第三章 IBM ILOG CPLEX在运输问题中的应用

（一）运输问题的概述

标准的运输问题：就是解决把某种产品从若干个产地调运到若干个销地，在每个产地的供应量与每个销地的需求量已知，并且供应总量与需求总量相等，且知道各地之间的运输单价的前提下，如何确定一个使得总的运输费用最小的方案的问题。

第三章 IBM ILOG CPLEX在运输问题中的应用

（二）数学模型

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

$$\text{s.t.}\quad \text{供应：}\quad \sum_{j=1}^{n} a_i \qquad (i = 1,2,\ldots,m)$$

$$\text{需求：}\quad \sum_{i=1}^{m} x_{ij} = b_j \qquad (j = 1,2,\ldots,n)$$

式中：

X_{ij} ——从产地A_i运输到销地B_j的物资数量；

C_{ij} ——从产地A_i运输到销地B_j的单位运费；

a_i ——各产地的产量是（i=1,2，…，m）；

b_j ——各销地的销量是（j=1,2，…，n）。

第三章 IBM ILOG CPLEX在运输问题中的应用

例1. 某公司从两个产地A1，A2将物品运往三个销地B1，B2，B3，各生产地的产量、各销售地的销量和各生产地运往各销售地的每件物品的运费如下表所示：

	B1	B2	B3	产量
A1	6	4	6	200
A2	6	5	5	300
销量	150	150	200	

问应如何调运，使得总运输费最小？

第三章 IBM ILOG CPLEX在运输问题中的应用

其数学模型可以是：

目标函数：

$$\text{Min } f=6x_{11}+4x_{12}+6x_{13}+6x_{21}+5x_{22}+5x_{23}$$

约束条件：

$x_{11} + x_{12} + x_{13} = 200$，

$x_{21} + x_{22} + x_{23} = 300$，

$x_{11} + x_{21} = 150$，

$x_{12} + x_{22} = 150$，

$x_{13} + x_{23} = 200$，

$x_{ij} \geq 0.\ (i = 1，2；j = 1，2，3)$

（三）模型的OPL语言

仅模型文件的编码：

```
{string} SCities ={"A1", "A2"};
{string} DCities ={"B1", " B2", " B3"};

float Supply[SCities] = [200, 300];
float Demand[DCities] = [150, 150, 200];
assert
    sum(o in SCities) Supply[o] == sum(d in DCities) Demand[d];
float Cost[SCities][DCities] = [ [6, 4, 6], [6, 5, 5] ];

dvar float+ Trans[SCities][DCities];
```

仅模型文件的编码（续）：

```
minimize
  sum( o in SCities ,d in DCities )
    Cost[o][d] * Trans[o][d];

subject to {
  forall( o in SCities )
    ctSupply:
      sum( d in DCities )
        Trans[o][d] == Supply[o];
  forall(d in DCities )
    ctDemand:
      sum( o in SCities )
        Trans[o][d] == Demand[d];
}
```

将数据与模型分开时

模型文件编码：

```
{string} SCities =...;
{string} DCities =...;

float Supply[SCities] = ...;
float Demand[DCities] = ...;
assert
   sum(o in SCities) Supply[o] == sum(d in DCities) Demand[d];
float Cost[SCities][DCities] = ...;

dvar float+ Trans[SCities][DCities];
```

模型文件编码（续）：

```
minimize
  sum( o in SCities ,d in DCities )
    Cost[o][d] * Trans[o][d];

subject to {
  forall( o in SCities )
    ctSupply:
      sum( d in DCities )
        Trans[o][d] == Supply[o];
  forall(d in DCities )
    ctDemand:
      sum( o in SCities )
        Trans[o][d] == Demand[d];
}
```

数据文件编码：

```
SCities = { A1 A2 };
DCities = { B1 B2 B3 };
Supply =#[A1: 200  A2: 300]#;
Demand =#[B1: 150 B2: 150 B3: 200]#;
Cost =  #[A1:  #[B1: 6  B2: 4 B3: 6]#
          A2:  #[B1: 6 B2: 5 B3: 5]#]#;
```

（四）求解结果为：

```
// solution (optimal) with objective 2500
// Quality There are no bound infeasibilities.
// There are no reduced-cost infeasibilities.
// Maximum Ax-b  residual              = 0
// Maximum c-B'pi residual             = 0
// Maximum |x|                         = 200
// Maximum |pi|                        = 6
// Maximum |red-cost|                  = 1
// Condition number of unscaled basis = 9.0e+000
Trans = [[50 150 0]
         [100 0 200]];
```

转化为表格如右表：

	B1	B2	B3
A1	50	150	
A2	100		200