

一、例题引入

【例】已知模型如下，其中 $a_i = i$, $i = 1, 2, \dots, 5$ ，请编程求解：

$$\begin{aligned} & \max S \\ & s.t. \begin{cases} S = a_i x_i, & i = 1, 2, \dots, 5 \\ \sum_{i=1}^5 x_i = 5000 \end{cases} \end{aligned}$$

【解】暴力枚举法：

① 先分解一下上式：

$$\begin{aligned} & \max S \\ & s.t. \begin{cases} S = a_1 x_1 \\ S = a_2 x_2 \\ S = a_3 x_3 \\ S = a_4 x_4 \\ S = a_5 x_5 \\ x_1 + x_2 + x_3 + x_4 + x_5 = 5000 \end{cases} \end{aligned}$$

② Lingo 解决：

```
a1=1;
a2=2;
a3=3;
a4=4;
a5=5;
```

```
max = s;
s = a1*x1;
s = a2*x2;
s = a3*x3;
s = a4*x4;
s = a5*x5;
x1+x2+x3+x4+x5=5000;
```

③ 点评：

暴力枚举法较为笨拙，所以本次课将使用 Lingo 中非常重要的“集合”，亦或称之为——“矩阵工厂”。

二、矩阵工厂

2.1 矩阵工厂：生产一维矩阵

先来看看例子，当然不必在意其中的空格（Lingo 不读取空格）：

```
sets:  
factory /1..6/ : a,b;  
plant /1..3/ : x,y;  
endsets
```

以上程序对应知识点：

- ① **factory** 和 **plant** 都是制造矩阵的工厂，但它们是两家不同的工厂。
- ② **factory** 工厂后面的 **/1..6/** 说明它专门生产 1×6 的矩阵。
factory 工厂最后面出现的 a 和 b ，都是 1×6 的矩阵。
- ③ **plant** 工厂后面的 **/1..3/** 说明它专门生产 1×3 的矩阵。
plant 工厂最后面出现的 x 和 y ，都是 1×3 的矩阵。
- ④ 矩阵工厂的名字 **factory** 是随便起的，工厂所生产行矩阵的名字 a 和 b 也是随便起的。
- ⑤ 以上这四句话，本质是定义了四个行矩阵的大小，矩阵工厂只是中介。
- ⑥ 生产完矩阵后，工厂和矩阵之间将脱开联系。
- ⑦ Lingo 不是一行一行读代码的，所以用 **sets:** 和 **endsets** 表示矩阵工厂生产流程的起止。

【例 1】 阅读以下 Lingo 代码，请问 a 和 b 两个矩阵有联系吗？

```
sets:  
nanfu /1..6/ : a,b;  
endsets
```

【解】

没有特殊联系，只是 a 和 b 都是一行六列的矩阵。

【例 2】 阅读以下 Lingo 代码，请问代码可否简洁一点？

```
sets:  
ctgu /1..6/ : a;  
mcm /1..6/ : b;  
endsets
```

【解】

```
sets:  
easy /1..6/ : a, b;  
endsets
```

【例 3】阅读以下 Lingo 代码，请问有何问题？

```
sets:  
ceshi /1..6/ : apple, Apple;  
endsets
```

【解】

Lingo 不区分大小写，所以 apple 和 Apple 是同一个矩阵，应该换个名字。

2.2 矩阵的赋值

矩阵工厂不能只生产矩阵，还要给矩阵赋初值才行，例子如下：

```
sets:  
factory /1..6/ : a,b;  
plant /1..3/ : c,x;  
endsets
```

```
data:  
a = 1, 2, 3, 4, 5, 6;  
b = 6.0, 5.0, 4.0, 3.0, 2.0, 1.0;  
c = 10, 20, 30;  
enddata
```

以上程序对应以下知识点：

- ① 不是每个矩阵都要赋值，有些矩阵正是我们要求解的变量。
- ② 需要赋值的矩阵必须赋满，不能给 6 个元素的矩阵只赋 3 个数值。
- ③ Lingo 中可以给矩阵赋整数，也可以赋小数。
- ④ Lingo 不是一行一行读代码的，所以用 data:和 enddata 表示矩阵赋值的起止。

2.3 循环与求和

【例】已知模型如下，其中 $a_i = i$, $i = 1, 2, \dots, 5$ ，请编程求解：

$$\begin{aligned} & \max S \\ & s.t. \begin{cases} S = a_i x_i, & i = 1, 2, \dots, 5 \\ \sum_{i=1}^5 x_i = 5000 \end{cases} \end{aligned}$$

【for 循环】

题中约束条件： $S = a_i x_i$, $i = 1, 2, \dots, 5$ 可以利用 for 循环一步到位。

```
@for( gc(i) : a(i)*x(i) = S );
```

① for 循环，括起整行语句，因为 $S = a_i x_i$, $i = 1, 2, \dots, 5$ 相当于 5 个约束条件：

$$S = a_1 x_1$$

$$S = a_2 x_2$$

$$S = a_3 x_3$$

$$S = a_4 x_4$$

$$S = a_5 x_5$$

② for 循环内部，先写工厂，以告诉 for 循环几次，之后再上接约束条件。

③ 此处的 i 可带可不带，甚至可以换成 j 、 k 或 m 等等。

④ 二维矩阵工厂出现后，同时会出现 i 和 j ，那时必须带 i 和 j 。

【sum 求和】

题中约束条件： $\sum_{i=1}^5 x_i = 5000$ 可以利用 sum 求和一步到位。

```
@sum( gc(i) : x(i) ) = 5000;
```

① sum 求和，不可以括起完整的约束条件，因为一般的求和的结构是这样的：

$$x_1 + x_2 + x_3 + x_4 + x_5 = 5000$$

② sum 求和内部，先写工厂，以告诉 sum 求和几次，之后再上接约束条件。

③ 此处的 i 可带可不带。

④ 二维矩阵工厂出现后，同时会出现 i 和 j ，那时必须带。

【for 与 sum 出现的标志】

① 约束条件后面有 $i = 1, 2, \dots, 5$ ，一定在最外层套上 for。

② 约束条件前面是 $\sum_{i=1}^5 x_i$ ，一定在中间加上 sum。

【例】已知模型如下，其中 $a_i = i$, $i = 1, 2, \dots, 5$ ，请编程求解：

$$\begin{aligned} & \max S \\ & s.t. \begin{cases} S = a_i x_i, & i = 1, 2, \dots, 5 \\ \sum_{i=1}^5 x_i = 5000 \end{cases} \end{aligned}$$

程序如下：

```
model:
```

```
sets:
```

```
gc /1..5/ : a,x;
```

```
endsets
```

```
data:
```

```
a = 1,2,3,4,5;
```

```
enddata
```

```
max = S;
```

```
@for( gc(i) : a(i)*x(i) = S );
```

```
@sum( gc(i) : x(i) ) = 5000;
```

```
end
```

PS：使用了矩阵工厂创建矩阵后，整个程序需用 `model:`和 `end` 包起来。

三、工厂合并

3.1 工厂合并——生产二维矩阵

先来看看例子：

```
sets:  
factory /1..6/ : a;  
plant /1..8/ : d;  
Cooperation(factory,plant) : c, x;  
endsets
```

以上程序可以得到以下结论：

- ① Cooperation 大工厂是由 factory 和 plant 两家小工厂合并而办，可生产 6×8 的矩阵。
- ② a 是 1×6 的矩阵， d 是 1×8 的矩阵， c 和 x 都是 6×8 的矩阵。
- ③ 如果将 Cooperation(factory,plant) 中的 factory 与 plant 调换位置，则生产 8×6 的矩阵。
- ④ 工厂合并的名字 Cooperation 是随便起的，矩阵的名字 c 和 x 也是随便起的。

3.2 矩阵的赋值

```
data:  
c=6,2,6,7,4,2,5,8  
  4,9,5,3,8,5,8,2  
  5,2,1,9,7,4,3,3  
  7,6,7,3,9,2,7,1  
  2,3,9,5,7,2,6,5  
  5,5,2,2,8,1,4,3;  
enddata
```

3.3 例题

【例】请编程求解以下模型，已知条件如右侧所示：

$$\begin{aligned} \min \quad & z = \sum_{i=1}^6 \sum_{j=1}^8 c_{ij} \cdot x_{ij} \\ \text{s.t.} \quad & \begin{cases} \sum_{j=1}^8 x_{ij} \leq a_i, i=1,2,\dots,6 \\ \sum_{i=1}^6 x_{ij} = d_j, j=1,2,\dots,8 \\ x_{ij} \geq 0, i=1,2,\dots,6, j=1,2,\dots,8 \end{cases} \end{aligned}$$

$$\begin{aligned} a &= 60, 55, 51, 43, 41, 52 \\ d &= 35, 37, 22, 32, 41, 32, 43, 38 \\ c &= \begin{matrix} 6, 2, 6, 7, 4, 2, 5, 8 \\ 4, 9, 5, 3, 8, 5, 8, 2 \\ 5, 2, 1, 9, 7, 4, 3, 3 \\ 7, 6, 7, 3, 9, 2, 7, 1 \\ 2, 3, 9, 5, 7, 2, 6, 5 \\ 5, 5, 2, 2, 8, 1, 4, 3 \end{matrix} \end{aligned}$$

程序如下：

```
model:
sets:
factory /1..6/ : a;
plant /1..8/ : d;
Cooperation(factory,plant) : c, x;
endsets

data:
a=60,55,51,43,41,52;
d=35,37,22,32,41,32,43,38;
c=6,2,6,7,4,2,5,8
  4,9,5,3,8,5,8,2
  5,2,1,9,7,4,3,3
  7,6,7,3,9,2,7,1
  2,3,9,5,7,2,6,5
  5,5,2,2,8,1,4,3;
enddata

min = @sum( Cooperation(i,j) : c(i,j)*x(i,j) );
@for( factory(i): @sum(plant(j):x(i,j)) <= a(i) );
@for( plant(j): @sum(factory(i):x(i,j)) = d(j) );

end
```

$$\begin{aligned} \min \quad & z = \sum_{i=1}^6 \sum_{j=1}^8 c_{ij} \cdot x_{ij} \\ \text{s.t.} \quad & \begin{cases} \sum_{j=1}^8 x_{ij} \leq a_i, i=1,2,\dots,6 \\ \sum_{i=1}^6 x_{ij} = d_j, j=1,2,\dots,8 \\ x_{ij} \geq 0, i=1,2,\dots,6, j=1,2,\dots,8 \end{cases} \end{aligned}$$