Министерство образования Республики Беларусь

Учреждение образования «Белорусский Государственный Университет Информатики и Радиоэлектроники»

Кафедра «Информатика»

Лабораторная работа №3

«Двойственный симплекс метод»

**Выполнил:**

Волчецкий А. М., гр. 253504

**Проверила:**

Саттарова П.С.

Минск 2015

**Задание:**

Решить задачу линейного программирования симплекс методом.

**Листинг:**

**public new Tuple<double[], double[]> Solve()**

**{**

**Report = "";**

**DenseVector pseudoPlan = null;**

**Step6CalculateB();**

**y = GetFirstBasePlan();**

**var iterationsCount = 0;**

**while (iterationsCount++ < MaxIterationsCount)**

**{**

**Report += string.Format("{0}-я итерация\n\n", iterationsCount);**

**Report += string.Format("Матрица B:\n{0}\n", Helper.ToPrint(matrixB));**

**Report += string.Format("y: {0}\n", Helper.ToPrint(y));**

**Report += string.Format("Jb: {0}\n\n", jB.Aggregate("", (acc, ind) => acc + ind + ", "));**

**Step1CalculatePseudoPlan();**

**Report += string.Format("Псевдоплан: {0}\n", Helper.ToPrint(pseudoPlanB));**

**if (Step2CheckForOptimum())**

**{**

**pseudoPlan = BuildFullPlan(pseudoPlanB);**

**Report += string.Format("Решение:\nЦелевая функция: {0}", CalculateTargetFunc(vectorC, pseudoPlan));**

**break;**

**}**

**int s;**

**DenseVector deltaY;**

**Dictionary<int, double> estimations;**

**if (!Step3CheckForCompatibility(out s, out deltaY, out estimations))**

**{**

**Report += "Решений нет";**

**break;**

**}**

**var step4Res = Step4GetMinimum(estimations);**

**double sigma0 = step4Res.Item1;**

**int j0 = step4Res.Item2;**

**Report += string.Format("s: {0}\n", s);**

**Report += string.Format("\ndeltaY: {0}", Helper.ToPrint(deltaY));**

**Report += string.Format("\nsigma0: {0}\n", sigma0);**

**Report += string.Format("\nj0: {0}\n\n", j0);**

**Step5RecalculatePlan(s, deltaY, sigma0, j0);**

**Step6CalculateB();**

**}**

**return new Tuple<double[], double[]>(pseudoPlan?.ToArray(), y?.ToArray());**

**}**

**private double CalculateTargetFunc(DenseVector c, DenseVector x)**

**{**

**return c.Select((t, i) => t \* x[i]).Sum();**

**}**

**#endregion**

**#region Private methods**

**private DenseVector GetFirstBasePlan()**

**{**

**var cB = new DenseVector(jB.Count);**

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

**{**

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

**}**

**return cB \* matrixB;**

**}**

**private DenseVector BuildFullPlan(DenseVector basePlan)**

**{**

**var fullPlan = new DenseVector(matrixA.ColumnCount);**

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

**{**

**fullPlan[jB[i]] = basePlan[i];**

**}**

**return fullPlan;**

**}**

**private void Step1CalculatePseudoPlan()**

**{**

**var pseudoPlanB1 = matrixB \* vectorB;**

**pseudoPlanB = new DenseVector(matrixA.ColumnCount);**

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

**{**

**pseudoPlanB[jB[i]] = pseudoPlanB1[i];**

**}**

**}**

**private bool Step2CheckForOptimum()**

**{**

**return pseudoPlanB.All(t => t >= 0);**

**}**

**private bool Step3CheckForCompatibility(out int s, out DenseVector deltaY, out Dictionary<int, double> estimations)**

**{**

**s = -1;**

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

**{**

**if (pseudoPlanB[i] < 0)**

**{**

**s = i;**

**break;**

**}**

**}**

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

**deltaY = null;**

**var isCompatible = false;**

**if (s >= 0)**

**{**

**//единичный вектор**

**var eS = new DenseVector(jB.Count) {[s] = 1 };**

**deltaY = eS \* matrixB;**

**for (var j = 0; j < matrixA.ColumnCount; j++)**

**{**

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

**{**

**var uJ = deltaY \* matrixA.Column(j);**

**estimations.Add(j, uJ);**

**}**

**}**

**if (estimations.Values.Any(uJ => uJ < 0))**

**{**

**isCompatible = true;**

**}**

**}**

**return isCompatible;**

**}**

**//return (sigma0, j0)**

**private Tuple<double, int> Step4GetMinimum(Dictionary<int, double> estimations)**

**{**

**var sigma0 = double.MaxValue;**

**var j0 = -1;**

**foreach (KeyValuePair<int, double> estimation in estimations)**

**{**

**int j = estimation.Key;**

**double uJ = estimation.Value;**

**if (uJ < 0)**

**{**

**var sigma = (vectorC[j] - matrixA.Column(j) \* y) / uJ;**

**if (sigma < sigma0)**

**{**

**sigma0 = sigma;**

**j0 = j;**

**}**

**}**

**}**

**if (j0 < 0)**

**{**

**throw new Exception("Ошибка в 4 шаге. Невозможно найти j0");**

**}**

**return new Tuple<double, int>(sigma0, j0);**

**}**

**private void Step5RecalculatePlan(int s, DenseVector deltaY, double sigma0, int j0)**

**{**

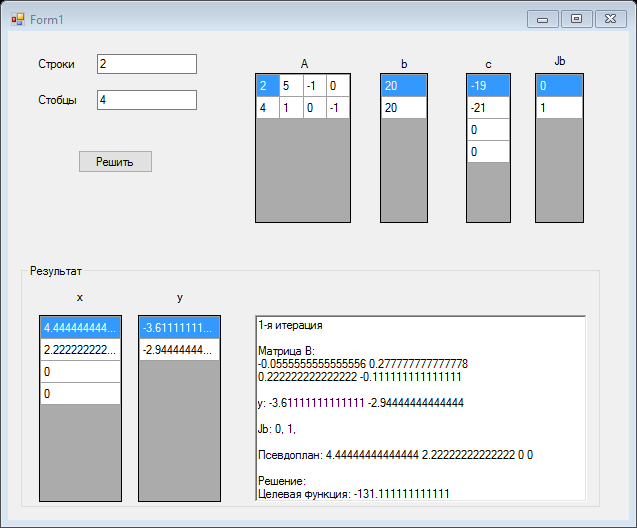
**y = y + sigma0 \* deltaY;**

**jB[s] = j0;**

**jB.Sort();**

**}**

**Пример:**

****