Одеський національний університет імені І. І. Мечникова

(повне найменування вищого навчального закладу)

Інститут математики, економіки та механіки

(повне найменування інституту, назва факультету (відділення))

Кафедра Оптимального керування та економічної кібернетики

(повна назва кафедри (предметної, циклової комісії))

**Звіт по практиці**

Бакалавр

(освітньо-кваліфікаційний рівень)

на тему «Багаторiвнева оптимiзацiя складних систем»

Виконав: студентка ІV курсу,

денної форми навчання

напряму підготовки / спеціальності

6.040301,Прикладна математика

(шифр і назва напряму підготовки, спеціальності)

Бондаренко О.М.

(прізвище та ініціали)

Керівник Єфімова Г.О.

(прізвище та ініціали)

|  |  |
| --- | --- |
|  |  |
|  |  |

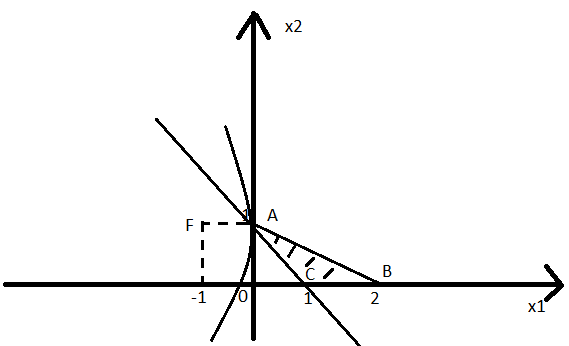
Одеса – 2014 року

С 19 мая по 15 июня 2014 года я проходил производственную практику на кафедре Оптимального управления и экономической кибернетики Одесского национального университета им. И.И.Мечникова.

Во время практики я выполнил несколько поставленных передо мною заданий: первое - набор методический указаний по теме: «Квадратичное программирование», второе - подготовка презентации о задаче на оптимизацию функции с двумя параметрами и третье – работа над практической частью диплома, а именно: программная реализация алгоритма оптимизации двухуровневой иерархической системы.

Квадратичное программирование (КП)

Одним из частных видов задачи выпуклого программирования является задача, в которой целевая функция содержит линейную и квадратичную формы, а ограничения линейны. В качестве основной в КП рассматривается задача минимизации функции  
  (1)  
при ограничениях  
 или  (2)  
    
Здесь  - симметричная и неотрицательно определенная матрица размерности , А – матрица размерности , , . В силу сделанных предположений целевая функция f(x) выпукла, допустимое множество  выпукло, поэтому задача (1)-(2) – задача выпуклого программирования.  
Функция Лагранжа задачи (1)-(2) имеет вид   
 (3)  
Необходимыми и достаточными условиями оптимальности решения являются условия Куна-Такера: вектор  является оптимальным планом задачи (1)-(2) тогда и только тогда, когда существует вектор такой, что пара  представляют собой седловую точку для , причем для них выполняется условие дополняющей нежесткости   
Так как в т. достигает минимума по , то   
  
  
Введем дополнительное требование:  
  
Тогда функциястрого выпукла вниз по x при любом , т.е. достигает своего минимума в единственной точке , которая является решением системы   
  
отсюда   
поэтому  (4)  
Подставляя (4) в (3), получим   
 (5)  
Обозначим   
  
Очевидно, что является выпуклой вверх функцией и поэтому имеет единственный максимум. Таким образом, для нахождения  нужно решить вспомогательную задачу   
 (6)  
  
Следовательно решение исходной задачи (1) – (2) состоит из двух этапов:  
1) решение задачи безусловной оптимизации в неотрицательной области (6)(получаем вектор )  
2) вычисление  по формулам(4) при   
  
Замечание Для решения задачи(6) удобно использовать метод сопряженных градиентов, т.к. известно, при оптимизации квадратичной функции без ограничений этот метод даёт решение за некоторое число итерация, не превышающее количество переменных задачи.  
  
Пример   
  
  
Решение Здесь , , ,  
  
  
  
где B – const.  
Решим аналитически вспомогательную задачу   
  
Здесь   
   
Значения и  найдем из системы уравнений   
   
имеем   
Так как ****, то вектор  является решением вспомогательной задачи, которая является задачей выпуклого программирования со строго выпуклой целевой функцией.

Восстановим вектор   
  
Проиллюстрируем решение графически   
Допустим множество Х – это треугольник ABC, линии уровня целевой функции – концентрические окружности в т.F(1,1). Функция f(x) на множестве Х достигает минимума в т.А(0,1)   


Квадратичный симплекс-метод (КСМ)

Свойства задачи (1) – (2) дают возможность модифицировать для неё симплекс-метод в алгоритм квадратичного программирования, который обеспечивает сходимость к оптимальному решению за ограниченное число итераций.  
В силу выпуклости задачи (1) – (2) по теореме Куна-Такера критерием решения задачи будет выполнение следующих условий:  
  
  
  
  
Сведем неравенства 1) и 2) к равенства с помощью n-мерной переменной  и m-мерной переменной . Тогда необходимые и достаточные условия можно сформулировать следующим образом: вектор  является оптимальным решением задачи (1) – (2) тогда и только тогда, когда существуют такие m-мерные вектора  и n- мерный вектор , которые удовлетворяют системе:

   
  
Условия (8), (9) являются системой n+m линейных уравнений с 2(n+m) неизвестными , а условия (10), (11) накладывают ограничения на переменные : если , то  и наоборот; если , то , то есть -тое основное ограничение задачи (1) – (2) активно, и наоборот, то есть одновременно не могут быть ненулевыми(базисными) и , а также  и .  
Благодаря этим условиям решение системы (8) – (9) должно быть допустимым базисным и для его нахождения можно использовать одну из модификаций симплекс- метода(например, М-метод).

Пример  (12)  
   
    
  
Решение Здесь , . Нетрудно убедиться, что , поэтому функция  выпукла (вниз) и задача (12) является задачей квадратичного программирования.  
Функция Лагранжа имеет вид:  
  
  
Введем систему к каноническому виду, вводя дополнительные переменные   
Получим систему   
 (13)  
  
и условия дополняющей нежесткости  
 (14)  
Нужно найти допустимое базисное решение системы (13), который бы удовлетворял условиям (14).

Для этого используем метод искусственных переменных. Достаточно ввести искусственные переменные z1 и z2, после чего получим следующую задачу линейного программирования:  
  
  
Результаты итераций – в таблице 1.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | с | 0 | 0 | 0 | 0 | 0 | 0 | -M | -M |  |
| b |  |  |  |  |  |  |  |  |
| -M |  | 4 | 4 | 2 | 1 | -1 | 0 | 0 | 1 | 0 | 4/4 |
| -M |  | 6 | 2 | 4 | 2 | 0 | -1 | 0 | 0 | 1 | 6/2 |
| 0 |  | 2 | 1 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 2/1 |
|  |  | -10M | -6M | -6M | -3M | M | M | 0 | 0 | 0 |  |
| 0 |  | 1 | 1 | ½ | ¼ | -1/4 | 0 | 0 | ¼ | 0 | 2 |
| -M |  | 4 | 0 | 3 | 3/2 | ½ | -1 | 0 | -1/2 | 1 | 4/3 |
| 0 |  | 1 | 0 | 3/2 | -1/4 | ¼ | 0 | 1 | -1/4 | 0 |  |
|  |  | -4M | 0 | -3M | -3/2M | -1/2M | M | 0 | 3/2M | 0 |  |
| 0 |  | 2/3 | 1 | 0 | 1/3 | -1/3 | 0 | -1/3 |  | 0 | 2 |
| -M |  | 2 | 0 | 0 | 2 | 0 | -1 | -2 |  | 1 | 1 |
| 0 |  | 2/3 | 0 | 1 | -1/6 | 1/5 | 0 | 2/3 |  | 0 | - |
|  |  | -2M | 0 | 0 | -2M | 0 | M | 2M |  | 0 |  |
| 0 |  | 1/3 | 1 | 0 | 0 | -1/3 | 1/6 | 0 |  |  |  |
| 0 |  | 1 | 0 | 0 | 1 | 0 | -1/2 | -1 |  |  |  |
| 0 |  | 5/6 | 0 | 1 | 0 | 1/6 | -1/12 | ½ |  |  |  |
|  |  | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |

Из последней симплекс – таблицы   
Проверим условия дополняющей нежесткости:  
  
Следовательно, решением данного примера(12) является вектор   
  
  
  
  
 Задания для самостоятельной работы  
  
Вариант1. (N – номер студента по списку группы)  
1) Решить задачу квадратичного программирования  
  
при ограничениях   
  
Решение проиллюстрировать графически.  
2) Решить задачу квадратичного программирования  
  
при ограничениях   
  
с помощью квадратичного симплекс – метода  
Вариант 2. (N – номер студента по списку группы)  
1) Решить задачу квадратичного программирования  
  
при ограничениях   
  
Решение проиллюстрировать графически.  
2) Решить задачу квадратичного программирования  
  
при ограничениях   
  
с помощью квадратичного симплекс – метода  
  
  
  
 Литература  
Зайченко Ю.П. Дослiдження операцiй – К.: ВIПОЛ, 2001

Зайченко Ю.П., Шумилова С.А. Исследование операций: Сб. задач. – К.: Вища шк., 1990

Дякон В.М., Ковальов Л.Е. Матиматичне програмування. – К.: Вид. Європейського унiверситету, 2007

Кузнецов А.В., Холод Н.И. Математическое программирование – Минск: Вышейшая шк., 1984

Бейко И.В., Бублик Б.Н., Зинько П.Н. Методы и алгоритмы решения задач оптимизации – К.: Вища шк., 1983

Найти значение параметра а, при котором достигает максимума функция  

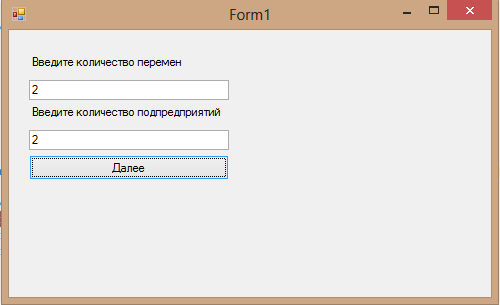
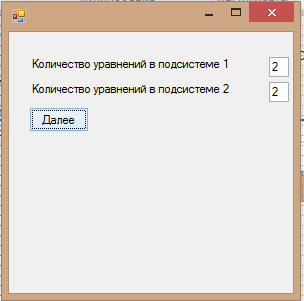
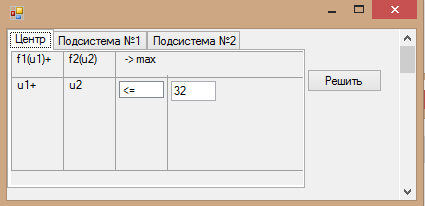
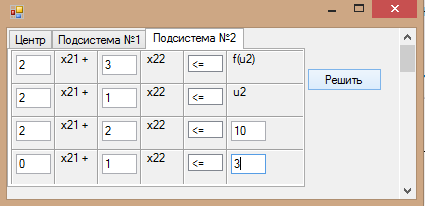

где- корни квадратного уравнения  
  
Решение   
По теореме Виета имеем 

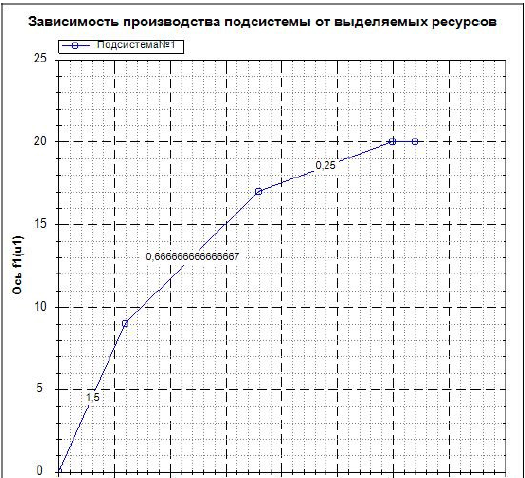
отсюда   
и тогда  Т.к. , a  убывает с возрастанием , то функция  достигает своего максимума при  и 

Замечание

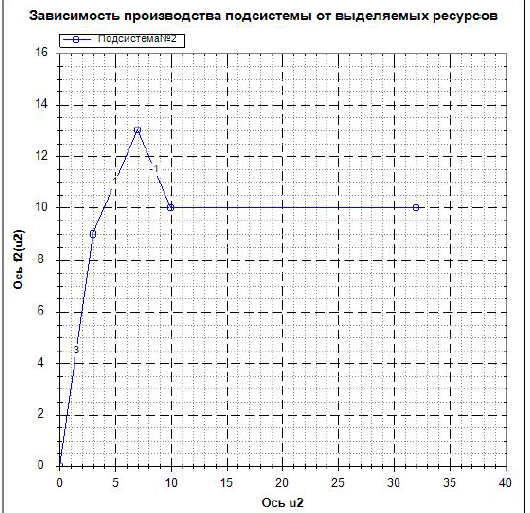
Применение теоремы Виета позволило в данном случае упростить задачу и даже избежать использования дифференциального исчисления для поиска  

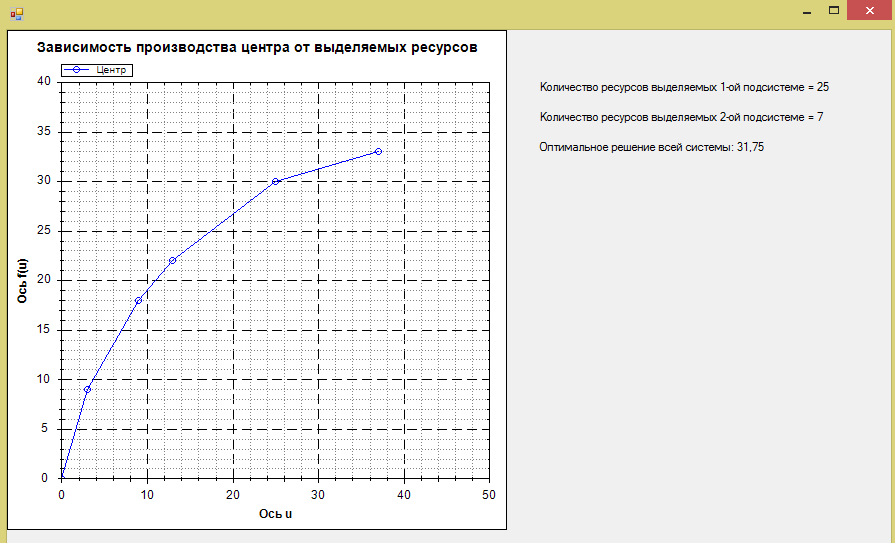

Тема моей дипломной работы «Многомерная оптимизация сложных систем».

1. Пользователю предоставляется право выбора количества переменных  
   в системе, количества подсистем, количества уравнений в каждой подсистеме.  
   
2. Затем пользователь заполняет подсистемы и количество ресурса, который имеется у центра.  
     
   
3. Для каждой из подсистем решаем оптимизационную параметрическую задачу относительно . Находим точки, в которых функция меняет свой вид. Для двумерного случая решаем задачу графическим методом оптимизации. Это будут точки пересечение прямых и точки пересечения прямых с осями координат.  
   O(0,0), A(3,4), B(3,0), C(0,10) – для первой подсистемы.  
   O(0,0), A(2,3), B(0,3), C(5,0) – для второй подсистемы.
4. Для каждой из точек находим соответсвующие им  и 

Для второй подсистемы:  
   
****Для первой подсистемы:  
  
  
  
  
Функция эффективности для первой подсистемы:   


Функция эффективности второй подсистемы:

Обобщенная функция имеет вид:  
  


Листинг кода:

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;

using System.Collections;

using System.Windows;

using ZedGraph;

using System.IO;

namespace MultilLeveloptimization

{

public partial class Form1 : Form

{

TabControl tabcontrol;

VScrollBar vScrollBar1;

Form Form2;

Form Form3;

Form Form4;

Form Form5;

TextBox variables = new TextBox();

System.Windows.Forms.Label amount\_variables = new System.Windows.Forms.Label();

TextBox systems = new TextBox();

Button next = new Button();

System.Windows.Forms.Label amount\_systems = new System.Windows.Forms.Label();

System.Windows.Forms.Label[] all\_labels;

TextBox[] all\_textboxs;

TextBox boxdownload = new TextBox();

Panel1 panel1;

Panel1[] paneles;

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

HelpProvider hp;

Button solveButton = new Button();

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

Button open = new Button();

open.Click += open\_Click;

List<Control> form1var = new List<Control>();

form1var.Add(this.amount\_variables);

form1var.Add(this.variables);

form1var.Add(this.amount\_systems);

form1var.Add(this.systems);

form1var.Add(this.next);

//form1var.Add(open);

//form1var.Add(boxdownload);

for (int i = 0; i < form1var.Count; i++)

{

form1var[i].Location = new Point(20, (i + 1) \* 25);

form1var[i].Visible = true;

form1var[i].Size = new Size(200, 25);

this.Controls.Add(form1var[i]);

form1var[i].Text = "1";

}

open.Text = "Загрузить данные";

this.amount\_variables.Text = "Введите количество перемен";

this.amount\_systems.Text = "Введите количество подпредприятий";

this.next.Text = "Далее";

this.next.Click += next\_Click;

}

void open\_Click(object sender, EventArgs e)

{

LoadFile();

}

void next\_Click(object sender, EventArgs e)

{

Second();

}

void Second()

{

Form2 = new Form();

this.Visible = false;

Form2.Visible = true;

Form4 = new Form();

int n = Convert.ToInt32(this.systems.Text);

all\_labels = new System.Windows.Forms.Label[n];

all\_textboxs = new TextBox[n];

string s;

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

{

s = "Количество уравнений в подсистеме" + " " + Convert.ToString(i + 1);

all\_labels[i] = new System.Windows.Forms.Label();

all\_textboxs[i] = new TextBox();

all\_labels[i].Text = Convert.ToString(s);

all\_labels[i].Visible = true;

all\_textboxs[i].Visible = true;

all\_textboxs[i].Text = "1";

all\_labels[i].Location = new Point(20, (1 + i) \* 25);

all\_labels[i].Size = new Size(225, 25);

Form2.Controls.Add(all\_labels[i]);

all\_textboxs[i].Location = new Point(260, (1 + i) \* 25);

all\_textboxs[i].Size = new Size(20, 25);

Form2.Controls.Add(all\_textboxs[i]);

}

Button second\_Next = new Button();

second\_Next.Click += second\_Next\_Click;

second\_Next.Location = new Point(20, (n + 1) \* 25);

second\_Next.Size = new Size(60, 25);

second\_Next.Text = "Далее";

second\_Next.Visible = true;

Form2.Controls.Add(second\_Next);

Form2.FormClosed += Form2\_FormClosed;

}

void Form2\_FormClosed(object sender, FormClosedEventArgs e)

{

this.Close();

}

void second\_Next\_Click(object sender, EventArgs e)

{

Third();

Form4.Visible = false;

}

void Third()

{

Form3 = new Form();

Form2.Visible = false;

Form3.Visible = true;

vScrollBar1 = new VScrollBar();

vScrollBar1.Dock = System.Windows.Forms.DockStyle.Right;

vScrollBar1.Location = new System.Drawing.Point(214, 0);

vScrollBar1.Maximum = 50;

vScrollBar1.Scroll += vScrollBar1\_Scroll;

//Form2.Visible = false;

int n = Convert.ToInt32(systems.Text);

tabcontrol = new TabControl();

TabPage[] tabpages = new TabPage[n];

tabcontrol.Size = new Size(50 \* (Convert.ToInt32(variables.Text) \* 2 + 2), 400);

paneles = new Panel1[n];

TabPage center = new TabPage();

center.Text = "Центр";

center.BorderStyle = BorderStyle.None;

center.Size = new Size(50 \* (Convert.ToInt32(variables.Text) \* 2 + 2), 40 \* 2);

panel1 = new Panel1(center, Convert.ToInt32(systems.Text));

tabcontrol.Controls.Add(center);

int maxh = 40 \* 2;

int maxw = 50 \* (Convert.ToInt32(variables.Text) \* 2 + 2);

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

{

tabpages[i] = new TabPage();

tabpages[i].Size = new Size(50 \* (Convert.ToInt32(variables.Text) \* 2 + 2), 40 \* (2 + Convert.ToInt32(all\_textboxs[i].Text)));

paneles[i] = new Panel1(tabpages[i], Convert.ToInt32(variables.Text), Convert.ToInt32(all\_textboxs[i].Text), i + 1);

tabpages[i].Text = "Подсистема №" + (i + 1);

tabcontrol.Controls.Add(tabpages[i]);

if ((50 \* (Convert.ToInt32(variables.Text) \* 2 + 2)) > maxw)

maxw = 50 \* (Convert.ToInt32(variables.Text) \* 2 + 2);

if (40 \* (2 + Convert.ToInt32(all\_textboxs[i].Text)) > maxh)

maxh = 40 \* (2 + Convert.ToInt32(all\_textboxs[i].Text));

}

tabcontrol.Size = new Size(maxw, maxh);

Form3.Controls.Add(vScrollBar1);

Form3.Controls.Add(tabcontrol);

Form3.FormClosed += Form2\_FormClosed;

solveButton = new Button();

solveButton.Text = "Решить";

solveButton.Location = new Point(tabcontrol.Location.X + tabcontrol.Size.Width, 40);

solveButton.Click += solveButton\_Click;

Form3.Controls.Add(solveButton);

Form3.Size = new Size(tabcontrol.Size.Width + solveButton.Size.Width + 50, tabcontrol.Size.Height + 50);

}

void solveButton\_Click(object sender, EventArgs e)

{

Forth\_Button();

}

void vScrollBar1\_Scroll(object sender, ScrollEventArgs e)

{

tabcontrol.Location = new Point(tabcontrol.Location.X, -vScrollBar1.Value);

}

void Forth\_Button()

{

int n = Convert.ToInt32(systems.Text);

TabControl tabcontrolsgraff = new TabControl();

TabPage[] tabpages = new TabPage[n];

try

{

for (int i = 0; i < tabcontrol.TabCount - 1; i++)

{

if (Convert.ToInt32(variables.Text) > 2)

{

paneles[i].reduce(Convert.ToInt32(variables.Text));

}

paneles[i].Koeff(Convert.ToInt32(all\_textboxs[i].Text), Convert.ToInt32(variables.Text));

paneles[i].Function();

tabpages[i] = new TabPage();

tabpages[i].Size = new Size(550, 550);

tabpages[i].Text = "Подсистема №" + (i + 1);

paneles[i].DrawGraph(tabpages[i], i + 1, Convert.ToDouble(panel1.getlimit().Text));

tabcontrolsgraff.Controls.Add(tabpages[i]);

}

tabcontrolsgraff.Size = new Size(570, 570);

Form4 = new Form();

Form4.Visible = true;

Form3.Visible = false;

Form4.Controls.Add(tabcontrolsgraff);

Form4.FormClosed += Form2\_FormClosed;

Form4.Size = new Size(700, 600);

Button form4button = new Button();

form4button.Text = "Обобщить системы";

Form4.Controls.Add(form4button);

int w = tabcontrolsgraff.Size.Width + 20;

int h = 40;

form4button.Location = new Point(w, h);

form4button.Size = new Size(100, 25);

form4button.Click += form4button\_Click;

}

catch (DivideByZeroException)

{

hp = new HelpProvider();

Form4 = new Form();

Form4.Visible = true;

Form3.Visible = false;

Form4.FormClosed += Form2\_FormClosed;

Button form4button = new Button();

Form4.Size = new Size(250, 100);

form4button.Text = "Введите корректные данные";

Form4.Controls.Add(form4button);

form4button.Size = new Size(250, 25);

form4button.Click += second\_Next\_Click;

}

}

void form4button\_Click(object sender, EventArgs e)

{

BigFunction();

}

void BigFunction()

{

double max = 0;

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

{

max = Math.Max(max, paneles[i].varU.Count);

}

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

first.Add(0);

first.Add(0);

for (int i = 1; i < max; i++)

{

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

for (int j = 0; j < paneles.Length; j++)

{

if (i == 0)

{

k.Add(0);

}

else

{

try

{

if (paneles[j].funcKoeff[i - 1][0] >= 0)

{

k.Add(paneles[j].varU[i]);

}

else

k.Add(paneles[j].varU[i - 1]);

}

catch (IndexOutOfRangeException)

{

k.Add(paneles[j].varU[paneles[i].varU.Count - 1]);

}

catch (ArgumentOutOfRangeException)

{

k.Add(paneles[j].varU[paneles[i].varU.Count - 1]);

}

}

}

k.Sort();

for (int j = 0; j < paneles.Length; j++)

{

first.Add(k[j]);

}

}

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

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

for (int i = 0; i < first.Count - paneles.Length + 1; i++)

{

double d = 0;

for (int j = i; j < i + paneles.Length; j++)

{

d += first[j];

}

result.Add(d);

}

result = result.Distinct().ToList();

funcResult.Add(0);

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

{

funcResult.Add(Maximus(result[i - 1], result[i]));

}

double Elim;

int index1 = 0, index2 = 0;

double z = Convert.ToDouble(panel1.getlimit().Text);

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

{

if (result[i - 1] <= z && result[i] >= z)

{

index1 = i - 1;

index2 = i;

}

}

Elim = (funcResult[index2] - funcResult[index1]) / (result[index2] - result[index1]);

double Fopt;

Fopt = funcResult[index1] + Elim \* (z - result[index1]);

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

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

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

{

int maxx = 0;

int j;

double dd = 0;

for (j = 0; j < paneles[i].funcKoeff.Count - 1; j++)

{

if (paneles[i].funcKoeff[j][0] >= Elim)

{

dd = paneles[i].funcKoeff[j][0];

maxx = j;

}

}

Ekoeff.Add(dd);

uopt.Add(paneles[i].varU[maxx + 1]);

}

Form5 = new Form();

Form5.Visible = true;

Form4.Visible = false;

Form5.FormClosed += Form2\_FormClosed;

Form5.Size = new Size(900, 600);

Draw(Form5, result, funcResult);

List<System.Windows.Forms.Label> last = new List<System.Windows.Forms.Label>();

if (uopt.Sum() <= z)

{

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

{

System.Windows.Forms.Label labelr = new System.Windows.Forms.Label();

labelr.Text = "Количество ресурсов выделяемых " + (i + 1) + "-ой подсистеме = " + uopt[i];

last.Add(labelr);

}

System.Windows.Forms.Label label = new System.Windows.Forms.Label();

label.Text = "Оптимальное решение всей системы: " + Fopt;

last.Add(label);

}

else

{

int min = 0;

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

{

if (Ekoeff[i - 1] > Ekoeff[i])

{

min = i;

}

}

uopt[min] = uopt[min] - (uopt.Sum() - z);

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

{

System.Windows.Forms.Label labelr = new System.Windows.Forms.Label();

labelr.Text = "Количество ресурсов выделяемых " + (i + 1) + "-ой подсистеме = " + uopt[i];

last.Add(labelr);

}

System.Windows.Forms.Label label = new System.Windows.Forms.Label();

label.Text = "Оптимальное решение всей системы: " + Fopt;

last.Add(label);

}

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

{

Form5.Controls.Add(last[i]);

last[i].Size = new Size(400, 25);

last[i].Visible = true;

last[i].Location = new Point(530, 50 + 30 \* i);

}

}

List<double> KoeffBigFunc(double a, double b, List<double> a1, List<double> b1)

{

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

return koeff;

}

public void Draw(Form t, List<double> varU, List<double> varFunc)

{

ZedGraphControl zed = new ZedGraphControl();

GraphPane pane = zed.GraphPane;

PointPairList list = new PointPairList();

pane.CurveList.Clear();

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

{

list.Add(varU[i], varFunc[i]);

}

LineItem myCurve = pane.AddCurve("Центр", list, Color.Blue, SymbolType.Circle);

pane.XAxis.MajorGrid.IsVisible = true;

pane.XAxis.MajorGrid.DashOn = 10;

pane.XAxis.MajorGrid.DashOff = 5;

pane.YAxis.MajorGrid.IsVisible = true;

pane.YAxis.MajorGrid.DashOn = 10;

pane.YAxis.MajorGrid.DashOff = 5;

pane.YAxis.MinorGrid.IsVisible = true;

pane.YAxis.MinorGrid.DashOn = 1;

pane.YAxis.MinorGrid.DashOff = 2;

pane.XAxis.MinorGrid.IsVisible = true;

pane.XAxis.MinorGrid.DashOn = 1;

pane.XAxis.MinorGrid.DashOff = 2;

pane.XAxis.Title.Text = "Ось u";

pane.YAxis.Title.Text = "Ось f(u)";

pane.Title.Text = "Зависимость производства центра от выделяемых ресурсов";

zed.AxisChange();

zed.Invalidate();

zed.Size = new Size(500, 500);

t.Controls.Add(zed);

}

public double Maximus(double start, double end)

{

double a = 0;

double b = 0;

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

List<double> tmp1 = new List<double>();

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

{

bool p = true;

for (int j = 0; j < Convert.ToInt32(end); j++)

{

if (paneles[i].varU.Contains(j))

{

if (j == paneles[i].varU[paneles[i].varU.IndexOf(j)])

{

a = paneles[i].funcKoeff[paneles[i].varU.IndexOf(j)][0];

try

{

b = paneles[i].funcKoeff[paneles[i].varU.IndexOf(j)][1];

}

catch (ArgumentOutOfRangeException)

{

b = 0;

p = false;

}

}

}

if (p)

{

jt.Add(j);

tmp1.Add(FunctionRes(a, b, j));

}

else

{

jt.Add(j);

tmp1.Add(a);

}

}

if (p)

{

jt.Add(Convert.ToInt32(end));

tmp1.Add(FunctionRes(a, b, Convert.ToInt32(end)));

}

else

{

jt.Add(Convert.ToInt32(end));

tmp1.Add(a);

}

}

double max = tmp1[0] + tmp1[tmp1.Count - 1];

for (int i = 0; i < tmp1.Count / 2; i++)

{

if (jt[i] + jt[tmp1.Count - 1 - i] == end)

max = Math.Max(max, tmp1[i] + tmp1[tmp1.Count - 1 - i]);

}

return max;

}

public double FunctionRes(double a, double b, double x)

{

return a \* x + b;

}

public List<List<double>> Cross(List<int> first, List<int> second)

{

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

double x1 = -((-first[2] \* second[1] + second[2] \* first[1]) / (first[0] \* second[1] - second[0] \* first[1]));

double x2 = -((-first[0] \* second[2] + second[0] \* first[2]) / (first[0] \* second[1] - second[0] \* first[1]));

List<double> crossx1x2 = new List<double>();

crossx1x2.Add(x1);

crossx1x2.Add(x2);

point.Add(crossx1x2);

List<double> crossx0;

if (first[0] != 0)

{

crossx0 = new List<double>();

x1 = first[2] / first[0];

x2 = 0;

crossx0.Add(x1);

crossx0.Add(x2);

point.Add(crossx0);

}

if (first[1] != 0)

{

crossx0 = new List<double>();

x1 = 0;

x2 = first[2] / first[1];

crossx0.Add(x1);

crossx0.Add(x2);

point.Add(crossx0);

}

if (second[0] != 0)

{

crossx0 = new List<double>();

x1 = second[2] / second[0];

x2 = 0;

crossx0.Add(x1);

crossx0.Add(x2);

point.Add(crossx0);

}

if (second[1] != 0)

{

crossx0 = new List<double>();

x1 = 0;

x2 = second[2] / second[1];

crossx0.Add(x1);

crossx0.Add(x2);

point.Add(crossx0);

}

return point;

}

public void LoadFile()

{

OpenFileDialog fdlg = new OpenFileDialog();

fdlg.Title = "Open file";

fdlg.InitialDirectory = @"C:";

fdlg.Filter = "All files (\*.\*)|\*.\*|Text files (\*.txt)|\*.txt";

fdlg.FilterIndex = 2;

fdlg.RestoreDirectory = true;

if (fdlg.ShowDialog() == DialogResult.OK)

{

boxdownload.Text = fdlg.FileName;

}

}

}

public class Panel1 : Panel

{

public TableLayoutPanel panel;

public List<System.Windows.Forms.Label> varLabel;

//public List<List<TextBox>> varTextbox;

public TextBox[,] varTextbox;

public List<TextBox> limits;

public System.Windows.Forms.Label name;

public List<ListBox> signs;

public static TextBox limitOfU;

public Panel biggerPanel;

public List<double> varU;// количество выделяемого ресурса подсистеме

public List<double> varFunc;// значения соотвествующие varU

public List<List<double>> koeffic;// точки, которые могут быть оптимальными

public List<List<double>> funcKoeff;// коэффициенты a и b в функциях посистемы

public TextBox getlimit()

{

return limitOfU;

}

public Panel1()

{

panel = new TableLayoutPanel();

biggerPanel = new Panel();

name = new System.Windows.Forms.Label();

name.Text = "Подсистема";

biggerPanel.Controls.Add(panel);

biggerPanel.CreateControl();

panel.RowCount = 2;

panel.ColumnCount = 2;

panel.Controls.Add(name);

panel.SetColumnSpan(name, 2);

biggerPanel.BorderStyle = BorderStyle.Fixed3D;

}

public Panel1(TabPage form)

{

panel = new TableLayoutPanel();

biggerPanel = new Panel();

name = new System.Windows.Forms.Label();

name.Text = "Подсистема";

biggerPanel.Controls.Add(panel);

biggerPanel.CreateControl();

panel.RowCount = 2;

panel.ColumnCount = 2;

panel.Controls.Add(name);

panel.SetColumnSpan(name, 2);

biggerPanel.BorderStyle = BorderStyle.Fixed3D;

form.Controls.Add(biggerPanel);

}

public Panel1(TabPage tabpage, int var, int equations, int number)

{

Size d = new Size(35, 25);

ListBox rest;

List<TextBox> uravnenie;

varTextbox = new TextBox[equations + 2, var + 1];

limits = new List<TextBox>();

signs = new List<ListBox>();

TextBox tmp;

panel = new TableLayoutPanel();

biggerPanel = new Panel();

biggerPanel.Size = new Size(50 \* (var \* 2 + 2), 40 \* (2 + equations));

panel.RowCount = 2 + equations;

panel.ColumnCount = var \* 2 + 2;

panel.Size = new Size(50 \* (var \* 2 + 2), 40 \* (2 + equations));

panel.CellBorderStyle = TableLayoutPanelCellBorderStyle.Inset;

System.Windows.Forms.Label z;

for (int i = 0; i < equations + 2; i++)

{

uravnenie = new List<TextBox>();

varLabel = new List<System.Windows.Forms.Label>();

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

{

tmp = new TextBox();

tmp.Size = d;

tmp.Text = "1";

uravnenie.Add(tmp);

panel.Controls.Add(uravnenie[j]);

varTextbox[i, j] = uravnenie[j];

z = new System.Windows.Forms.Label();

if (j != (var - 1)) { z.Text = "x" + number + (j + 1) + " " + "+" + " "; }

else { z.Text = "x" + number + (j + 1) + " "; }

z.Size = d;

varLabel.Add(z);

panel.Controls.Add(varLabel[j]);

}

rest = new ListBox();

rest.Items.Add("<=");

//rest.Items.Add(">=");

//rest.Items.Add("=");

rest.Size = d;

signs.Add(rest);

panel.Controls.Add(signs[i]);

tmp = new TextBox();

tmp.Size = d;

tmp.Text = "1";

limits.Add(tmp);

varTextbox[i, var] = limits[i];

System.Windows.Forms.Label u = new System.Windows.Forms.Label();

u.Size = d;

u.Text = "u" + number;

System.Windows.Forms.Label f = new System.Windows.Forms.Label();

f.Size = d;

f.Text = "f(u" + number + ") -> max";

if (i == 1)

{

panel.Controls.Add(u);

}

else if (i == 0)

{

panel.Controls.Add(f);

}

else

{

panel.Controls.Add(limits[i]);

}

}

biggerPanel.Controls.Add(panel);

tabpage.Controls.Add(biggerPanel);

}

public Panel1(TabPage tabpage, int equations)

{

Size d = new Size(45, 25);

ListBox rest;

panel = new TableLayoutPanel();

biggerPanel = new Panel();

biggerPanel.Size = new Size(50 \* (equations \* 2 + 2), 30 \* (2 + equations));

panel.RowCount = 2;

panel.ColumnCount = equations + 2;

panel.Size = new Size(50 \* (equations \* 2 + 2), 30 \* (2 + equations));

panel.CellBorderStyle = TableLayoutPanelCellBorderStyle.Single;

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

{

System.Windows.Forms.Label tmp = new System.Windows.Forms.Label();

tmp.Size = d;

if (i != (equations - 1))

tmp.Text = "f" + (i + 1) + "(u" + (i + 1) + ")" + "+" + " ";

else

tmp.Text = "f" + (i + 1) + "(u" + (i + 1) + ")" + " ";

panel.Controls.Add(tmp);

}

System.Windows.Forms.Label tmp1 = new System.Windows.Forms.Label();

tmp1.Text = " -> max";

panel.SetColumnSpan(tmp1, 2);

panel.Controls.Add(tmp1);

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

{

System.Windows.Forms.Label tmp = new System.Windows.Forms.Label();

tmp.Size = d;

if (i != (equations - 1))

tmp.Text = "u" + (i + 1) + "+" + " ";

else

tmp.Text = "u" + (i + 1) + " ";

panel.Controls.Add(tmp);

}

rest = new ListBox();

rest.Items.Add("<=");

//rest.Items.Add(">=");

//rest.Items.Add("=");

rest.Size = d;

panel.Controls.Add(rest);

limitOfU = new TextBox();

limitOfU.Size = d;

limitOfU.Text = "1";

panel.Controls.Add(limitOfU);

biggerPanel.Controls.Add(panel);

tabpage.Controls.Add(biggerPanel);

}

public Panel big()

{

return biggerPanel;

}

public void Koeff(int equations, int var)

{

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

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

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

for (int i = 0; i < var + 1; i++)

{

first.Add(Convert.ToInt32(varTextbox[equations, i].Text));

second.Add(Convert.ToInt32(varTextbox[equations + 1, i].Text));

third.Add(Convert.ToInt32(varTextbox[equations - 1, i].Text));

}

third[2] =Convert.ToInt32( limitOfU.Text);

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

double x1 = ((-(int)first[2] \* (int)second[1] + (int)second[2] \* (int)first[1]) / ((int)first[0] \* (int)second[1] - (int)second[0] \* (int)first[1])) \* (-1);

double x2 = ((-(int)first[0] \* (int)second[2] + (int)second[0] \* (int)first[2]) / ((int)first[0] \* (int)second[1] - (int)second[0] \* (int)first[1])) \* (-1);

List<double> crossx1x2 = new List<double>();

crossx1x2.Add(x1);

crossx1x2.Add(x2);

point.Add(crossx1x2);

List<double> crossx0;

if (first[0] != 0)

{

crossx0 = new List<double>();

x1 = first[2] / first[0];

x2 = 0;

crossx0.Add(x1);

crossx0.Add(x2);

point.Add(crossx0);

}

if (first[1] != 0)

{

crossx0 = new List<double>();

x1 = 0;

x2 = first[2] / first[1];

crossx0.Add(x1);

crossx0.Add(x2);

point.Add(crossx0);

}

if (second[0] != 0)

{

crossx0 = new List<double>();

x1 = second[2] / second[0];

x2 = 0;

crossx0.Add(x1);

crossx0.Add(x2);

point.Add(crossx0);

}

if (second[1] != 0)

{

crossx0 = new List<double>();

x1 = 0;

x2 = second[2] / second[1];

crossx0.Add(x1);

crossx0.Add(x2);

point.Add(crossx0);

}

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

{

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

{

if (i != j)

{

if (point[i][0] <= point[j][0] && point[i][1] <= point[j][1] && (point[j][0] == 0 || point[j][1] == 0))

{

point.Remove(point[j]);

if (i != 0)

i--;

}

//else if (point[i][0] <= point[j][0] && point[i][1] <= point[j][1] && (point[i][0] != 0 && point[i][1] != 0))

//{

// point.Remove(point[j]);

// if (i != 0)

// i--;

//}

}

}

}

koeffic = point;

}

public void Function()

{

Znachenie();

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

List<double> funcview;

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

{

funcview = new List<double>();

double z;

z = (varFunc[i + 1] - varFunc[i]) / (varU[i + 1] - varU[i]);

funcview.Add(z);

z = varFunc[i] - varU[i] \* z;

funcview.Add(z);

funcKoeff.Add(funcview);

}

funcview = new List<double>();

funcview.Add(varFunc[varFunc.Count - 1]);

funcKoeff.Add(funcview);

}

public void Znachenie()

{

varU = new List<double>();

varFunc = new List<double>();

varU.Add(0.0);

varFunc.Add(0.0);

double summ = 0.0;

double summfunc = 0.0;

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

{

summ = 0.0;

summfunc = 0.0;

for (int j = 0; j < koeffic[i].Count; j++)

{

summ += koeffic[i][j] \* Convert.ToDouble(varTextbox[1, j].Text);

summfunc += koeffic[i][j] \* Convert.ToDouble(varTextbox[0, j].Text);

}

varU.Add(summ);

varFunc.Add(summfunc);

}

double max = varU[0];

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

{

int min = i;

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

{

if (varU[j] < varU[min])

{

min = j;

}

}

double tmp;

tmp = varU[i];

varU[i] = varU[min];

varU[min] = tmp;

tmp = varFunc[i];

varFunc[i] = varFunc[min];

varFunc[min] = tmp;

}

//varU.Sort();

//varFunc.Sort();

}

public void DrawGraph(TabPage t, int l, double textl)

{

ZedGraphControl zed = new ZedGraphControl();

GraphPane pane = zed.GraphPane;

PointPairList list = new PointPairList();

pane.CurveList.Clear();

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

{

list.Add(varU[i], varFunc[i]);

if (i > 0)

{

TextObj text = new TextObj(Convert.ToString(funcKoeff[i - 1][0]), (varU[i] + varU[i - 1]) / 2, (varFunc[i] + varFunc[i - 1]) / 2);

text.FontSpec.Border.IsVisible = false;

pane.GraphObjList.Add(text);

}

}

list.Add(textl, varFunc[varU.Count - 1]);

LineItem myCurve = pane.AddCurve("Подсистема№" + l, list, Color.Blue, SymbolType.Circle);

pane.XAxis.MajorGrid.IsVisible = true;

pane.XAxis.MajorGrid.DashOn = 10;

pane.XAxis.MajorGrid.DashOff = 5;

pane.YAxis.MajorGrid.IsVisible = true;

pane.YAxis.MajorGrid.DashOn = 10;

pane.YAxis.MajorGrid.DashOff = 5;

pane.YAxis.MinorGrid.IsVisible = true;

pane.YAxis.MinorGrid.DashOn = 1;

pane.YAxis.MinorGrid.DashOff = 2;

pane.XAxis.MinorGrid.IsVisible = true;

pane.XAxis.MinorGrid.DashOn = 1;

pane.XAxis.MinorGrid.DashOff = 2;

pane.XAxis.Title.Text = "Ось u" + l;

pane.YAxis.Title.Text = "Ось f" + l + "(u" + l + ")";

pane.Title.Text = "Зависимость производства подсистемы от выделяемых ресурсов";

zed.AxisChange();

zed.Invalidate();

zed.Size = t.Size;

t.Controls.Add(zed);

}

public List<double> SolvesByIndex(double a1, double a2)

{

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

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

{

if (varU[i] <= a1 && varU[i] > a2)

{

lst.Add(varFunc[i]);

}

}

return lst;

}

public List<List<double>> NKoeff()

{

List<List<double>> a1 = new List<List<double>>();

return a1;

}

public List<List<double>> reduce(int variables)

{

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

for (int i = 2; i < limits.Count; i++)

{

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

for (int j = 0; j < variables + 1; j++)

{

a.Add(Convert.ToDouble(varTextbox[i, j].Text));

}

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

{

if (i == j+2)

a.Add(1);

else

a.Add(0);

}

red.Add(a);

}

return red;

}

}

}