SECTION 2. Applied mathematics. Mathematical modeling.

Shevtsov Alexandr Nikolayevich
candidate of technical Sciences,
President, Theoretical & Applied Science, LLP,
associate Professor of the Department «Mathematics»
Taraz State University named after M.Kh. Dulati, Kazakhstan

Tangirbergenova Assem Kuanishbaevna
Teacher of mathematics, Master of 1 course of specialty "Mathematics"
Gymnasium №45 named after B. Momyshuly, Taraz, Kazakhstan

OF THE BASIC DECISIONS OF THE METHOD OF SUCCESSIVE IMPROVEMENTS.

The article considers some aspects of implementation of computer method of basis solutions.

Key words: the simplex method basis, Delphi.

О БАЗИСНЫХ РЕШЕНИЯХ МЕТОДА ПОСЛЕДОВАТЕЛЬНЫХ УЛУЧШЕНИЙ.

В статье рассматриваются некоторые аспекты компьютерной реализации метода базисных решений.

Ключевые слова: симплекс метод, базис, Дельфи.

Для изучения выпуклых множеств S, рассмотривают - «крайные точки» [1].

Условие существования крайней точки.

Если из того, что

$$\mathrm{x}[N] = \frac{1}{2} \cdot x'[N] + \frac{1}{2} \cdot x''[N], \tag{1}$$

и

$$x'[N], \quad x''[N] \in S, \tag{2}$$

следует, что

$$x[N] = x'[N] = x''[N]. \tag{3}$$

Геометрическая интерпретация.

Не существует лежаещего в S отрезка положительный длины, серединой которого являлась бы крайняя точка x[N].

Рассмотрим карйние точки множества допустимых решений с стандартной задачи линейного программирования.

Найти x[N], удетлотворяющий условиям

$$x[N] \geq 0[N], \tag{4}$$
$$a[M,N] \cdot x[N] = b[N], \tag{5}$$

и минимизирующий

$$\mathrm{c}[N] \cdot x[N]. \tag{6}$$

Пусть $r = rank(a[M,N])$. Будем считать (это необходимо для разрешимости условий (5)), что добавление к матрице $a[M,N]$ вектора $b[N]$ в качестве еще одного столбца не увеличивает ранга матрицы.

Пусть множество $N' \subset N$ такого, что $rank(a[M,N]) = r$ и $|N'| = r$. Вектор $x[N]$, удовлетворяющий равенствам

$$a[M,N] \cdot x[N'] = b[M], \tag{7}$$
$$x[N \backslash N'] = 0[N \backslash N'], \tag{8}$$

называется базисным решением. Множество N' называется *базисом*. Напомним, что $x[N]$ находится из (7), (8) однозначно. Действительно, достаточно выбрать $M' \subset M$ ткаим образом, чтобы

$$|M'| = r,\ rank\ (a[M',N']) = r, \tag{9}$$

и положить

$$x[N'] = a^{-}[N',M'] \cdot b[M'], \tag{10}$$

где $a^{-}[N',M']$ - матрица обратная к $a[N',M']$.

Одно и то же базисное решение может соответствовать различным базисам.

Пример :

Рассмотрим систему для задачи линейного программиривания.

$$\begin{cases} 3x_1 + 4x_2 + x_3 + 6x_4 = 6 \\ x_1 - x_2 + 2x_3 + 2x_4 = 2 \end{cases} \tag{11}$$

базисное решение $x_1 = 2,\ x_2 = x_3 = x_4 = 0$ соответствует любому из базисов$\{1, 2\}$ и $\{1, 3\}$.

Основные теоремы и свойства.

Теорема 1. Для того чтобы вектор $x[N]$ был крайней точкой множество допустимых решений, необходимо и достаточно, чотбы он был допустимым базисным решением.

Теорема 2. Если множество допустимых решений задачи линейного программирования ограничено, оно является выпуклой оболочкой допустимых базисных решений.

Теорема 3. Любое допустимое решение задачи линейного программирования представимо в виде суммы

$$x[N] = x_0[N] + y[N], \tag{12}$$

где $y[N]-$ элемент конуса неотрицательных решений однородной системы линейных ограничений, а $x_0[N]-$ выпуклая комбинация базисных решений.

Теорема 4. Если в задаче линейного программирования имеется оптимальное решение, то в ней имеется и оптимальное базисное решение.

Свойство 1. Вектор $x[N]$ называется *выпуклой комбинацией* векторов $y_i[M]$, $i \in M$, если найдется такой с - вероятностный вектор $\lambda[M]$, что

$$x[N] = \lambda[M] \cdot y_M[N] = \sum_{i \epsilon M} \lambda[i] \cdot y_i[N]. \tag{13}$$

Множество всех выпуклых комбинаций векторов $y_i[N]$ называется выпуклой оболочкой этих векторов.

Свойство 2. Если $x[N'] \geq 0[N']$, базисное решение называется *допустимым базисным решением.*

Методы решения.

Для решения задач линейного программирования отыскиваются оптимальные базисные решения. Наибольшее распространение получил *метод последовательных улучшений.* В основе этого метода лежит способ, по которому, имея некоторое допустимое базисное решение, можно проверить его оптимальность и в случае его неоптимальности построить новое допустимое базисное решение с меньшим значением целевой функции. К рассмотрению этого метода мы сейчас и перейдем.

Рассмотрим один шаг улучшений, который может повторяться многократно. Однако должно быть какое-то допустимое базисное решение, с которого можно было бы начать – начальное допустимое базисное решение. Проще всего такое решение можно получить с помощью так называемого *метода искусственных переменных.*

Метод искусственного базиса.

Для многих задач линейного программирования, записанных в форме основной задачи и имеющих опорные планы, среди векторов P_j не всегда есть m единичных.

Рассмотрим следующую задачу:

Пусть требуется найти максимум функции

$$F = c_1x_1 + c_2x_2 + \ldots\ldots + c_nx_n \tag{14}$$

при условиях

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots\ldots\ldots + a_{1n}x_n = b_1 , \\ a_{21}x_1 + a_{22}x_2 + \ldots\ldots\ldots + a_{2n}x_n = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{2m}x_1 + a_{m2}x_2 + \ldots\ldots\ldots + a_{mn}x_n = b_m \\ x_j \geq 0, \quad j = \overline{1, n} \end{cases} \tag{15}$$

где $b_i \geq 0$ (i=l, m), $m<n$ и среди векторов P_1, P_2, …, P_n нет m единичных.

<u>Определение</u>. Задача, состоящая в определении максимального значения функции

$$F = c_1x_1 + c_2x_2 + \ldots\ldots + c_nx_n - Mx_{n+1} - \ldots - Mx_{n+m} \tag{16}$$

при условиях

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots\dots\dots + a_{1n}x_n + x_{n+1} = b_1 \ , \\ a_{21}x_1 + a_{22}x_2 + \dots\dots\dots + a_{2n}x_n + x_{n+2} = b_2 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots\dots\dots + a_{mn}x_n + x_{n+m} = b_m \\ \boldsymbol{x_j \geq 0, \quad j = \overline{1, n+m}} \end{cases} \tag{17}$$

где M — некоторое достаточно большое положительное число, конкретное значение которого обычно не задается, называется расширенной задачей по отношению к задаче (14) — (15).

Расширенная задача имеет опорный план

$$X=(0;\ 0;\ ...;\ 0;\ b_1;\ b_2;\ ...;b_m).$$

определяемых системой единичных векторов P_{n+1}; P_{n+2}, ... P_{n+m}, образующих базис m-го векторного пространства, который называется *искусственным.* Сами векторы, так же как и переменные x_{n+i} (i=l, m), называются *искусственными.* Так как расширенная задача имеет опорный план, то ее решение может быть найдено симплексным методом.

Теорема. *Если в оптимальном плане* $X^*=(x^*_1,\ x^*_2,\ ...;\ x^*_n,\ x^*_{n+1};\ ...;\ x^*_{n+m})$ *расширенной задачи* (16) — (17) *значения искусственных переменных* $x^*_{n+i}=0$ (i=1, m), *то* $X^*=(x^*_1,\ x^*_2,\ ...;\ x^*_n)$ *является оптимальным планом задачи* (14) — (15).

Таким образом, если в найденном оптимальном плане расширенной задачи, значения искусственных переменных равны нулю, то тем самым получен оптимальный план исходной задачи.

Значения индексной строки Δ_0, Δ_1, …, Δ_n состоят из двух частей, одна из которых зависит от *M*, а другая — нет. Заполняют симплекс - таблицу, которая содержит на одну строку больше, чем обычная симплексная таблица. При этом в (m+2)-ю строку помещают коэффициенты при *M*, а в (m+1)-ю – слагаемые, не содержащие *M*. При переходе от одного опорного плана к другому в базис вводят вектор, соответствующий наибольшему по абсолютной величине отрицательному числу (m+2)-й строки. Искусственный вектор, исключенный из базиса, в следующую симплекс-таблицу не записывают. Пересчет симплекс-таблиц при переходе от одного опорного плана к другому производят по общим правилам симплексного метода.

Итерационный процесс по (m+2) -и строке ведут до тех пор, пока:
либо все искусственные векторы не будут исключены из базиса;
либо не все искусственные векторы исключены, но (m+2)-я строка не содержит больше отрицательных элементов в индексах Δ_1, …, Δ_n.

В первом случае базис отвечает некоторому опорному плану исходной задачи и определение ее оптимального плана продолжают по (m+1)-й строке.

Во втором случае, если значение Δ_0 отрицательное, исходная задача не имеет решения; если же $\Delta_0=0$, то найденный опорный план исходной задачи является вырожденным и базис содержит по крайней мере один из векторов искусственного базиса.

Этапы нахождения решения задачи (14) — (15) методом искусственного базиса:

Составляют расширенную задачу (16) — (17).

Находят опорный план расширенной задачи.

С помощью обычных вычислений симплекс-метода исключают искусственные переменные из базиса. В результате либо находят опорный план исходной задачи (14) — (15), либо устанавливают ее неразрешимость.

Используя найденный опорный план задачи (14) — (15), либо находят симплекс-методом оптимальный план исходной задачи, либо устанавливают ее неразрешимость.

Рассмотрим пример.

Найти минимум функции $F= - 2x_1 + 3x_2 - 6x_3 - x_4$

при ограничениях:

$$\begin{cases} 2x_1+x_2-2x_3+x_4=24 \\ x_1+2x_2+4x_3\le 22 \\ x_1-x_2+2x_3\ge 10 \\ x_i\ge 0,\ i=1,4 \end{cases}$$

Решение.

Запишем данную задачу в форме основной задачи: найти максимум функции $F= 2x_1 - 3x_2 + 6x_3 + x_4$

при ограничениях:

$$\begin{cases} 2x_1+x_2-2x_3+x_4=24 \\ x_1+2x_2+4x_3+x_5=22 \\ x_1-x_2+2x_3- x_6=10 \\ x_i\ge 0,\ i=1, 6 \end{cases}$$

В системе уравнений последней задачи рассмотрим расширенные векторы из коэффициентов при неизвестных

$$P_1=\begin{pmatrix}-2\\2\\1\\1\end{pmatrix};\ P_2=\begin{pmatrix}3\\1\\2\\-1\end{pmatrix};\ P_3=\begin{pmatrix}-6\\-2\\4\\2\end{pmatrix};\ P_4=\begin{pmatrix}-1\\1\\0\\0\end{pmatrix};$$

$$P_5 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}; \quad P_6 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \end{pmatrix}; \qquad P_7 = \begin{pmatrix} M \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

Расссмотрим расширенную задачу, состоящую в максимизации функции

$$F = 2x_1 - 3x_2 + 6x_3 + x_4 - Mx_7$$

При ограничениях:

$$\begin{cases} 2x_1 + x_2 - 2x_3 + x_4 = 24 \\ x_1 + 2x_2 + 4x_3 + x_5 = 22 \\ x_1 - x_2 + 2x_3 - x_6 + x_7 = 10 \end{cases}$$

Расширенная задача имеет опорный план $X = (0;\ 0;\ 0;\ 24;\ 22;\ 0;\ 10)$, определяемый системой трех единичных векторов: $P_4,\ P_5,\ P_7$.

$$F - 2x_1 + 3x_2 - 6x_3 - x_4 + Mx_7 = 0$$

Решим ее симплекс методом:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	Решение	Отношение
F	-2	3	-6	-1	0	0	M	0	
x_5	2	1	-2	1	0	0	0	24	24/-2=-12
x_6	1	2	4	0	1	0	0	22	22/4=5,5
x_7	1	-1	2	0	0	-1	1	10	10/2=5

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	Решение	Отношение
F	1	0	0	-1	0	-3	M+3	30	30/-3=-10
x_5	3	0	0	1	0	-1	1	34	34/-1=-34
x_6	-1	4	0	0	1	2	-2	2	2/2=1
x_7	1/2	-1/2	1	0	0	-1/2	1/2	5	10*(-2)=-20

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	Решение	Отношение
F	-1/2	6	0	-1	3/2	0	M	33	33/-1=-33
x_5	5/2	2	0	1	1/2	0	0	35	35/1=35
x_6	-1/2	2	0	0	1/2	1	-1	22	-
x_7	1/2	1/2	1	0	1/4	0	0	11/2	-

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	Решение

F	2	8	0	0	2	0	M	68
x_5	5/2	2	0	1	1/2	0	0	35
x_6	-1/2	2	0	0	1/2	1	-1	22
x_7	1/2	1/2	1	0	1/4	0	0	11/2

Рассмотрим алгоритм ввода данных в программу для данной задачи. Будем загружать систему с условиями и функцией *F* из текстового документа(рис.1).

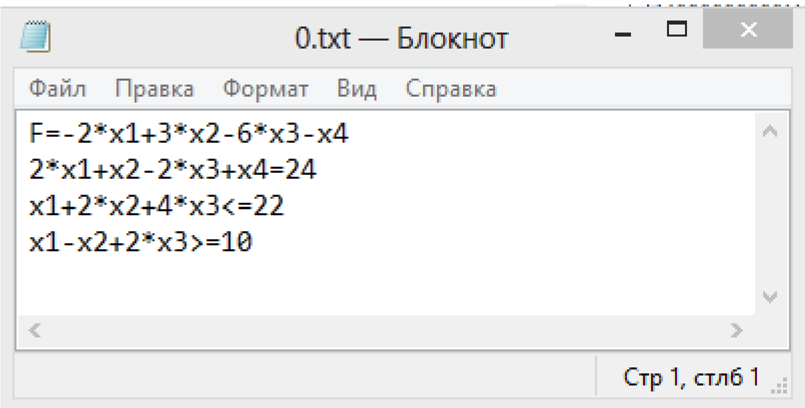

Рисунок 1 – Исходная система ограничений.

После загрузки данных в Memo, необходимо выделить матрицу коэффициентов системы условий и управляющую функцию, а также с учетом введенных условий задать искуственный базис.

```
type
xxx=array[1..9]of string;      aaa=array[1..9] of integer;
bb=array[0..9] of real;        ab=array[0..9,0..9] of real;
var
  Form1: TForm1;
   name1:string;
n,m,i,j:integer;
aa,xx,zz:xxx;
pn:aaa;
b:bb;       a:ab;
implementation

{$R *.dfm}

procedure TForm1.FileListBox1Click(Sender: TObject);
var i,j,p,pp:integer;
s,s0,f,bi,ix:string;
begin
```

```
name1:=FileListBox1.FileName;
if fileexists(name1) then
begin
memo1.Lines.LoadFromFile(name1);
// определим число уравнений
m:=memo1.Lines.Count-1;
// определим число переменных
n:=0;
for j := 1 to 9 do
begin
s0:=xx[j];
for I := 0 to m do
begin
  s:=memo1.Lines.Strings[i];
  if pos(s0,s)<>0 then  n:=j;
end;
end;
label1.Caption:='Кол-во переменных : '+inttostr(n);
//  F
s:=memo1.Lines.Strings[0];
f:=s;
delete(f,1,2);

for I := 1 to m  do
begin
s:=memo1.Lines.Strings[i];
for j := 1 to 5 do
    if pos(zz[j],s)<>0 then
    begin
    p:=pos(zz[j],s);
    pn[i]:=j;
    end;

aa[i]:=s;
delete(aa[i],p,length(s));
bi:=s;
if pn[i]>3 then pp:=2 else pp:=1;

delete(bi,1,p+pp-1);
b[i]:=strtofloat(bi);
end;
memo2.Clear;
memo3.Clear;
```

```
memo2.Lines.Add(f);
b[0]:=0;
memo3.Lines.Add(floattostr(b[0]));
memo4.Clear;
memo4.Lines.Add('');
s:=f;
for j:=1 To n do
begin
s0:=xx[j];
p:= pos(s0,s);
if p>0 then
if copy(s,p-1,1)<>'*' then insert('1*',s,p);
end;
while length(s)>0 do
begin
s0:=copy(s,1,pos('*',s)-1);
ix:=copy(s,pos('*',s)+2,1);
delete(s,1,pos('*',s)+2);
a[0,strtoint(ix)]:=strtofloat(s0);
end;
for I := 1 to m do
begin
memo2.Lines.Add(aa[i]);
memo3.Lines.Add(floattostr(b[i]));

s:=aa[i];
for j:=1 to n do
begin
s0:=xx[j];
p:= pos(s0,s);
if p>0 then
if copy(s,p-1,1)<>'*' then insert('1*',s,p);
end;
memo4.Lines.Add(s);

while length(s)>0 do
begin
s0:=copy(s,1,pos('*',s)-1);
ix:=copy(s,pos('*',s)+2,1);
delete(s,1,pos('*',s)+2);
a[i,strtoint(ix)]:=strtofloat(s0);
end;
end;
```

```
for j := 1 to n do     stringgrid1.Cells[j,0]:=xx[j];
for I := 0 to m do  for j := 1 to n do
stringgrid1.Cells[j,i+1]:=floattostr(a[i,j]);

stringgrid1.RowCount:=m+2;          stringgrid1.ColCount:=n+2;
end; end;

procedure TForm1.FormCreate(Sender: TObject);
begin
FileListBox1.ItemIndex:=0;
for I := 1 to 9 do      xx[i]:='x'+inttostr(i);
zz[1]:='=';
zz[2]:='<';
zz[3]:='>';
zz[4]:='<=';
zz[5]:='>=';
end;
```

Получаем следующую матрицу(рис.1). Теперь учитывая знаки неравенств умножаем на -1 и заполняем базис (рис.2).

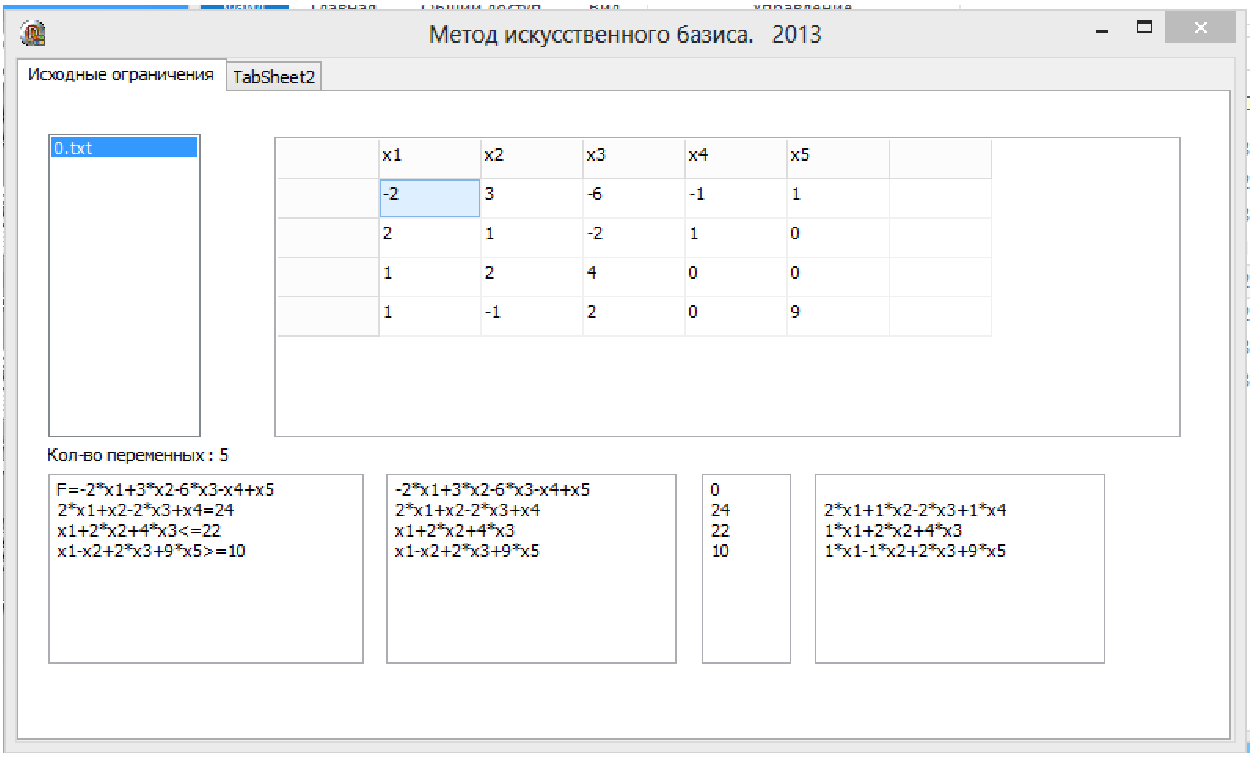

Рисунок 1 – Матрица ограничений.

	x1	x2	x3	x4	x5	x6	x7	x8	
	-2	3	-6	-1	0	0	0	0	0
x5	2	1	-2	1	1	0	0	0	24
x6	1	2	4	0	0	1	0	0	22
x8	1	-1	2	0	0	0	-1	1	10

Рисунок 2 – Искуственный базис.

Дополнительно к этому необходимо определить, что мы хотим найти: минимум или максимум функции F. В первом случае необходимо поменять знак у всех слагаемых первой строки матрицы(рис.3).

Проведем апробацию разработанных алгоритмов на примере другой системы с большим количеством переменных и ограничивающих условий (рис.4).

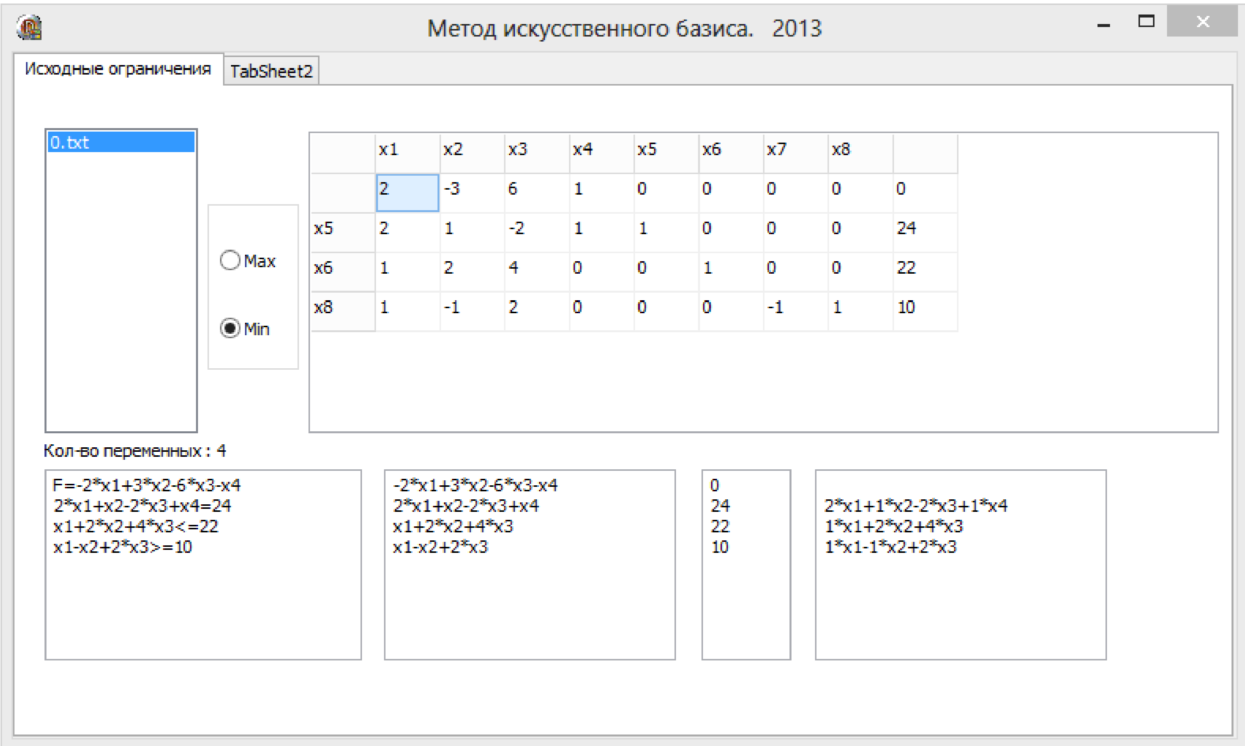

Рисунок 3 – Нахождение минимума функции.

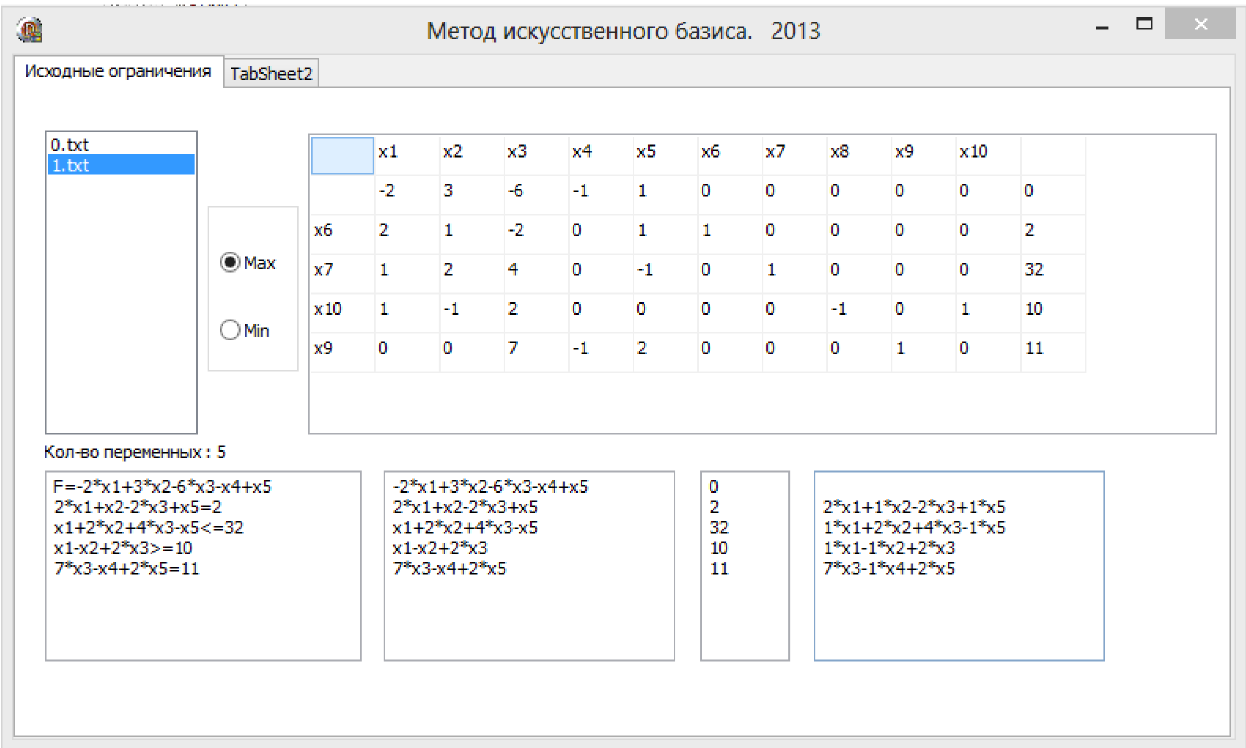

Рисунок 4 – Апробация алгоритмов.

```
// базис
j:=n+1;
k:= n+m; kk:=1;

for I := 1 to m do
for j := n+1 to n+3 do

if i=j-n then
if pn[i]=5 then  begin stringgrid1.Cells[j,i+1]:='-1';
                 stringgrid1.Cells[n+m+kk,i+1]:='1';
                 stringgrid1.Cells[0,i+1]:=xx[n+m+kk];
                 inc(kk);  end else
              begin stringgrid1.Cells[j,i+1]:='1';
                 stringgrid1.Cells[0,i+1]:=xx[j];
              end
        else
              stringgrid1.Cells[j,i+1]:='0';

kk:=n+m+kk-1;

for I := 0 to m do
for j := n+1 to kk do
begin
if stringgrid1.Cells[j,i+1]='' then stringgrid1.Cells[j,i+1]:='0';
```

```
a[i,j]:=strtofloat(stringgrid1.Cells[j,i+1]);
end;

  for j := 1 to kk do
stringgrid1.Cells[j,0]:=xx[j];

stringgrid1.ColCount:=kk+2;

for I := 1 to m+1 do
stringgrid1.Cells[kk+1,i]:=floattostr(b[i-1]);

end;
RadioGroup1.ItemIndex:=0;
```

Полученные алгоритмы могут быть использованны для решения экстремальных задач с применением метода искуственного базиса.

Литература

1. Романовский И.В. Алгоритмы решения экстремальных задач. Главная редакция физико-математической литературы из-ва «Наука», М., 1977.
2. Метод искусственного базиса (М-метод). [Электронный ресурс]. URL: http://old.tisbi.org/resource/lib/linprog/main3.htm (дата обращения: 25.10.2013).