Lab05-Linear Programming

CS214-Algorithm and Complexity, Xiaofeng Gao, Spring 2019.

∗ If there is any problem, please contact TA Jiahao Fan.
∗ Name:田雪飞 Student ID:515030910347 Email: 13487426939@qq.com

1. **Solution.** The following is solution.

(a) we assume that we invest x_1 in project1 and x_2 in project2 at beginning of 2018. Then we can take out $(1+20\%)x_1$ and we can not take out money from project2. then we use the money we take out from project1 to invest to project1 and project3. and best strategy is to invest 100000 to project3 and invest $1.2x_1 - 100000$ dollars. and at the end of 2019, we can take out $1.2x_1 - 100000$ form project1, $1.5x_2$ form project2 and 140000,then we can formulate a programming:

$$\begin{cases} max\_amount = 1.44x_1 + 1.5x_2 + 20000 \\ x_2 \leq 150000 \\ x_1 + x_2 = 300000 \\ x_1, x_2 \geq 0 \end{cases}$$

(b) The standard form is:

$$\begin{cases} max\_amount = 1.44x_1 + 1.5x_2 + 20000 \\ x_2 \leq 150000 \\ x_1 + x_2 \leq 300000 \\ -x_1 - x_2 \leq -300000 \\ x_1, x_2 \geq 0 \end{cases}$$

The slack form is:

$$\begin{cases} max\_amount = 1.44x_1 + 1.5x_2 + 20000 \\ x_1 + x_2 = 300000 \\ x_2 + x_3 = 150000 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

(c)

max $1.44x_1 + 1.5x_2 + 20000$

s.t.

$$\begin{aligned} x_2 &\leq 150000 \\ x_1 + x_2 &\leq 300000 \\ -x_1 - x_2 &\leq -300000 \\ x_1, x_2 &\leq 0 \end{aligned}$$

Multiplier	Constraint
y_1	$x_1 + x_2 \leq 300000$
y_2	$x_2 \leq 150000$
y_1	$-x_1 - x_2 \leq -300000$

Then we can know: $1.44x_1 + 1.5x_2 \leq 300000y_1 + 150000y_2$

and the constraint us following:

$$\begin{cases} y_1 \geq 1.44 \\ y_1 + y_2 \geq 1.5 \\ y_1, y_2 \geq 0 \end{cases}$$

So,the dual form is:

$$\begin{cases} min = 1.44x_1 + 1.5x_2 \leq 300000y_1 + 150000y_2 \\ y_1 \geq 1.44 \\ y_1 + y_2 \geq 1.5 \\ y_1, y_2 \geq 0 \end{cases}$$

(d) According to initial formula,we can know $x_2$150000,amount is maximum.

$$max\_amount = 1.44 * 150000 + 1.5 * 150000 + 20000 = 461000 \tag{1}$$

Then maximum amount we can get at the end of 2019 is 0.461 million dollars.

□

2. **Solution.** The following is solution.

(a) Optimization Programming Language:

$X_{ij} - -$ the number of product i produced in month j.
$R_{ij} - -$ the number of product i stored in month j.
$SC_{ij} - -$ the largest number of product i sold in month j.
$M_{kj} - -$ the number of device k used in month j.
$T_{ik} - -$ the time of product i used in device k.
$P_i - -$ the profit of product i
constant $- - -$ the hours of each month $t = 8 * 2 * 24 = 384(hours)$.
mark: $i = 1, 2 \ldots 7, \quad j = 1, 2 \ldots 6, \quad k = 1, 2 \ldots 5.$

Then according to the constraints:

Maximum stock(100): $R_{ij} \leq 100$
Stock in June(50): $R_{i6} = 50$
Jan: $X_{i1} - R_{i1} \leq SC_{i1}$,$R_{i1} = 0$,$i = 1, 2 \ldots 7$.
Feb-Jun:$X_{ij} + R_{ij-1} - R_{ij} \leq SC_{ij}, i = 1, 2 \ldots 7, j = 2, 3 \ldots 6$.
Time limited by devices: $\sum_{j=1}^{6} \sum_{i=1}^{7} T_{ik} X_{ij} \leq \sum_{j=1}^{6} M_{kj} * 384, k = 1, 2 \ldots 5$

We assume that the total profits are MAX:
so there have these relations:

$$\begin{cases} MAX = \sum_{i=1}^{7} \sum_{j=1}^{6} (X_{ij} P_i - 0.5 R_{ij}) - 50 \sum_{i=1}^{7} P_i \\ R_{ij} \leq 100 \\ R_{ij} \geq 0 \\ R_{i6} = 50 \\ i = 1, 2 \ldots 7, j = 1, 2 \ldots 6 \\ X_{i1} - R_{i1} \leq SC_{i1} \text{ ,} R_{i1} = 0, i = 1, 2 \ldots 7 \\ X_{ij} + R_{ij-1} - R_{ij} \leq SC_{ij}, i = 1, 2 \ldots 7, j = 2, 3 \ldots 6 \\ \sum_{j=1}^{6} M_{kj} = (20, 10, 15, 5, 5), k = 1, 2 \ldots 5 \\ \sum_{j=1}^{6} \sum_{i=1}^{7} T_{ik} X_{ij} \leq \sum_{j=1}^{6} M_{kj} * 384, k = 1, 2 \ldots 5 \end{cases}$$

The .mod and .dat documents in the compressed file(.zip).
The following is output of solution in OPL.

```
// solution (optimal) with objective 108855
// Quality Incumbent solution:
```

```
// MILP objective                                  1.0885500000e+05
// MILP solution norm |x| (Total, Max)             1.42440e+04  1.15000e+03
// MILP solution error (Ax=b) (Total, Max)         0.00000e+00  0.00000e+00
// MILP x bound error (Total, Max)                 0.00000e+00  0.00000e+00
// MILP x integrality error (Total, Max)           0.00000e+00  0.00000e+00
// MILP slack bound error (Total, Max)             1.36424e-12  2.27374e-13
//

pro_x = [[500
             600 400 0 -1.1369e-13 550]
             [1000 500 700 0 100 550]
             [300 200 100 0 500 150]
             [300 0 100 0 100 350]
             [800 400 600 0 1000 1150]
             [200 300 400 0 300 550]
             [100 150 200 0 0 110]];
pro_r = [[0 0 100 0 -1.1369e-13 50]
             [0 -1.1369e-13 100 0 -1.1369e-13 50]
             [0 0 100 0 0 50]
             [0 0 100 0 -1.1369e-13 50]
             [0 -1.1369e-13 100 0 0 50]
             [0 0 0 0 0 50]
             [0 0 100 0 0 50]];
num_device = [[2 2 2 1 1 2]
             [1 1 1 1 1 1]
             [1 1 1 2 2 1]
             [1 1 1 0 1 1]
             [1 1 1 0 1 1]];
```

(b)

i. A.

```
         num_device = [[2 2 2 1 1 2]
             [1 1 1 1 1 1]
             [1 1 1 2 2 1]
             [1 1 1 0 1 1]
             [1 1 1 0 1 1]];
```

ii. A.

```
  pro_x = [[500 600 400 0 0 550]
             [1000 500 700 0 100 550]
             [300 200 100 0 500 150]
             [300 0 100 0 100 350]
             [800 400 600 0 1000 1150]
             [200 300 400 0 300 550]
             [100 150 200 0 0 110]];
```

B.

```
sell=pro_x+pro_r=[[500,600,300,200,0,500],
  [1000,500,600,300,100,500],
  [300,200,0,400,500,100],
 [300,0,0,500,100,300],
 [800,400,500,200,1000,1100],
 [200,300,400,0,300,500],
 [100,150,100,100,0,60]];
```

C.

```
pro_r = [[0 0 100 0 0 50]
         [0 0 100 0 0 50]
         [0 0 100 0 0 50]
         [0 0 100 0 0 50]
         [0 0 100 0 0 50]
         [0 0 0 0 0 50]
         [0 0 100 0 0 50]];
```

iii. The total selling profit: 108380 £
iv. The total holding cost: 475 £
v. The total net profit: 108855 £

□

Remark: You need to include your .mod, .dat, .pdf and .tex files in your uploaded .zip file.