Министерство образования Республики Беларусь

Учреждение образования «Белорусский Государственный Университет Информатики и Радиоэлектроники»

Кафедра «Информатика»

Лабораторная работа №2

«Симплекс-метод для решения задачи линейного программирования в канонической форме»

**Выполнил:**

Волчецкий А. М, , гр. 253504

**Проверила:**

Саттарова П.С.

Минск 2015

**Задание:**

Дана матрица A, векторы c, b, начальное допустимое базисное решение xb, базисные переменные jb. Найти решение задачи линейного программирования в канонической форме симплекс-методом.

**Листинг:**

public virtual double[] Solve()

{

Report = "";

const int maxIterationsCount = 5000;

for (int iteration = 0; iteration < maxIterationsCount; iteration++)

{

Report += $"{iteration} итерация:\n\n";

Report += $"Матрица A:\n\n{Helper.ToPrint(matrixA)}\n";

Report += $"Базисный план x:\n\n{Helper.ToPrint(xBaseVector)}\n";

Step6CalculateB();

Report += $"Матрица B:\n\n{Helper.ToPrint(matrixB)}\n";

var estimations = Step1GetEstimations();

Report += "Оценки:\n\n";

foreach (var deltaI in estimations)

{

Report += $"delta{deltaI.Key} = {deltaI.Value}\n";

}

if (Step2IsPlanOptimal(estimations))

{

var x = xBaseVector;

Report += "\nПлан оптимален";

return x.ToArray();

}

var step3Res = Step3GetZAndJ0(estimations);

if (step3Res == null)

{

return null;

}

var z = step3Res.Item1;

Report += $"\nZ vector:\n\n{Helper.ToPrint(z)}";

int j0 = step3Res.Item2;

Report += $"\nJ0 = {j0}";

var step4Res = Step4GetNewBaseIndexAndMinimum(z);

int s = step4Res.Item1;

double teta0 = step4Res.Item2;

Report += $"\ns = {s}";

Report += $"\nteta0 = {teta0}";

Step5RecalculateBasePlanAndBasis(j0, z, s, teta0);

Report += $"\n\nНовый базисный план x:\n\n{Helper.ToPrint(xBaseVector)}";

string baseInd = jB.Aggregate("", (current, baseIndex) => current + (baseIndex + " "));

Report += $"\nJb = {baseInd}\n\n";

}

return null;

}

//возвращает оценки

private Dictionary<int, double> Step1GetEstimations()

{

//cB = (cj, j принадлежит jB)

var cB = new DenseVector(jB.Count);

for(int i = 0; i < jB.Count; i++)

{

cB[i] = vectorC[jB[i]];

}

var u = cB \* matrixB;

Report += $"Вектор потенциалов u:\n\n{Helper.ToPrint(u)}\n";

//deltaJ = u \* Aj - cj, j принадл. J\jb

var estimations = new Dictionary<int, double>();

for (var j = 0; j < vectorC.Count; j++)

{

if (jB.Contains(j) == false)

{

var deltaJ = u \* matrixA.Column(j) - vectorC[j];

estimations.Add(j, deltaJ);

}

}

return estimations;

}

private bool Step2IsPlanOptimal(Dictionary<int, double> estimations)

{

//deltaJ >= 0, j прин. J\jB => план оптимальный

return estimations.All(delta => delta.Value >= 0);

}

private Tuple<DenseVector, int> Step3GetZAndJ0(Dictionary<int, double> estimations)

{

foreach (var delta in estimations)

{

//найдем j0 для кот. deltaJ0 < 0

if (delta.Value < 0)

{

int j0 = delta.Key;

//найдем z = B \* A(j0)

var z = matrixB \* (DenseVector)matrixA.Column(j0);

//если хоть один элемент вектора z положительный,

//то задача пока имеет решения

foreach (var zi in z)

{

if (zi > 0)

{

return new Tuple<DenseVector, int>(z, j0);

}

}

}

}

return null;

}

private Tuple<int, double> Step4GetNewBaseIndexAndMinimum(DenseVector z)

{

//teta0 = x[j[i]] / z[i], z[i] > 0

//s - индекс при котором мы получаем teta0

double min = double.MaxValue;

var s = -1;

for (var i = 0; i < z.Count; i++)

{

if (z[i] > 0)

{

var jI = jB[i];

var teta0 = xBaseVector[jI] / z[i];

if (teta0 < min)

{

min = teta0;

s = i;

}

}

}

if (s == -1)

{

throw new Exception("Step4: s not founded");

}

return new Tuple<int, double>(s, min);

}

private void Step5RecalculateBasePlanAndBasis(int j0, DenseVector z, int s, double teta0)

{

//xj = 0, j принадлежит Jн\J0

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

{

if (!jB.Contains(j) && j != j0)

{

xBaseVector[j] = 0;

}

}

xBaseVector[j0] = teta0;

//x[Ji] = x[Ji] - teta0 \* zi

for (int i = 0; i < z.Count; i++)

{

int jI = jB[i];

xBaseVector[jI] = jI == j0 ? 0 : xBaseVector[jI] - teta0 \* z[i];

}

//Jб = Jб\Js

jB[s] = j0;

jB.Sort();

}

protected void Step6CalculateB()

{

//aB - базисная матрица

var aB = new DenseMatrix(matrixA.RowCount, jB.Count);

//aB = (Aj, j принадлежит jB)

for (int i = 0; i < jB.Count; i++)

{

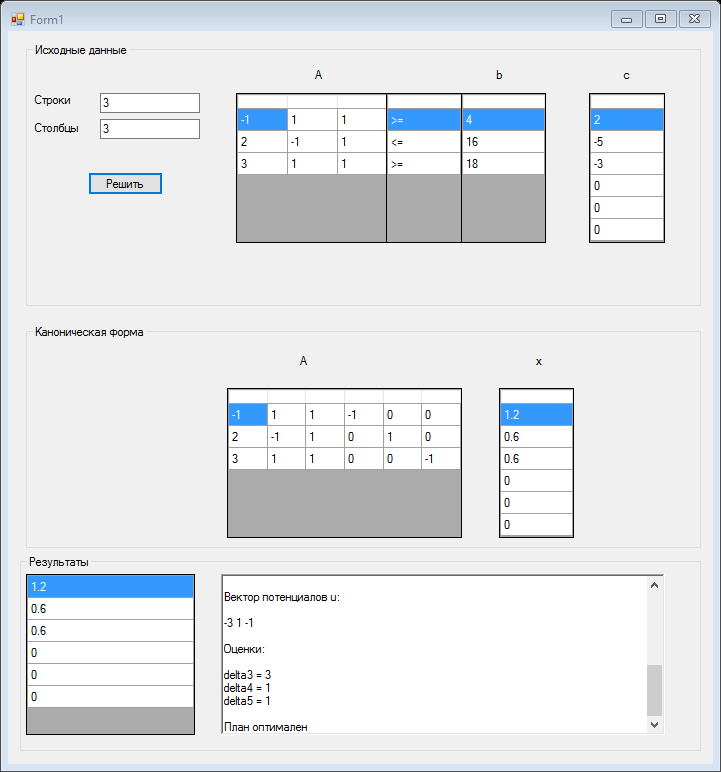
aB.SetColumn(i, matrixA.Column(jB[i]));

}

matrixB = (DenseMatrix)aB.Inverse();

}

**Пример**

****