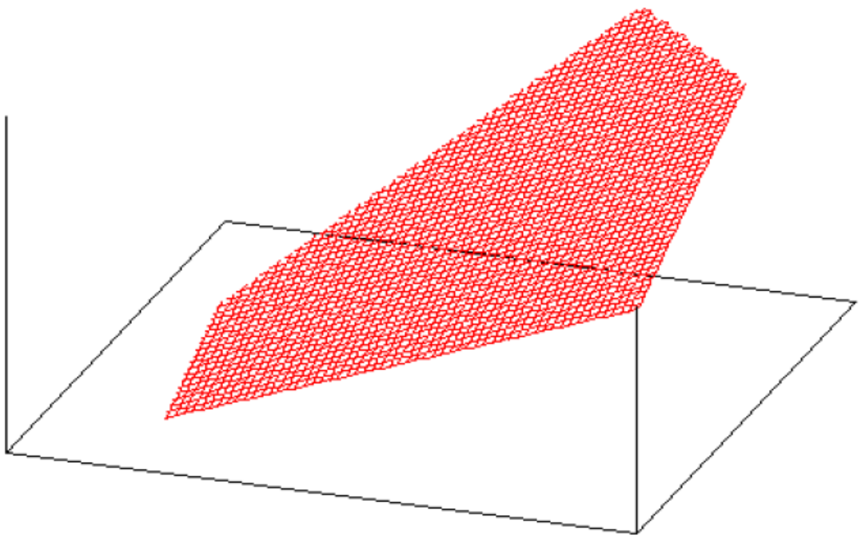
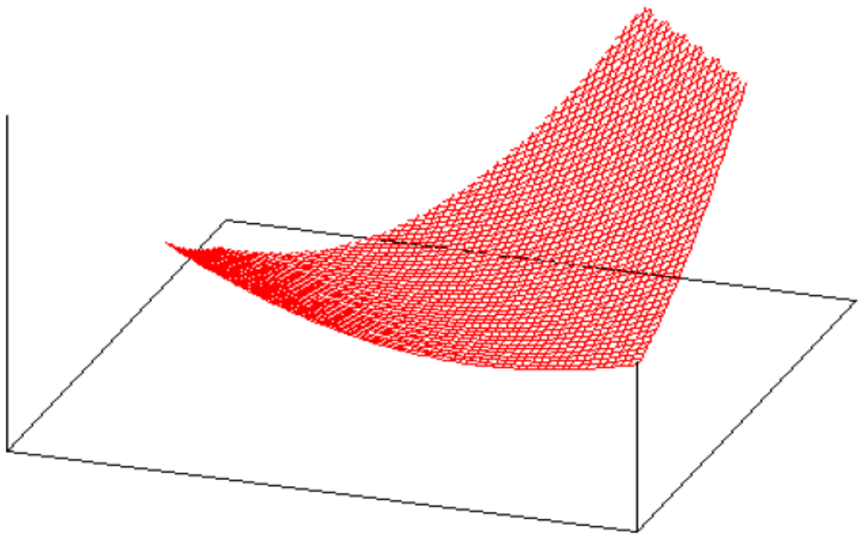
Chapter 9. Finding a minimum or maximum of a function under constraints for variables





9-1. Solving optimization problems under constraints using graphs.

Example. Find a maximum of the linear function

$$F(x,y) = x+3y$$

under constraints:

$$x - y \leq 1$$

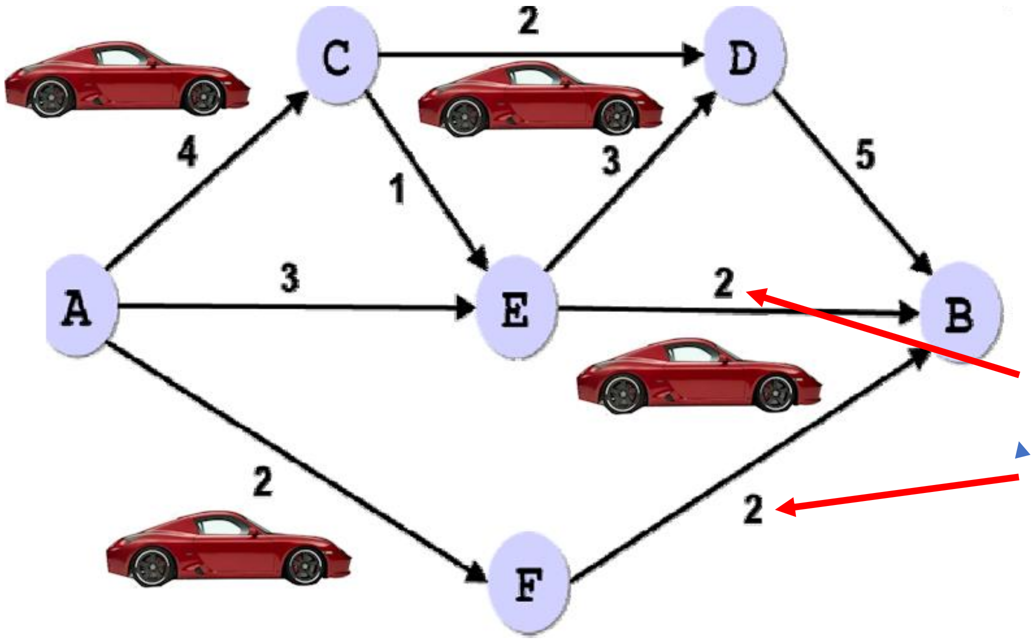
$$2x + y \leq 2$$

$$x - y \geq 0$$

$$x \geq 0$$

$$y \geq 0$$

9-2. Problem of Maximum Traffic



1) There are roads between cities A, B, C, D, E, F,

2) there are upper limits c_{ij} on the number of cars which can pass along each road per hour.

(the limits c_{ij} are given in thousands of cars).

Problem: find the maximum number of cars that can arrive in town B per hour.

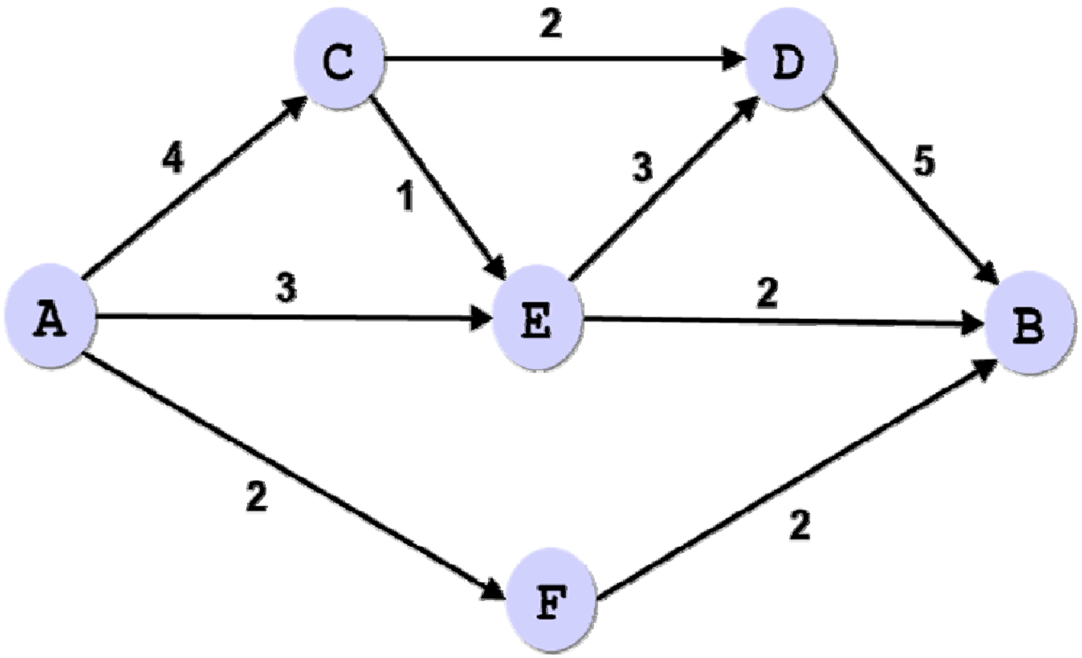
Unknown quantities are the optimum numbers of cars x_{ij} passing along each road per hour. **Here we have CONSTRAINTS c_{ij}.**

Conditions:

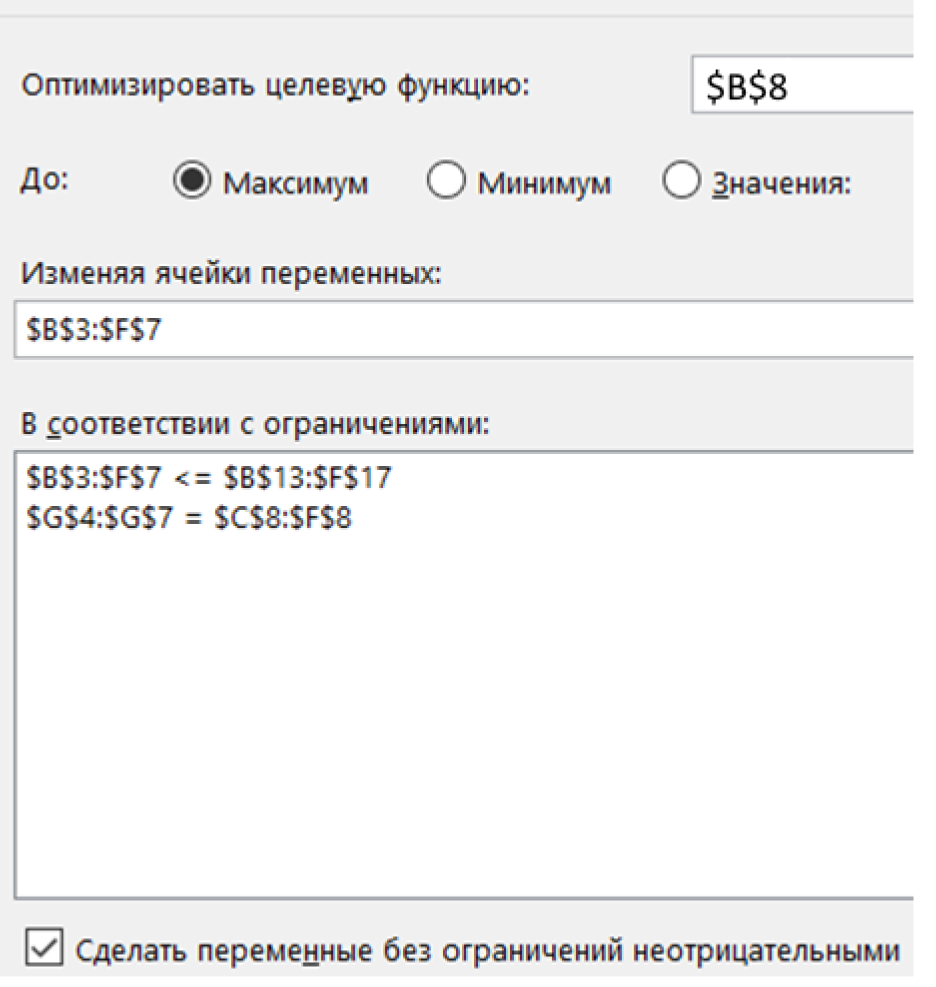
1) the number of cars that enter and depart from the city must be the same.

2) traffic limits: $0 \leq x_{ij} \leq c_{ij}$

3) target function: $\Sigma x_{iB} \rightarrow \max$



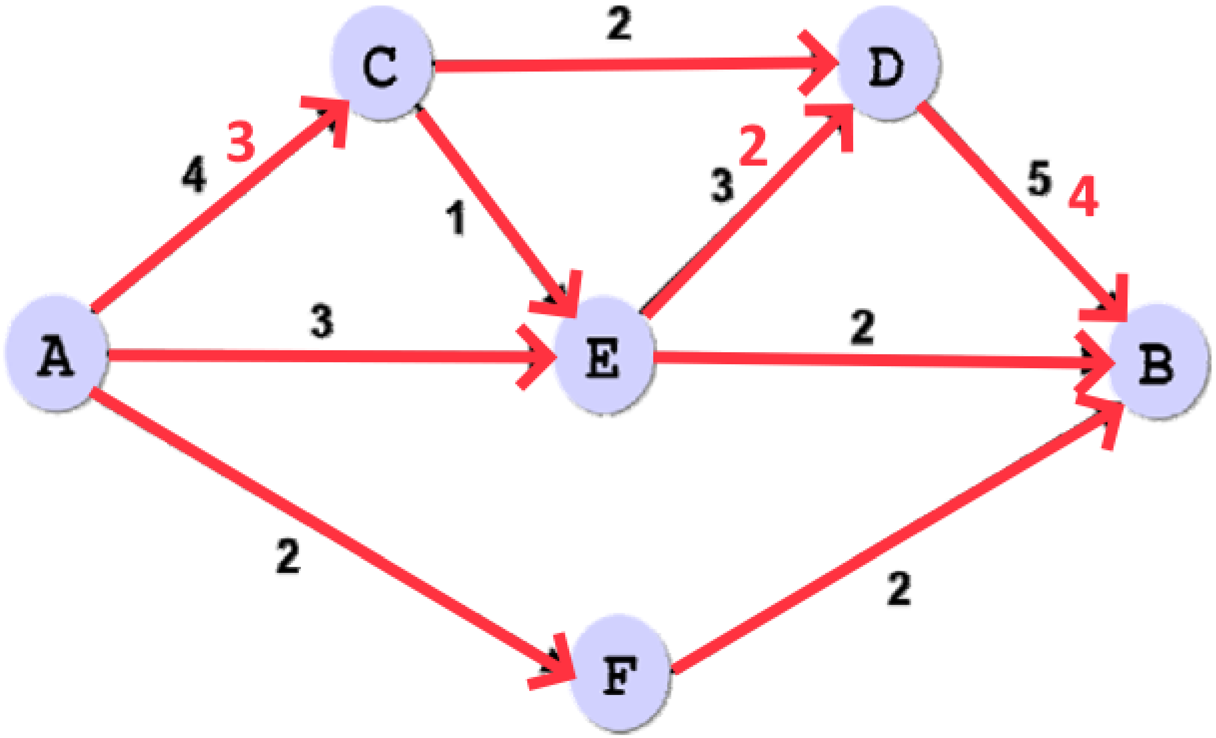
	A	B	C	D	E	F	G
1		unknown optimum traffic xij					
2		city B	city C	city D	city E	city F	Total from city
3	cityA	1	1	1	1	1	5
4	cityC	1	1	1	1	1	5
5	cityD	1	1	1	1	1	5
6	cityE	1	1	1	1	1	5
7	cityF	1	1	1	1	1	5
8	Total to city	5	5	5	5	5	
9							
10							
11		maximum traffic					
12		to B	to C	to D	to E	to F	
13	from A	0	4	0	3	2	
14	from C	0	0	2	1	0	
15	from D	5	0	0	0	0	
16	from E	2	0	3	0	0	
17	from F	2	0	0	0	0	
18							



In the option “Solve” we set the conditions 1) – 3):

Solution:

	A	B	C	D	E	F	G
1		unknown optimum traffic xij					
2		city B	city C	city D	city E	city F	Total from city
3	cityA	0	3	0	3	2	8
4	cityC	0	0	2	1	0	3
5	cityD	4	0	0	0	0	4
6	cityE	2	0	2	0	0	4
7	cityF	2	0	0	0	0	2
8	Total to city	8	3	4	4	2	
9							


C
D
A
E
B
F
2
4
3
1
3
2
5
4
3
2
2
2

9-3. Minimization of the delivery cost in case of a multi-variable function under constraints.

Example.

Shops / Available goods (e.g., bottles of juice)	Shop 1 needs b_1=110	Shop 2 needs b_2=350	Shop 3 needs b_3=140
Warehouse 1: a_1= 180	c_{11}	c_{12}	c_{13}
Warehouse 2: a_2= 300	c_{21}	c_{22}	c_{23}
Warehouse 3: a_3= 120	c_{31}	c_{32}	c_{33}

c_{ij} – delivery cost for 1 piece of goods from Warehouse i to Shop j

Shop

Warehouse/Storage

Shops / Available goods	Shop 1 needs b_1=110	Shop 2 needs b_2=350	Shop 3 needs b_3=140
Warehouse 1: a_1= 180	2 $	5 $	2 $
Warehouse 2: a_2= 300	7 $	7 $	13 $
Warehouse 3: a_3= 120	3 $	6 $	8 $

c_{ij} – **delivery cost for 1 piece of goods from Warehouse** i **to Shop** j

X_{ij} – **number of goods to be delivered from Warehouse** i **to Shop** j **- ?**

Suppose that total need $\Sigma_i a_i = \Sigma_j b_j$ total storage

Full cost:

$$F = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \to \min$$

Each shop's need must be satisfied

$$\begin{cases} \sum_{i=1}^{m} x_{ij} = b_j, \quad j = \overline{1,n} \\ \sum_{j=1}^{n} x_{ij} = a_i, \quad i = \overline{1,m} \\ x_{ij} \geq 0 \end{cases}$$

All available Warehouse's goods must be sent out

1st step of solving the problem:

Search of a first approximation to solution x_{ij} by

(*) Method of "North-West corner" , or

(*) Method of the least element.

2nd step: refinement of x_{ij} using a tool of Excel

(*) Method of "North-West corner"

	Shop 1 b_1=110	Shop 2 b_2=350	Shop 3 b_3=140
Warehouse 1: a1= 180	c_{11} x_{11}	c_{12} x_{12}	c_{13} x_{13}
Warehouse 2: a2= 300	c_{21} x_{21}	c_{22} x_{22}	c_{23} x_{23}
Warehouse 3: a3= 120	c_{31} x_{31}	c_{32} x_{32}	c_{33} x_{33}

	Shop 1 initial need b_1=110 current **0**	Shop 2 b_2=350	Shop 3 b_3=140
Warehouse 1: initial a_1= 180 current **70**	2 \$ 110	5 \$	2 \$
Warehouse 2: a_2= 300	7 \$	7 \$	13 \$
Warehouse 3: a_3= 120	3 \$	6 \$	8 \$

	Shop 1 initial need b_1=110 current 0	Shop 2 initial need b_2=350 current 280	Shop 3 b_3=140
Warehouse 1: initial a_1= 180 current 0	2 \$ 110	5 \$ 70	2 \$
Warehouse 2: a_2= 300	7 \$	7 \$	13 \$
Warehouse 3: a_3= 120	3 \$	6 \$	8 \$

	Shop 1 initial need b_1=110 current 0	Shop 2 initial need b_2=350 current 0	Shop 3 initial need b_3=140
Warehouse 1: initial a_1= 180 current 0	2 \$ 110	5 \$ 70	2 \$
Warehouse 2: initial a_2= 300 current 20	7 \$	7 \$ 280	13 \$
Warehouse 3: a_3= 120	3 \$	6 \$	8 \$

	Shop 1 initial need b_1=110 current 0	Shop 2 initial need b_2=350 current 0	Shop 3 initial need b_3=140 current 120
Warehouse 1: initial a_1= 180 current 0	2 \$ 110	5 \$ 70	2 \$
Warehouse 2: initial a_2= 300 current 0	7 \$	7 \$ 280	13 \$ 20
Warehouse 3: a_3= 120	3 \$	6 \$	8 \$

	Shop 1 initial need b_1=110 current 0	Shop 2 initial need b_2=350 current 0	Shop 3 initial need b_3=140 current 0
Warehouse 1: initial a_1= 180 current 0	2 \$ 110	5 \$ 70	2 \$
Warehouse 2: initial a_2= 300 current 0	7 \$	7 \$ 280	13 \$ 20
Warehouse 3: initial a_3= 120 current 0	3 \$	6 \$	8 \$ 120

Total cost x_{ij} **:** $F = 110\times2+70\times5+280\times7+20\times13+120\times8 = 3750$ \$

(*) Method of the least element:

We choose step-by-step the shop with minimum c_{32} and maximum b_j

	Shop 1 initial need b_1=110	Shop 2 b_2=350	Shop 3 b_3=140
Warehouse 1: a1= 180	2 \$	5 \$	2 \$
Warehouse 2: a2= 300	7 \$	7 \$	13 \$
Warehouse 3: a3= 120	3 \$	6 \$	8 \$

	Shop 1 initial need b_1=110	Shop 2 initial need b_2=350	Shop 3 initial need b_3=140 current 0
Warehouse 1: initial a_1= 180 current 40	2 $	5 $	2 $ 140
Warehouse 2: a_2= 300	7 $	7 $	13 $
Warehouse 3: a_3= 120	3 $	6 $	8 $

	Shop 1 initial need b_1=110 current 70	Shop 2 initial need b_2=350	Shop 3 initial need b_3=140 current 0
Warehouse 1: initial a_1= 180 current 0	2 \$ 40	5 \$	2 \$ 140
Warehouse 2: a_2= 300	7 \$	7 \$	13 \$
Warehouse 3: a_3= 120	3 \$	6 \$	8 \$

	Shop 1 initial need b_1=110 current 0	Shop 2 initial need b_2=350	Shop 3 initial need b_3=140 current 0
Warehouse 1: initial a_1= 180 current 0	2 $ 40	5 $	2 $ 140
Warehouse 2: a_2= 300	7 $	7 $	13 $
Warehouse 3: initial a_3= 120 current 50	3 $ 70	6 $	8 $

	Shop 1 initial need b_1=110 current 0	Shop 2 initial need b_2=350 current 300	Shop 3 initial need b_3=140 current 0
Warehouse 1: initial a_1= 180 current 0	2 \$ 40	5 \$	2 \$ 140
Warehouse 2: a_2= 300	7 \$	7 \$	13 \$
Warehouse 3: initial a_3= 120 current 0	3 \$ 70	6 \$ 50	8 \$

	Shop 1 initial need b_1=110 current 0	Shop 2 initial need b_2=350 current 0	Shop 3 initial need b_3=140 current 0
Warehouse 1: initial a_1= 180 current 0	2 \$ 40	5 \$	2 \$ 140
Warehouse 2: initial a_2= 300 current 0	7 \$	7 \$ 300	13 \$
Warehouse 3: initial a_3= 120 current 0	3 \$ 70	6 \$ 50	8 \$

Total cost x_{ij} : $F = 40\times2+70\times3+300\times7+50\times6+140\times2 = 2970$ \$

Solving the same problem with EXCEL

First, we insert given c_{ij} a_i b_j

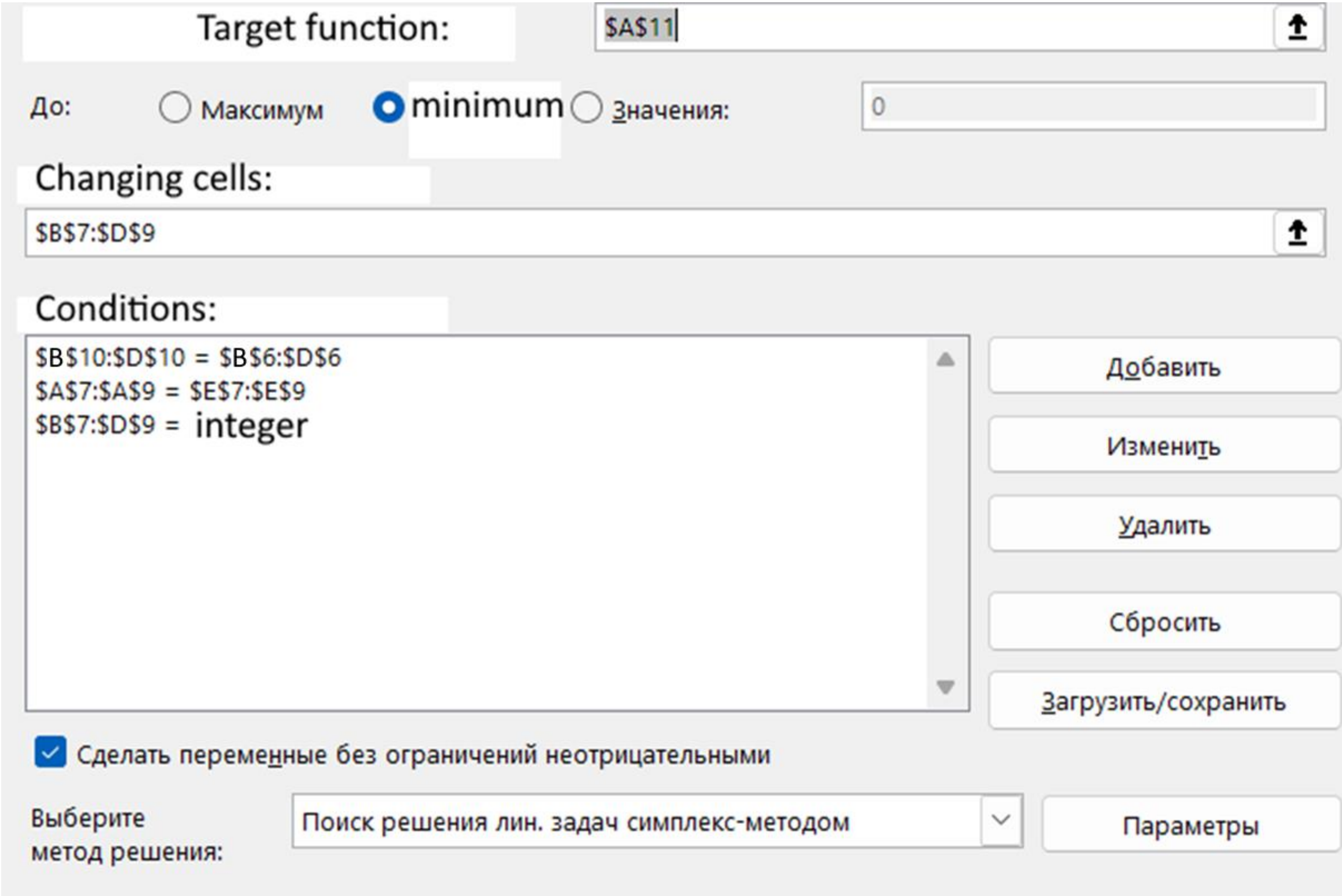
	A	B	C	D	E
1		110	350	140	
2	180	2	5	2	
3	300	7	7	13	
4	120	3	6	8	
5					

initial values for x_{ij}

6		110	350	140	
7	180	50	50	50	=sum
8	300	50	50	50	=sum
9	120	50	50	50	=sum
10		=sum	=sum	=sum	
11	=sumproducts(B2:D4;B7:D9)				

?

Search of solution:



Target function: **A11**
Target: **minimum**
Changing **B7 : D9**

Conditions:

B10 : D10 = B6 : D6 **need of each store**
A7 : A9 = E7 : E9 **amount at warehouse**

B7 : D9 = integer
B7 : D9 >=0

Solution:

	110	350	140	
180	0	40	140	180
300	0	300	0	300
120	110	10	0	120
	110	350	140	
2970				

Выберите метод решения: Поиск решения лин. задач симплекс-методом

Параметры

We supposed above that $\Sigma_i \, a_i = \Sigma_j \, b_j$ (*)

If not, then we can add extra an Shop or Warehouse to meet condition (*) :

Example: needs are less than reserves

Shops / Warehouses	Shop 1 b_1=110	Shop 2 b_2=350	Shop 3 b_3=40
Warehouse 1: a_1=180	2	5	2
Warehouse 2: a_2= 300	7	7	13
Warehouse 3: a_3= 120	3	6	8

Shops / Warehouses	Shop 1 b_1=110	Shop 2 b_2=350	Shop 3 b_3=40	Fictituous shop b_3=100
Warehouse 1: a_1=180	2	5	2	0
Warehouse 2: a_2= 300	7	7	13	0
Warehouse 3: a_3= 120	3	6	8	0

Contribution of the last column to total cost of delivery will be zero. Therefore, the obtained solution will be optimized from the viewpoint of Shops 1 – 3. $x_{i,\text{ fictituous}}$ **must remain at warehouses.**

Example: needs are larger than reserves

Shops / Warehouses	Shop 1 b_1=110	Shop 2 b_2=350	Shop 3 b_3=<u>240</u>
Warehouse 1: a_1=180	2	5	2
Warehouse 2: a_2= 300	7	7	13
Warehouse 3: a_3= 120	3	6	8

	Shop 1 b_1=110	Shop 2 b_2=350	Shop 3 b_3=240
Warehouse 1: a_1=180	2	5	2
Warehouse 2: a_2= 300	7	7	13
Warehouse 3: a_3= 120	3	6	8
Fictituous warehouse a_4= 100	0	0	0

Contribution of the last row/line to total cost of delivery will be zero. Therefore, the obtained solution will be optimized from the viewpoint of Shops 1 – 3.

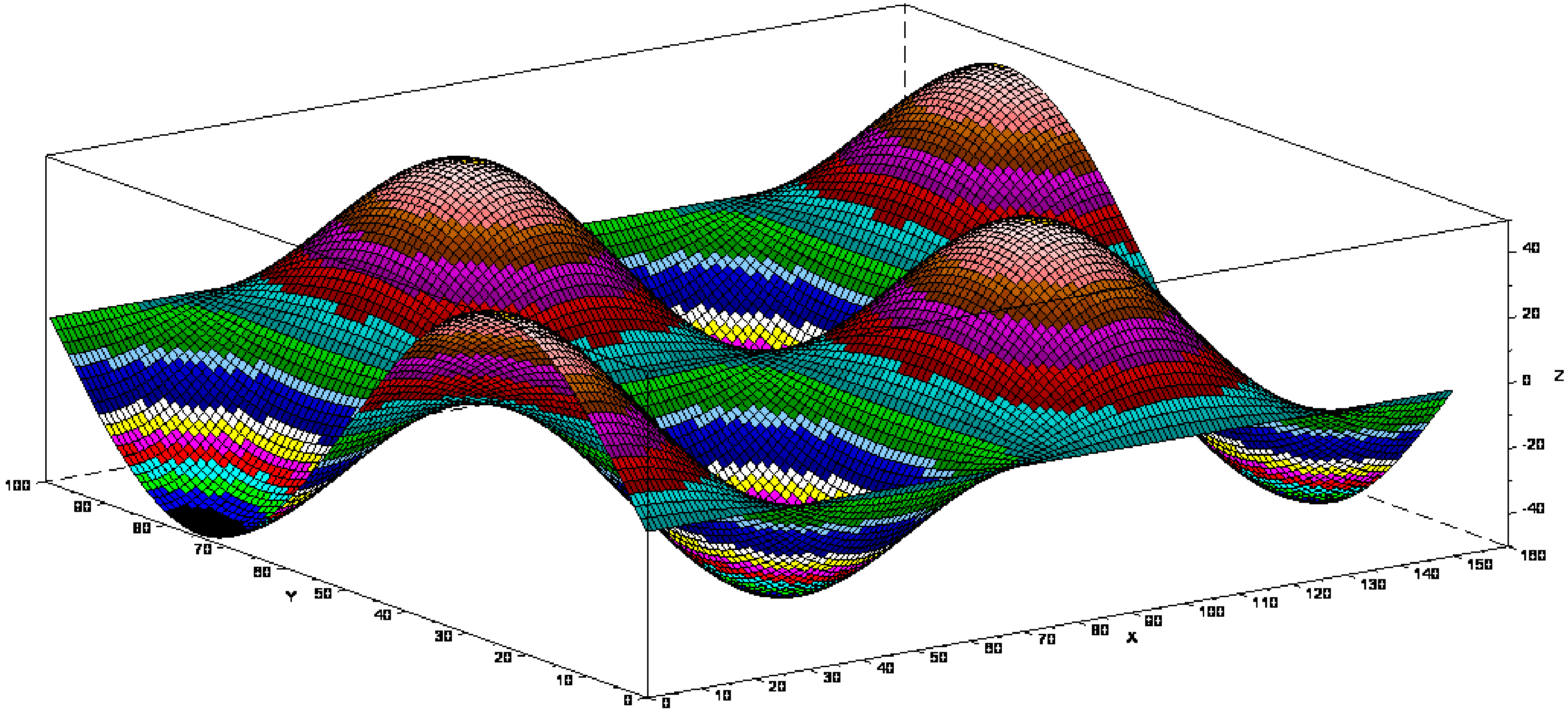
Last line will show deficit/undersupplies x_{4j}

9-4. Solving nonlinear optimization problems under constraints using penalty method

penalty = 罚款

In Section 9-3, we assumed that the cost of delivery is proportional to the number of goods delivered. Actually, this dependence is nonlinear.

In the case of <u>nonlinear functions</u> with constraints, the problem of finding a minimum becomes more difficult.



We can use gradient methods considered in Chapter 8

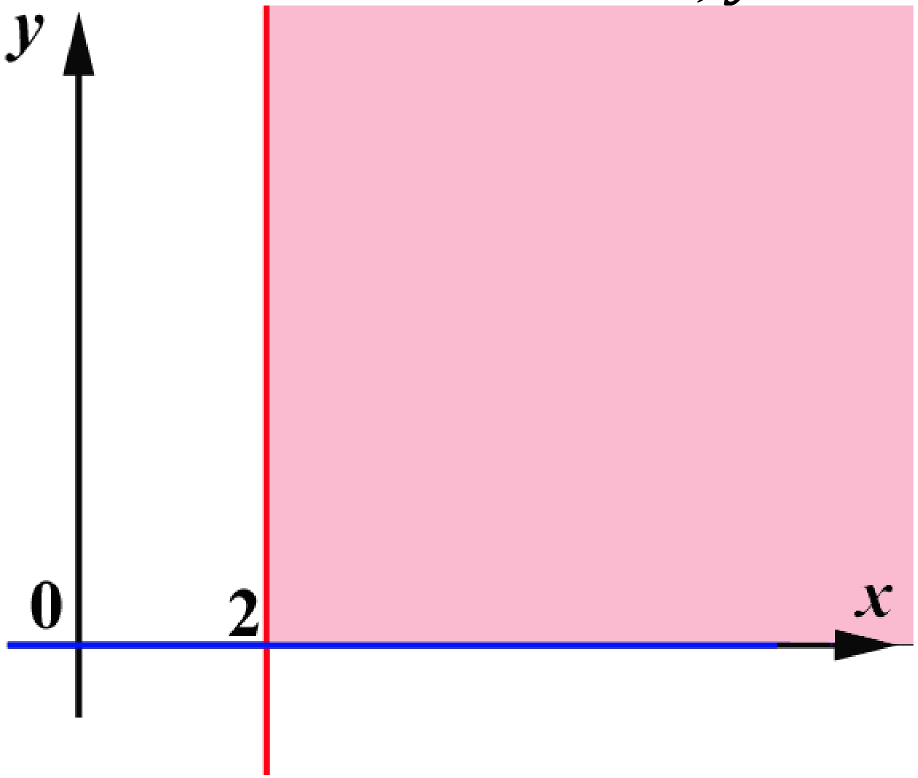
$$x^{(k+1)} = x^{(k)} - h \cdot \mathrm{grad}\ F(x^{(k)})$$

but we must modify them to retain $x^{(k+1)}$ **<u>inside</u> the given domain.**

The idea of the method of fines is to introduce an extra function $P(x, y)$ **that is nearly zero in the given domain, except for a vicinity of the boundary where it <u>increases abruptly</u>.**

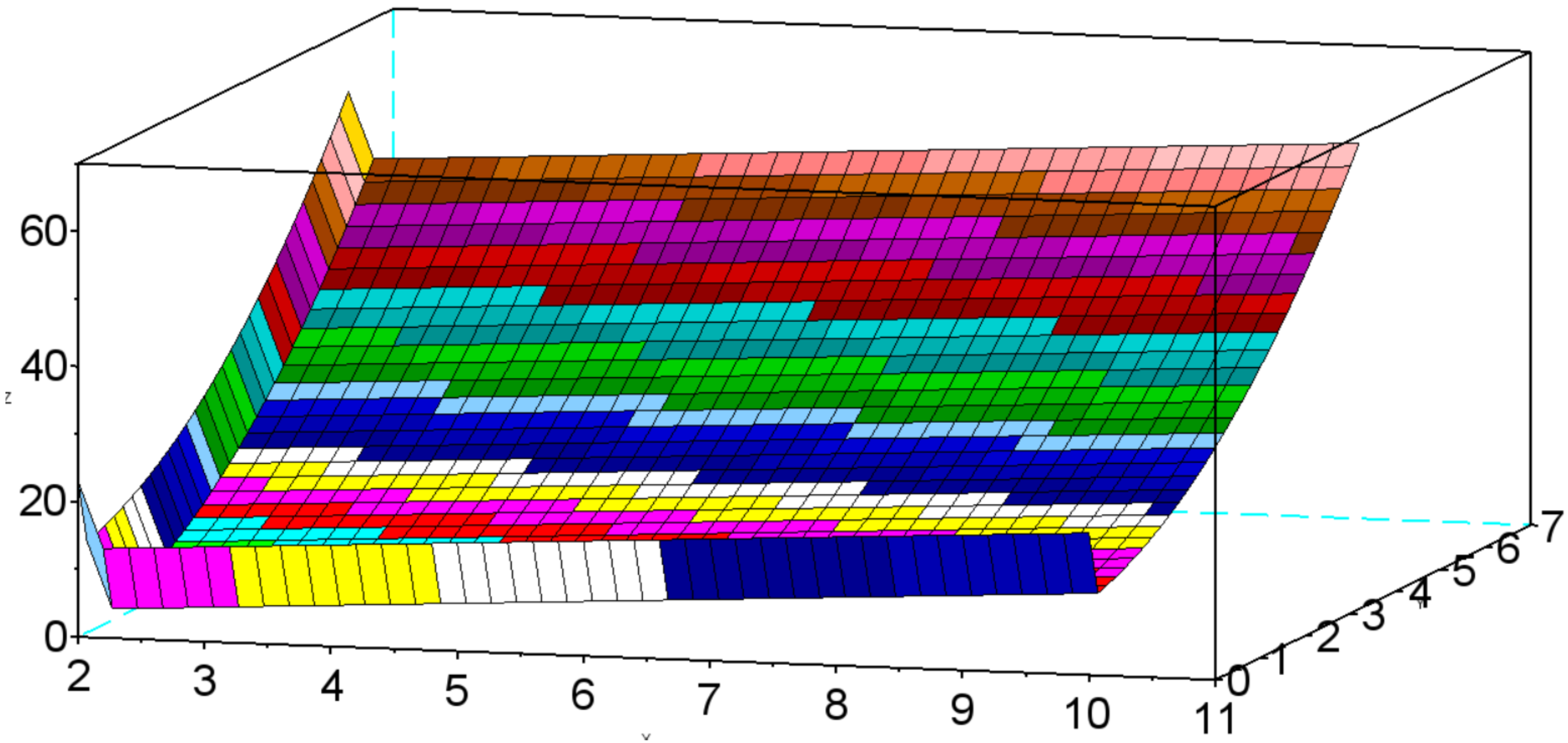
This prevents $x^{(k+1)}$ **from crossing the boundary.**

Example. Find a minimum of the function $f(x, y) = x + (y+1)^2$ under constraints $x \geq 2,\ y \geq 0$.



Let us choose a penalty function : $P(x, y) = 0.0001 \times [\,1/(x-2) + 1/y\,]$ and add it to $f(x, y)$:

$$F(x, y) = x + (y+1)^2 + 0.0001 \times [\,1/(x-2) + 1/y\,]$$



60
40
z
20
0
2
3
4
5
6
7
8
9
10
11
x
0
1
2
3
4
5
6
7

```
clear
for i=1:41
for j=1:31
x(i,j)=0.2*(i-1)+2+0.00001 ;
y(i,j)=0.2*(j-1)+0.00001 ;
end
end
F=x+(y+1).^2 + 0.0001*(1./(x-2) +1./y) ;
//surf(x,y,F)
// Using Gradient descent:
h=0.006
xx=5 ;
yy=5 ;
for k=1:500
```

```
dFdx= 1 -0.0001/(xx-2)^2 ;
dFdy= 2*(yy+1) - 0.0001/yy^2 ;
xx=xx-h*dFdx
yy=yy-h*dFdy
disp(k,xx,yy)
plot(xx,yy,'o')
end
```

General formulation:

If a minimum of a function $f(x_1, x_2, \ldots, x_n)$ **is to be found under** m **constraints**

$$g_i(x_1, x_2, \ldots, x_n) \geq 0, \quad i=1, 2, \ldots, m$$

then it makes sense to consider the function

$$F(x_1, x_2, \ldots, x_n) = f(x_1, x_2, \ldots, x_n) + P(x_1, x_2, \ldots, x_n)$$

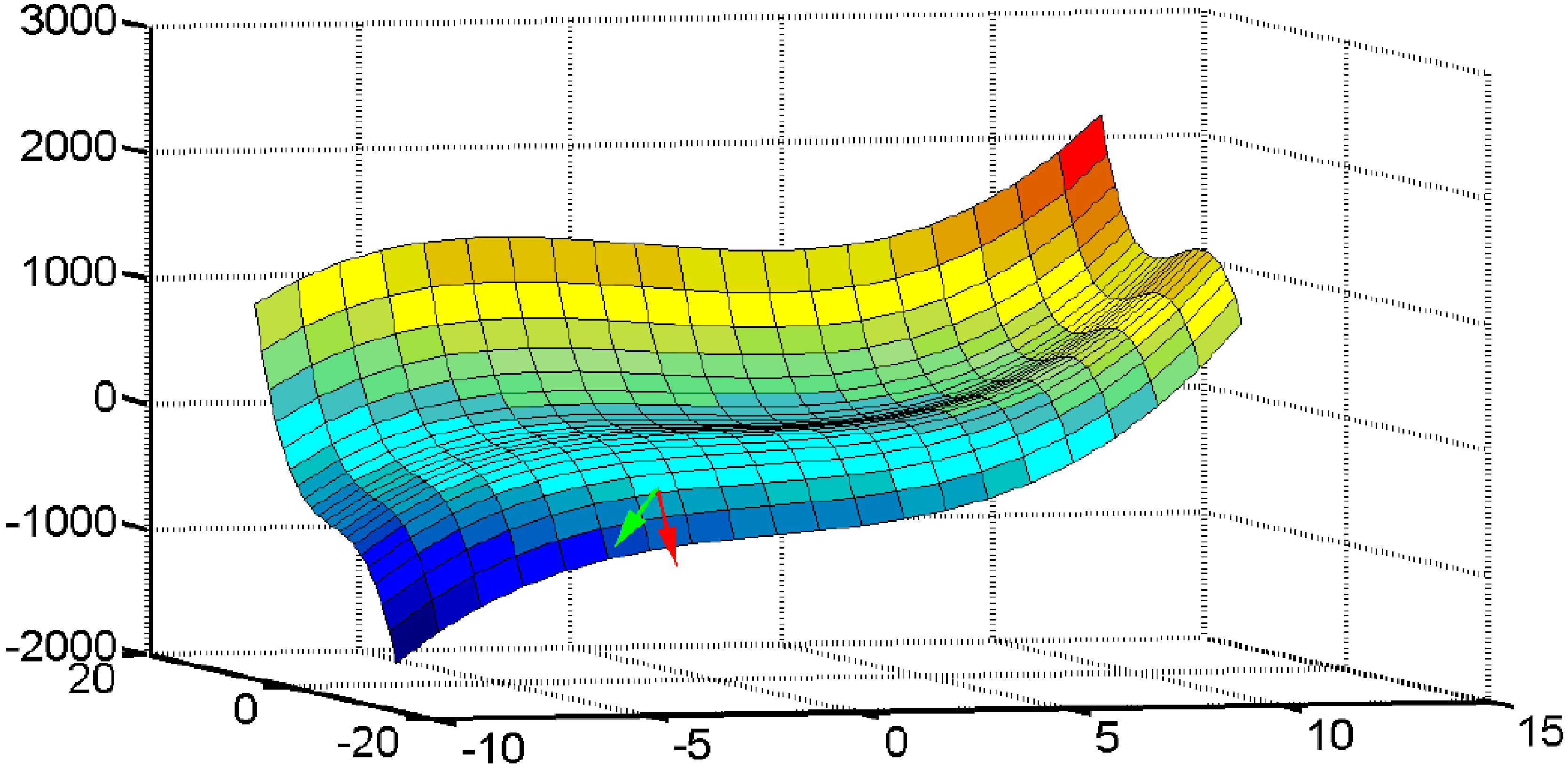
where penalty P **can be chosen as follows:**

$$P(x_1, x_2, \ldots, x_n) = r \sum_{i=1}^{m} [\, 1/g_i(x_1, x_2, \ldots, x_n) \,]$$

Method of admissible directions of gradient descent

Idea: a direction of next step of gradient descent

$x^{(k+1)} = x^{(k)} - h\cdot\mathrm{grad}\ F(x^{(k)})$ **is not allowed if the point** $x^{(k+1)}$ **gets outside of the domain specified by constraints.**



EXCEL

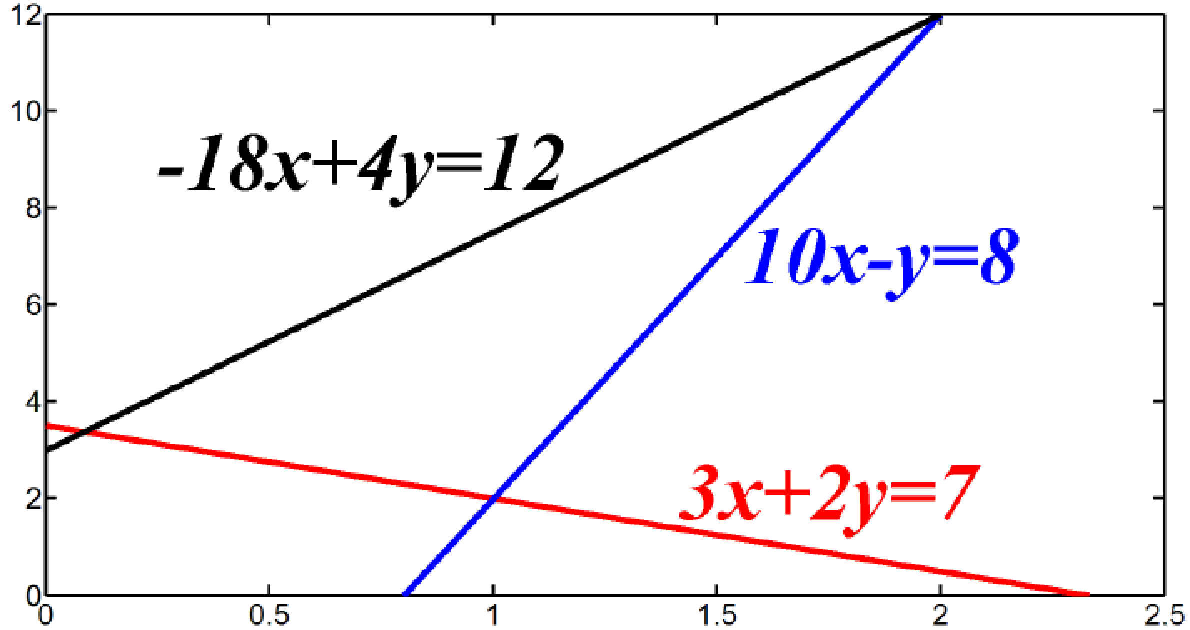
Solution of an optimization problem can be obtained with the extension "Solver".

Example. Find a minimum of function

$$f(x,y) = (x-3)^2 + (y-4)^2$$

under constraints

$$3x + 2y \geq 7$$
$$10x - y \leq 8$$
$$-18x + 4y \leq 12, \qquad x \geq 0, \qquad y \geq 0$$



-18x+4y=12
10x-y=8
3x+2y=7
12
10
8
6
4
2
0
0
0.5
1
1.5
2
2.5

	A	B	C
1	1	4	=(A1-3)^2+(B1-4)^2
2			
3			
4	=3*A1+2*B1		
5	=10*A1 - B1		
6	=-18*A1+ 4*B1		

Matlab

One of the available in Matlab commands for solving optimization problems under constraints is fmincon

<u>Example.</u> Find a <u>maximum</u> of the function

$f(x)=x_1 \cdot x_2 \cdot x_3$

under linear constraints

$0 \le x_1+2x_2+2x_3 \le 72$

Initial point:

x= [10; 10; 10]

First, create file myfun.m which determines the target function:

```
function f = myfun(x)
f = -x(1)*x(2)*x(3);
endfunction
```

Then we rewrite constraints in the form “ $\leq$ ”

$-x_1-2x_2-2x_3 \leq 0$

$x_1+2x_2+2x_3 \leq 72$

In the matrix form this can be written as Ax $\leq$ b

```
>> A = [-1 -2 -2; 1 2 2];
>> b = [0; 72];
```

```
>> x0 = [10; 10; 10];
>> [x,fval] = fmincon('myfun',x0,A,b)
```

Answer:

x = 24.0000 , 12.0000 , 12.0000

fval =-3.4560e+003

P.S. If constraints are nonlinear, then there is need to use more available options in fmincon