

تمرین 1 تحقیق در عملیات

فاطمه ناسیان ۱۱۰۱۹۸۴۷۹

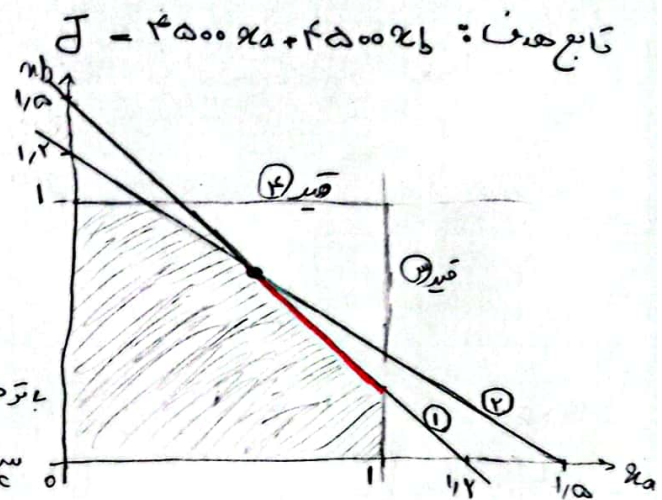
A: ۵۰۰۰ \$ → ۴۰۰ h → ۴۵۰۰ \$ سود
B: ۴۰۰۰ \$ → ۵۰۰ h → ۴۵۰۰ \$ سود

→ ۴۰۰۰ \$
۴۰۰ h

①

چون هزینه در کم سود و سرمایه و زمان دو ضریب x_a و x_b خواهد بود

- قیدها \Rightarrow
- ① $5000 x_a + 4000 x_b = 4000$
 - ② $400 x_a + 500 x_b \leq 400$
 - ③ $0 \leq x_a \leq 1$
 - ④ $0 \leq x_b \leq 1$



① قید $\Rightarrow 5x_a + 4x_b = 4 \rightarrow \begin{bmatrix} 5 \\ 4 \end{bmatrix}, \begin{bmatrix} 4 \\ 0 \end{bmatrix}$
 ② قید $\Rightarrow 4x_a + 5x_b \leq 4 \rightarrow \begin{bmatrix} 4 \\ 5 \end{bmatrix}, \begin{bmatrix} 0 \\ 4 \end{bmatrix}$

با توجه به شکل خط هدف و خط تقاطع می‌توانیم نتیجه بگیریم که جواب در آن قرار دارد
 پس محل تلاقی این دو خط جایی است که تابع هدف ماکسیمم می‌شود

$$\begin{cases} 5x_a + 4x_b = 4 \\ 4x_a + 5x_b = 4 \end{cases} \rightarrow x_a = \frac{4}{3}, x_b = \frac{4}{3}$$

مقدار تابع هدف $\Rightarrow J = \frac{4}{3} \times 4500 + \frac{4}{3} \times 4500 = 4000$ \$
 به ازای مقایسه x_a و x_b سود تحقیق می‌شود

maximize $2x_1 - x_2 + 2x_3$ \rightarrow minimize $-2x_1 + x_2 - 2x_3$

Subject to

- ① $x_1 + x_2 \leq 10$
- ② $x_1 + 2x_2 - 2x_3 \leq 2$
- ③ $x_2 + 2x_3 \leq 5$

where $x_1, x_2, x_3 \geq 0$

\Rightarrow

$$\begin{cases} 2x_1 + x_2 + x_4 = 10 \rightarrow x_4 \\ x_1 + 2x_2 - 2x_3 + x_5 = 2 \rightarrow x_5 \\ x_2 + 2x_3 + x_6 = 5 \rightarrow x_6 \end{cases}$$

③

	x_4	x_5	x_6	x_1	x_2	x_3	RHS
-2	0	0	-2	1	+1	0	0
x_4	1	0	0	2	1	0	10
x_5	0	1	-2	1	2	0	2
x_6	0	0	2	0	1	1	5

	x_4	x_5	x_6	x_1	x_2	x_3	RHS
-2	0	0	0	-2	+2	+1	+5
x_4	1	0	0	2	1	0	10
x_5	0	1	0	1	2	0	2
x_6	0	0	1	0	1	1	5

در این مرحله x_4 خارج می‌شود

	x_1	x_2	x_3	x_4	x_5	x_6	RHS
-Z	-1	0	0	0	1	1	0
x_1	1	0	0	1	1	0	10
x_2	1	1	0	0	1	1	1
x_3	0	0	1	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{0}{2}$

	x_1	x_2	x_3	x_4	x_5	x_6	RHS
-Z	0	0	0	1	1	1	10
x_1	1	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0	10
x_2	0	1	0	$-\frac{1}{2}$	$\frac{0}{2}$	1	1
x_3	0	0	1	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{0}{2}$

$$\Rightarrow \begin{matrix} x_1^* = 0 \\ x_2^* = 1 \\ x_3^* = \frac{0}{2} \end{matrix} \quad \begin{matrix} x_4^* = 0 \\ x_5^* = 0 \\ x_6^* = 0 \end{matrix}$$

$$\Rightarrow \text{Max } J^* = 10$$

Minimize $1x_1 + 1x_2 + 1x_3$

Subject to ① $1x_1 + 1x_2 + 1x_3 = 40$

② $1x_1 + 1x_2 + 0x_3 \geq 10$

where $x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$

① $1x_1 + 1x_2 + 1x_3 = 40 \rightarrow x_3$

② $1x_1 + 1x_2 + 0x_3 - x_4 = 10 \rightarrow x_4$

$x_i \geq 0$

	x_1	x_2	x_3	x_4	RHS
-Z	1	1	1	0	0
x_1	1	1	1	0	40
x_2	1	0	1	-1	10

	x_1	x_2	x_3	x_4	RHS
-Z	0	-1	-1	0	10
x_1	1	1	1	0	40
x_2	0	-1	-1	-1	-10

	x_1	x_2	x_3	x_4	RHS
-Z	0	0	0	0	-90
x_1	1	0	$-\frac{1}{2}$	$-\frac{1}{2}$	10
x_2	0	1	$\frac{1}{2}$	$\frac{1}{2}$	10

$$\Rightarrow \begin{cases} x_1^* = 0 \\ x_2^* = 0 \\ x_3^* = 10 \\ x_4^* = 10 \end{cases} \quad J^*_{min} = -90$$

2) همانگونه که در ویدیو آموزشی دیدیم به صورت زیر متغیر ها ، قیود و تابع هدف را تعریف میکنیم. معادلات را در سوال 1 به دست آورده بودیم. سپس نوع بهینه سازی را ماکسیمایز انتخاب کرده و solver را نیز انتخاب میکنیم و نتایج را نشان می دهیم.

```
1 import pyomo.environ as pyo
2 from pyomo.environ import *
3 from pyomo.opt import SolverFactory
4
5 model = pyo.ConcreteModel()
6
7 model.xa = pyo.Var(bounds=(0,1))
8 model.xb = pyo.Var(bounds=(0,1))
9
10 xa = model.xa
11 xb = model.xb
12
13 model.c1 = pyo.Constraint(expr= 5000*xa+4000*xb==6000)
14 model.c2 = pyo.Constraint(expr= 400*xa+500*xb<=600)
15
16 model.obj = pyo.Objective(expr= 4500*xa+4500*xb, sense=maximize)
17
18 opt = SolverFactory('glpk')
19 opt.solve(model)
20
21 model.pprint()
22
23 xaval = pyo.value(xa)
24 xbval = pyo.value(xb)
25
26 print('xa = ',xaval)
27 print('xb = ',xbval)
28
29 print('maximize J = ',4500*xaval+4500*xbval)
```

نتیجه بهینه سازی به صورت زیر می شود که این اطلاعات را خود مدل در اختیار ما میگذارد.

2 Var Declarations

```
xa : Size=1, Index=None
  Key : Lower : Value          : Upper : Fixed : Stale : Domain
  None :      0 : 0.6666666666666666 :      1 : False : False : Reals
xb : Size=1, Index=None
  Key : Lower : Value          : Upper : Fixed : Stale : Domain
  None :      0 : 0.6666666666666667 :      1 : False : False : Reals
```

1 Objective Declarations

```
obj : Size=1, Index=None, Active=True
  Key : Active : Sense      : Expression
  None :      True : maximize : 4500*xa + 4500*xb
```

2 Constraint Declarations

```
c1 : Size=1, Index=None, Active=True
  Key : Lower : Body          : Upper : Active
  None : 6000.0 : 5000*xa + 4000*xb : 6000.0 : True
c2 : Size=1, Index=None, Active=True
  Key : Lower : Body          : Upper : Active
  None : -Inf : 400*xa + 500*xb : 600.0 : True
```

5 Declarations: xa xb c1 c2 obj

حال مقادیر xa و xb را فراخوانی میکنیم و مقدار تابع هدف را محاسبه میکنیم.

```
xa = 0.6666666666666666
xb = 0.6666666666666667
maximize J = 5999.999999999998
```

5) دو متغیر x و y داریم و برای مثلث مد نظر، باید دو شرط را در نظر بگیریم. شرط اول نا منفی بودن اضلاع و شرط دوم این است که مجموع دوضلع از ضلع دیگر بزرگتر باشد. به این ترتیب 6 قید داریم و تابع هدف را نیز تعریف میکنیم و پاسخ را برای آن محاسبه میکنیم.

در حالت اول تابع هدف را ماکسیمایز میکنیم.

```
model = pyo.ConcreteModel()

model.x = pyo.Var()
model.y = pyo.Var()

x = model.x
y = model.y

model.c1 = pyo.Constraint(expr= 8-x-0.5*y>=0)
model.c2 = pyo.Constraint(expr= 7+x-1.5*y>=0)
model.c3 = pyo.Constraint(expr= 1+x>=0)
model.c4 = pyo.Constraint(expr= 8-x-0.5*y+7+x-1.5*y>=1+x)
model.c5 = pyo.Constraint(expr= 8-x-0.5*y+1+x>=7+x-1.5*y)
model.c6 = pyo.Constraint(expr= 7+x-1.5*y+1+x>=8-x-0.5*y)

model.obj = pyo.Objective(expr= 16+x-2*y, sense=maximize)

opt = SolverFactory('glpk')
opt.solve(model)

model.pprint()

xval = pyo.value(x)
yval = pyo.value(y)
print('x = ',xval)
print('y = ',yval)
print('Triangle Districts = {} , {} , {}'.format(8-xval-0.5*yval , 7+xval-1.5*yval , 1+xval ))
print('maximize J = ',16+xval-2*yval)
```

خروجی solver به صورت زیر است:

```

2 Var Declarations
  x : Size=1, Index=None
    Key : Lower : Value : Upper : Fixed : Stale : Domain
    None : None : -1.0 : None : False : False : Reals
  y : Size=1, Index=None
    Key : Lower : Value : Upper : Fixed : Stale : Domain
    None : None : -3.0 : None : False : False : Reals

1 Objective Declarations
  obj : Size=1, Index=None, Active=True
    Key : Active : Sense : Expression
    None : True : maximize : 16 + x - 2*y

6 Constraint Declarations
  c1 : Size=1, Index=None, Active=True
    Key : Lower : Body : Upper : Active
    None : 0.0 : 8 - x - 0.5*y : +Inf : True
  c2 : Size=1, Index=None, Active=True
    Key : Lower : Body : Upper : Active
    None : 0.0 : 7 + x - 1.5*y : +Inf : True
  c3 : Size=1, Index=None, Active=True
    Key : Lower : Body : Upper : Active
    None : 0.0 : 1 + x : +Inf : True
  c4 : Size=1, Index=None, Active=True
    Key : Lower : Body : Upper : Active
    None : -Inf : 1 + x - (8 - x - 0.5*y + 7 + x - 1.5*y) : 0.0 : True
  c5 : Size=1, Index=None, Active=True
    Key : Lower : Body : Upper : Active
    None : -Inf : 7 + x - 1.5*y - (8 - x - 0.5*y + 1 + x) : 0.0 : True
  c6 : Size=1, Index=None, Active=True
    Key : Lower : Body : Upper : Active
    None : -Inf : 8 - x - 0.5*y - (7 + x - 1.5*y + 1 + x) : 0.0 : True

9 Declarations: x y c1 c2 c3 c4 c5 c6 obj

```

حال مقادیر x و y و اضلاع و محیط را نمایش میدهیم:

```

x = -1.0
y = -3.0
Triangle Districts = 10.5 , 10.5 , 0.0
maximize J = 21.0

```

در حالت دوم تابع هدف را مینیمم میکنیم:

```

2 Var Declarations
  x : Size=1, Index=None
    Key : Lower : Value : Upper : Fixed : Stale : Domain
    None : None : 2.0 : None : False : False : Reals
  y : Size=1, Index=None
    Key : Lower : Value : Upper : Fixed : Stale : Domain
    None : None : 6.0 : None : False : False : Reals

1 Objective Declarations
  obj : Size=1, Index=None, Active=True
    Key : Active : Sense : Expression
    None : True : minimize : 16 + x - 2*y

6 Constraint Declarations
  c1 : Size=1, Index=None, Active=True
    Key : Lower : Body : Upper : Active
    None : 0.0 : 8 - x - 0.5*y : +Inf : True
  c2 : Size=1, Index=None, Active=True
    Key : Lower : Body : Upper : Active
    None : 0.0 : 7 + x - 1.5*y : +Inf : True
  c3 : Size=1, Index=None, Active=True
    Key : Lower : Body : Upper : Active
    None : 0.0 : 1 + x : +Inf : True
  c4 : Size=1, Index=None, Active=True
    Key : Lower : Body : Upper : Active
    None : -Inf : 1 + x - (8 - x - 0.5*y + 7 + x - 1.5*y) : 0.0 : True
  c5 : Size=1, Index=None, Active=True
    Key : Lower : Body : Upper : Active
    None : -Inf : 7 + x - 1.5*y - (8 - x - 0.5*y + 1 + x) : 0.0 : True
  c6 : Size=1, Index=None, Active=True
    Key : Lower : Body : Upper : Active
    None : -Inf : 8 - x - 0.5*y - (7 + x - 1.5*y + 1 + x) : 0.0 : True

9 Declarations: x y c1 c2 c3 c4 c5 c6 obj

```

مقادیر اضلاع به صورت زیر می شود:

```

x = 2.0
y = 6.0
Triangle Districts = 3.0 , 0.0 , 3.0
minimize J = 6.0

```