

问题：

某设备制造商有 2 个生产厂 (A1, A2) 和 3 个分销中心 (T1, T2, T3)，生产厂生产出的产品必须运到分销中心，由分销中心销售，生产厂 A1 每个月的产量是 400 台，A2 每个月的产量是 600 台。某月接到 4 个地区 (B1、B2、B3、B4) 的订单，分别是 200 台、150 台、350 台和 300 台。各地之间的单位运价如表所示，运输费用由制造商承担，问制造商应该如何调运才能使总的运输费用最低。

单位运价		产地		销售地			
		A1	A2	B1	B2	B3	B4
分销中心	T1	13	14	23	57	36	41
	T2	15	23	48	52	49	24
	T3	29	13	46	48	36	59

解：

总运费为：45300

调运方案：

运送数量		产地		销售地			
		A1	A2	B1	B2	B3	B4
分销中心	T1	100	100	200	0	0	0
	T2	300	0	0	0	0	300
	T3	0	500	0	150	350	0

解题思路：

分析问题，将运输成本分为两部分：

1. 从生产厂 (A1 和 A2) 到分销中心 (T1, T2, T3) 的费用
2. 从分销中心 (T1, T2, T3) 到销售地 (B1, B2, B3, B4) 的费用

显然可以定义如下决策变量：

x_{ij} 表示从生产厂 A_i 运送到分销中心 T_j 的数量, 其中 $i \in \{1,2\}, j \in \{1,2,3\}$

c_{ij} 表示从生产厂 A_i 运送到分销中心 T_j 的单位运价, 其中 $i \in \{1,2\}, j \in \{1,2,3\}$

y_{jk} 表示从生产厂 T_j 运送到销售地 B_k 的数量, 其中 $j \in \{1,2,3\}, k \in \{1,2,3,4\}$

d_{jk} 表示从生产厂 T_j 运送到销售地 B_k 的单位运价, 其中 $j \in \{1,2,3\}, k \in \{1,2,3,4\}$

总运输成本可以表示为：

$$Z = \sum_{i=1}^2 \sum_{j=1}^3 x_{ij} \cdot c_{ij} + \sum_{j=1}^3 \sum_{k=1}^4 y_{jk} \cdot d_{jk}$$

需要考虑的约束条件为：

1. 各生产厂生产量不超过产能上限
2. 各分销中心运送到最终销售地的数量和为订单标记的数量

3. 各分销中心接收的产品数量等于运出的产品数量
4. 所有变量均非负

可以得出如下优化函数：

$$\min Z = \sum_{i=1}^2 \sum_{j=1}^3 x_{ij} \cdot c_{ij} + \sum_{j=1}^3 \sum_{k=1}^4 y_{jk} \cdot d_{jk}$$

Subject to

$$\begin{aligned} x_{11} + x_{12} + x_{13} &\leq 400 \\ x_{21} + x_{22} + x_{23} &\leq 600 \\ y_{11} + y_{21} + y_{31} &= 200 \\ y_{12} + y_{22} + y_{32} &= 150 \\ y_{13} + y_{23} + y_{33} &= 350 \\ y_{14} + y_{24} + y_{34} &= 300 \\ x_{11} + x_{21} &= y_{11} + y_{12} + y_{13} + y_{14} \\ x_{12} + x_{22} &= y_{21} + y_{22} + y_{23} + y_{24} \\ x_{13} + x_{23} &= y_{31} + y_{32} + y_{33} + y_{34} \\ x_{11}, x_{21}, y_{11}, y_{12}, y_{13}, y_{14} &\geq 0 \end{aligned}$$

运行代码后可得如下结果：

运送数量		产地		销售地			
		A1	A2	B1	B2	B3	B4
分销中心	T1	100	100	200	0	0	0
	T2	300	0	0	0	0	300
	T3	0	500	0	150	350	0

总运费为：45300

MATLAB Code

```
% Define cost matrices
c = [13 15 29; 14 23 13]; % c_ij matrix
d = [23 57 36 41; 48 52 49 24; 46 48 36 59]; % d_jk matrix
f = [c(1,1); c(1,2); c(1,3); c(2,1); c(2,2); c(2,3); d(1,1);
d(1,2); d(1,3); d(1,4);
d(2,1); d(2,2); d(2,3); d(2,4); d(3,1); d(3,2); d(3,3);
d(3,4)];
A = [
1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0; % x11 + x12 + x13 <= 400
0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 % x21 + x22 + x23 <= 600
];
b = [400; 600];
Aeq1 = [
0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0; % y11 + y21 + y31 = 200
```

```

    0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0; % y12 + y22 + y32 = 150
    0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0; % y13 + y23 + y33 = 350
    0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1; % y14 + y24 + y34 = 300
];
beq1 = [200; 150; 350; 300];
% Define equality constraints - Part 2: Relationship between x
and y
Aeq2 = [
    1 0 0 1 0 0 -1 -1 -1 -1 0 0 0 0 0 0 0 0; % x11 + x21 = y11 +
y12 + y13 + y14
    0 1 0 0 1 0 0 0 0 0 -1 -1 -1 -1 0 0 0 0; % x12 + x22 = y21 +
y22 + y23 + y24
    0 0 1 0 0 1 0 0 0 0 0 0 0 0 -1 -1 -1 -1 % x13 + x23 = y31 +
y32 + y33 + y34
];
beq2 = [0; 0; 0];
Aeq = [Aeq1; Aeq2];
beq = [beq1; beq2];
lb = zeros(18, 1);
options = optimoptions('linprog', 'Display', 'iter');
[x, fval] = linprog(f, A, b, Aeq, beq, lb, [], options);
disp('Optimal values of decision variables:')
disp(x)
disp('Minimum value of the objective function:')
disp(fval)

```

Python code

```

import pulp

# 定义问题
model = pulp.LpProblem("Transportation_Cost_Minimization",
pulp.LpMinimize)

# 定义决策变量，确保变量为非负数
x = pulp.LpVariable.dicts("x", [(i, j) for i in range(1, 3) for j in
range(1, 4)], lowBound=0, cat='Continuous')
y = pulp.LpVariable.dicts("y", [(j, k) for j in range(1, 4) for k in
range(1, 5)], lowBound=0, cat='Continuous')

# 定义成本系数
c = {(1, 1): 13, (1, 2): 15, (1, 3): 29,
      (2, 1): 14, (2, 2): 23, (2, 3): 13}
d = {(1, 1): 23, (1, 2): 57, (1, 3): 36, (1, 4): 41,
      (2, 1): 48, (2, 2): 52, (2, 3): 49, (2, 4): 24,
      (3, 1): 46, (3, 2): 48, (3, 3): 36, (3, 4): 59}

```

```
# 目标函数
model += pulp.lpSum(c[i, j] * x[i, j] for i in range(1, 3) for j in
range(1, 4)) + \
    pulp.lpSum(d[j, k] * y[j, k] for j in range(1, 4) for k in
range(1, 5)), "Total_Transportation_Cost"

# 生产能力约束
model += x[1, 1] + x[1, 2] + x[1, 3] <= 400, "A1_Production_Capacity"
model += x[2, 1] + x[2, 2] + x[2, 3] <= 600, "A2_Production_Capacity"

# 地区需求约束
model += y[1, 1] + y[2, 1] + y[3, 1] == 200, "B1_Demand"
model += y[1, 2] + y[2, 2] + y[3, 2] == 150, "B2_Demand"
model += y[1, 3] + y[2, 3] + y[3, 3] == 350, "B3_Demand"
model += y[1, 4] + y[2, 4] + y[3, 4] == 300, "B4_Demand"

# 分销中心流量平衡约束
model += y[1, 1] + y[1, 2] + y[1, 3] + y[1, 4] == x[1, 1] + x[2, 1],
"T1_Flow_Balance"
model += y[2, 1] + y[2, 2] + y[2, 3] + y[2, 4] == x[1, 2] + x[2, 2],
"T2_Flow_Balance"
model += y[3, 1] + y[3, 2] + y[3, 3] + y[3, 4] == x[1, 3] + x[2, 3],
"T3_Flow_Balance"

# 求解
model.solve()

# 输出结果
print("Status:", pulp.LpStatus[model.status])
for v in model.variables():
    print(v.name, "=", v.varValue)

print("Total Cost =", pulp.value(model.objective))
```