 3;

dataGridViewСкалярныеОценки.Rows.Add();

dataGridViewСкалярныеОценки.Rows[6].Cells[0].Value = "x7";

dataGridViewСкалярныеОценки.Rows[6].Cells[1].Value = 4;

dataGridViewСкалярныеОценки.Rows[6].Cells[2].Value = 5;

dataGridViewСкалярныеОценки.Rows[6].Cells[3].Value = 3;

dataGridViewСкалярныеОценки.Rows[6].Cells[4].Value = 4;

dataGridViewСкалярныеОценки.Rows[6].Cells[5].Value = 3;

dataGridViewВажности.Columns.Add("K1", "K1");

dataGridViewВажности.Columns.Add("K2", "K2");

dataGridViewВажности.Columns.Add("K3", "K3");

dataGridViewВажности.Columns.Add("K4", "K4");

dataGridViewВажности.Columns.Add("K5", "K5");

dataGridViewВажности.Rows.Add();

dataGridViewВажности.Rows[0].Cells[1].Value = 2;

dataGridViewВажности.Rows[0].Cells[2].Value = 1;

dataGridViewВажности.Rows[0].Cells[3].Value = 1;

dataGridViewВажности.Rows[0].Cells[4].Value = 1;

dataGridViewВажности.Rows[0].Cells[5].Value = 2;

}

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

{

dataGridViewСкалярныеОценки.Columns.Add("K" + dataGridViewСкалярныеОценки.Columns.Count, "K" + dataGridViewСкалярныеОценки.Columns.Count);

dataGridViewВажности.Columns.Add("K" + dataGridViewВажности.Columns.Count, "K" + dataGridViewВажности.Columns.Count);

}

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

{

dataGridViewСкалярныеОценки.Columns.Remove("K" + (dataGridViewСкалярныеОценки.Columns.Count - 1));

dataGridViewВажности.Columns.Remove("K" + (dataGridViewВажности.Columns.Count - 1));

}

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

{

SortedList<int, SortedList<int, double>> матрицаСкалярныхОценок = СчитываниеМатрицыСкалярныхОценок();

if (матрицаСкалярныхОценок == null)

{

return;

}

SortedList<int, double> векторВажностиКритериев = СчитываниеВектораВажностиКритериев();

if (векторВажностиКритериев == null)

{

return;

}

#region Вычисление коэфициентов относительной важности

SortedList<int, SortedList<int, double>> КоэффициентыОтносительнойВажностиКритериевQij =

ВычислениеМатрицыКоэфициентовОтносительнойВажностиQij(векторВажностиКритериев);

FormDataGridView formDataGridViewОтносительнаяВажностьКритериевQij = new FormDataGridView();

formDataGridViewОтносительнаяВажностьКритериевQij.Text = "Матрица коэффициентов относительной важности критериев";

SortedList<int, string> строкиQij = new SortedList<int, string>();

SortedList<int, string[]> столбцыQij = new SortedList<int, string[]>();

string[] нулевойСтолбецQij = { "Критерий", "Критерий" };

столбцыQij.Add(0, нулевойСтолбецQij);

foreach (int i in векторВажностиКритериев.Keys)

{

строкиQij.Add(i, "K" + (i + 1));

столбцыQij.Add(i + 1, new string[2]);

столбцыQij[i + 1][0] = "K" + (i + 1);

столбцыQij[i + 1][1] = "K" + (i + 1);

}

formDataGridViewОтносительнаяВажностьКритериевQij.Visible = true;

formDataGridViewОтносительнаяВажностьКритериевQij.DoubleЗаполнение(

КоэффициентыОтносительнойВажностиКритериевQij, строкиQij, столбцыQij);

#endregion

#region Модифицированние значений критериев

SortedList<int, SortedList<int, double>> модифицированныеЗначенияКритериев =

ВычислениеМатрицыВекторныхОценок(матрицаСкалярныхОценок,

КоэффициентыОтносительнойВажностиКритериевQij, векторВажностиКритериев);

FormDataGridView formDataGridViewМодифицированныеЗначенияКритериев = new FormDataGridView();

formDataGridViewМодифицированныеЗначенияКритериев.Text = "Модифизированные значения критериев";

formDataGridViewМодифицированныеЗначенияКритериев.Visible = true;

SortedList<int, string> строкиКмз = new SortedList<int, string>();

SortedList<int, string[]> столбцыКмз = new SortedList<int, string[]>();

string[] нулевойСтолбецКмз = new string[2];

нулевойСтолбецКмз[0] = "Варианты";

нулевойСтолбецКмз[1] = "Варианты";

столбцыКмз.Add(0, нулевойСтолбецКмз);

foreach (int i in векторВажностиКритериев.Keys)

{

столбцыКмз.Add(i + 1, new string[2]);

столбцыКмз[i + 1][0] = "K" + (i + 1);

столбцыКмз[i + 1][1] = "K" + (i + 1);

}

foreach (int i in модифицированныеЗначенияКритериев.Keys)

{

строкиКмз.Add(i, "x" + (i + 1));

}

formDataGridViewМодифицированныеЗначенияКритериев.Visible = true;

formDataGridViewМодифицированныеЗначенияКритериев.DoubleЗаполнение(

модифицированныеЗначенияКритериев, строкиКмз, столбцыКмз);

#endregion

#region Вычисление и вывод множества несравнимых решений

SortedList<int, SortedList<int, double>> множествоНесравнимыхРешений1 =

ВычислениеМатрицыМножестваНесравнимыхМеждуСобойРешенийX(модифицированныеЗначенияКритериев);

FormDataGridView formDataGridViewМножествоНесравнимыхРешений1 = new FormDataGridView();

formDataGridViewМножествоНесравнимыхРешений1.Text = "Множество несравнимых решений";

SortedList<int, string> строкиXнр1 = new SortedList<int, string>();

SortedList<int, string[]> столбцыXнр1 = new SortedList<int, string[]>();

string[] нулевойСтолбецXнр1 = { "Варианты", "Варианты" };

столбцыXнр1.Add(0, нулевойСтолбецXнр1);

foreach (int i in векторВажностиКритериев.Keys)

{

столбцыXнр1.Add(i + 1, new string[2]);

столбцыXнр1[i + 1][0] = "K" + (i + 1);

столбцыXнр1[i + 1][1] = "K" + (i + 1);

}

foreach (int i in множествоНесравнимыхРешений1.Keys)

{

строкиXнр1.Add(i, "x" + (i + 1));

}

formDataGridViewМножествоНесравнимыхРешений1.Visible = true;

formDataGridViewМножествоНесравнимыхРешений1.DoubleЗаполнение(

множествоНесравнимыхРешений1, строкиXнр1, столбцыXнр1);

#endregion

}

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

{

Dispose();

}

private void dataGridViewСкалярныеОценки\_RowsAdded(object sender, DataGridViewRowsAddedEventArgs e)

{

dataGridViewСкалярныеОценки.Rows[dataGridViewСкалярныеОценки.Rows.Count - 1].Cells[0].Value = "x" + dataGridViewСкалярныеОценки.Rows.Count;

}

public SortedList<int, SortedList<int, double>> СчитываниеМатрицыСкалярныхОценок()

{

SortedList<int, SortedList<int, double>> матрицаСкалярныхОценок = new SortedList<int, SortedList<int, double>>();

for (int i = 0; i < dataGridViewСкалярныеОценки.Rows.Count - 1; i++)

{

матрицаСкалярныхОценок.Add(i, new SortedList<int, double>());

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

{

double result = 0;

if (dataGridViewСкалярныеОценки.Rows[i].Cells[j].Value != null &&

double.TryParse(dataGridViewСкалярныеОценки.Rows[i].Cells[j].Value.ToString(), out result) == true)

{

матрицаСкалярныхОценок[i].Add(j - 1, result);

}

else

{

MessageBox.Show("Ошибка оценках");

return null;

}

}

}

return матрицаСкалярныхОценок;

}

public SortedList<int, double> СчитываниеВектораВажностиКритериев()

{

SortedList<int, double> векторВажностиКритериев = new SortedList<int, double>();

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

{

double result = 0;

if (dataGridViewВажности.Rows[0].Cells[j].Value != null &&

double.TryParse(dataGridViewВажности.Rows[0].Cells[j].Value.ToString(), out result) == true)

{

векторВажностиКритериев.Add(j - 1, result);

}

}

return векторВажностиКритериев;

}

public SortedList<int, SortedList<int, double>> ВычислениеМатрицыКоэфициентовОтносительнойВажностиQij(

SortedList<int, double> векторВажностиКритериев)

{

SortedList<int, SortedList<int, double>> матрицаКоэффициентовОтносительнойВажностиКритериевQij = new SortedList<int, SortedList<int, double>>();

foreach (int i in векторВажностиКритериев.Keys)

{

матрицаКоэффициентовОтносительнойВажностиКритериевQij.Add(i, new SortedList<int, double>());

foreach (int j in векторВажностиКритериев.Keys)

{

матрицаКоэффициентовОтносительнойВажностиКритериевQij[i].Add(j,

(double)векторВажностиКритериев[j] / (double)(векторВажностиКритериев[i] + векторВажностиКритериев[j]));

}

}

return матрицаКоэффициентовОтносительнойВажностиКритериевQij;

}

public SortedList<int, SortedList<int, double>> ВычислениеМатрицыМножестваНесравнимыхМеждуСобойРешенийX(

SortedList<int, SortedList<int, double>> матрицаСкалярныхОценок)

{

SortedList<int, SortedList<int, double>> матрицаМножестваНесравнимыхМеждуСобойРешенийX = new SortedList<int, SortedList<int, double>>();

foreach (int i in матрицаСкалярныхОценок.Keys)

{

матрицаМножестваНесравнимыхМеждуСобойРешенийX.Add(i, матрицаСкалярныхОценок[i]);

}

foreach (int k in матрицаСкалярныхОценок.Keys)

{

foreach (int i in матрицаСкалярныхОценок.Keys)

{

bool kяОценкаБольшеИлиРавнаiйОценки = true;

foreach (int j in матрицаСкалярныхОценок[i].Keys)

{

if (матрицаСкалярныхОценок[k][j] < матрицаСкалярныхОценок[i][j] && i != k)

{

kяОценкаБольшеИлиРавнаiйОценки = false;

}

}

bool однаjяkйОценкиБольшеОднойjйiйОценки = false;

if (kяОценкаБольшеИлиРавнаiйОценки == true)

{

foreach (int j in матрицаСкалярныхОценок[i].Keys)

{

if (матрицаСкалярныхОценок[k][j] > матрицаСкалярныхОценок[i][j] && i != k)

{

однаjяkйОценкиБольшеОднойjйiйОценки = true;

break;

}

}

}

if (kяОценкаБольшеИлиРавнаiйОценки == true && однаjяkйОценкиБольшеОднойjйiйОценки == true)

{

матрицаМножестваНесравнимыхМеждуСобойРешенийX.Remove(i);

}

}

}

return матрицаМножестваНесравнимыхМеждуСобойРешенийX;

}

public SortedList<int, SortedList<int, double>> ВычислениеМатрицыВекторныхОценок(

SortedList<int, SortedList<int, double>> матрицыМножестваНесравнимыхМеждуСобойРешенийX,

SortedList<int, SortedList<int, double>> матрицаКоэфициэнтовОтносительнойВажности,

SortedList<int, double> векторВажностиКритериев)

{

SortedList<int, SortedList<int, double>> матрицаВекторыхОценок = new SortedList<int, SortedList<int, double>>();

foreach (int i in матрицыМножестваНесравнимыхМеждуСобойРешенийX.Keys)

{

матрицаВекторыхОценок.Add(i, new SortedList<int, double>());

foreach (int j in матрицыМножестваНесравнимыхМеждуСобойРешенийX[i].Keys)

{

матрицаВекторыхОценок[i].Add(j, матрицыМножестваНесравнимыхМеждуСобойРешенийX[i][j]);

}

}

foreach (int k in матрицыМножестваНесравнимыхМеждуСобойРешенийX.Keys)

{

foreach (int i in векторВажностиКритериев.Keys)

{

foreach (int j in векторВажностиКритериев.Keys)

{

if (векторВажностиКритериев[j] > векторВажностиКритериев[i])

{

матрицаВекторыхОценок[k][j] = (double)((double)(матрицыМножестваНесравнимыхМеждуСобойРешенийX[k][i] \* матрицаКоэфициэнтовОтносительнойВажности[j][i]) +

(double)(матрицыМножестваНесравнимыхМеждуСобойРешенийX[k][j] \* (1 - матрицаКоэфициэнтовОтносительнойВажности[j][i])));

}

}

}

}

return матрицаВекторыхОценок;

}

}

}

1. ТЕСТОВЫЕ ПРИМЕРЫ

На рисунках 1-4 изображены тестовые примеры работы программы.

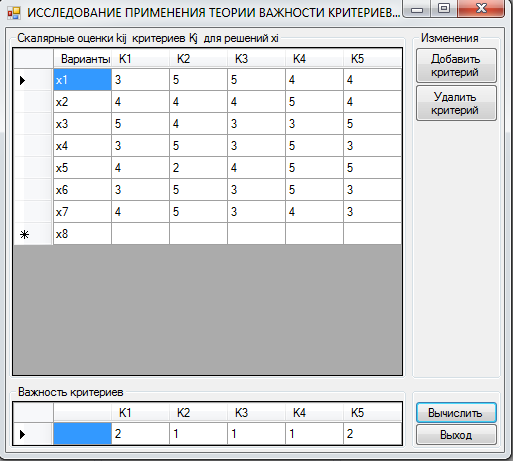


Рисунок 1 – Главное меню, задание значений и вывод результатов

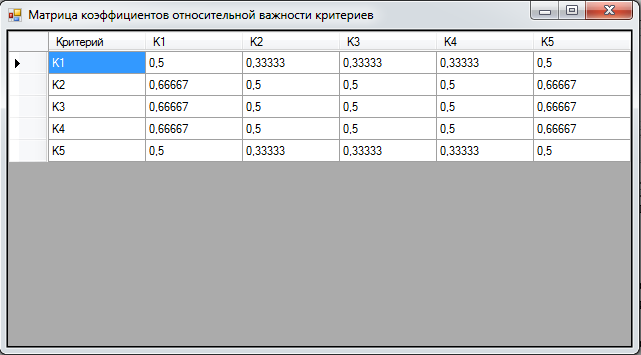


Рисунок 2 – Матрица коэффициентов относительной важности критериев

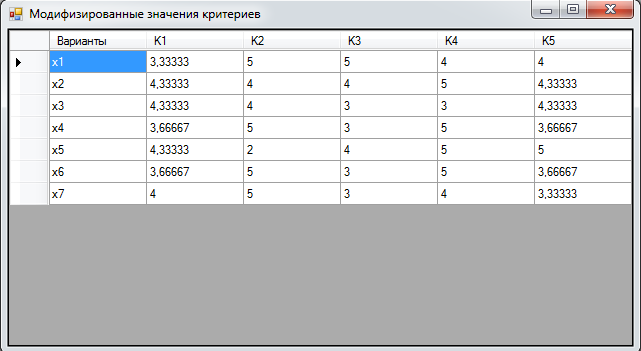


Рисунок 3 – Модифицированные значения критериев

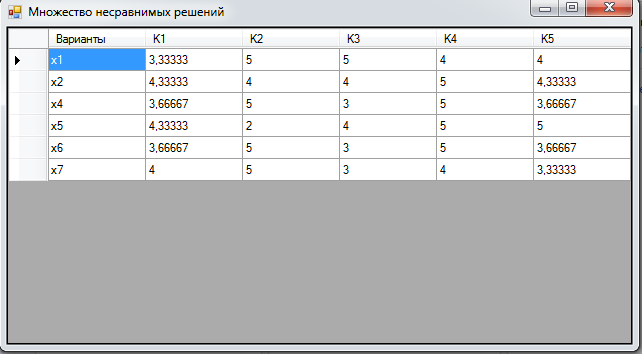


Рисунок 4 –Множество несравнимых решений

ВЫВОДЫ

В данной лабораторной работе были исследовано применение аппарата теории важности критериев при принятии решений по выбору альтернатив.