第3章

CHAPTER 3

物流网络模型

物流(logistics)是供应链中的一部分,它被定义为一种对产地到销地间物品的有效流通及其存储、服务等相关信息进行计划、实施及管理(尤其是供应、配送及存储)的全过程给予优化的综合活动。物流原本是以发挥其本身最大限度的机能为目标的,这意味着它要以供给者与需求者之间的原材料、产品、商品的调配和供给为核心,从产品或服务的自策划、开发、设计、制造到使用、撤销、废弃、设备维护的整个产品生命周期为对象,优化并提高其效率的节约型的企业间贸易和物流结构。

3.1 物流模型

物流的有效管理是从战略性的网络规划开始的。也就是说,分析企业的地点和顾客的地点并制定配送产品到最终目的地的最有效的方法。物流的最优化,是通过供应链中配送费用的降低、物流过程中内部效率的提高、顾客服务的最大化以及成本的降低等来实现的。在此,若考虑将多个产地生产的产品配送给多个销地(比如配送中心或仓库等)的时候,在满足各自的供应量和需求量的条件下,要实现成本最低,则有必要制定关于从哪个产地到哪个销地应该配送多少产品的配送计划,称这种计划为配送计划(transportation planning,TP)模型。

配送计划模型是物流系统,即物流网络问题中的基本模型。因其模型结构

的特殊性，迄今为止已有很多研究者对它的解法进行研究并取得了一定的成果，并向多目标多阶段等扩展的配送计划模型发展。

配送计划模型根据目标函数的特性划分如下。

- 线性问题或非线性问题，
- 单目标问题或多目标问题。

根据物流系统的约束条件划分如下。

- 平面(planar)或一般配送计划(solid TP)模型，
- 平衡(balanced)或不平衡配送计划(unbalanced TP)模型。

例 3.1　简单的配送计划举例

通常的配送计划模型是以最低成本将某种商品从供给方配送到需求地的优化问题。如图 3.1 所示，从产地(工厂)$i=1,2,3$ 向销地(家庭、配送中心或者仓库)$j=1,2,3,4$ 配送产品时，在满足各产地 i 的供应量 a_i 和各销地 j 的需求量 b_j 这一约束条件下，考虑制定配送总成本最小的配送计划。这需要根据相应的目标函数和约束条件建立线性和非线性的整数规划数学模型。

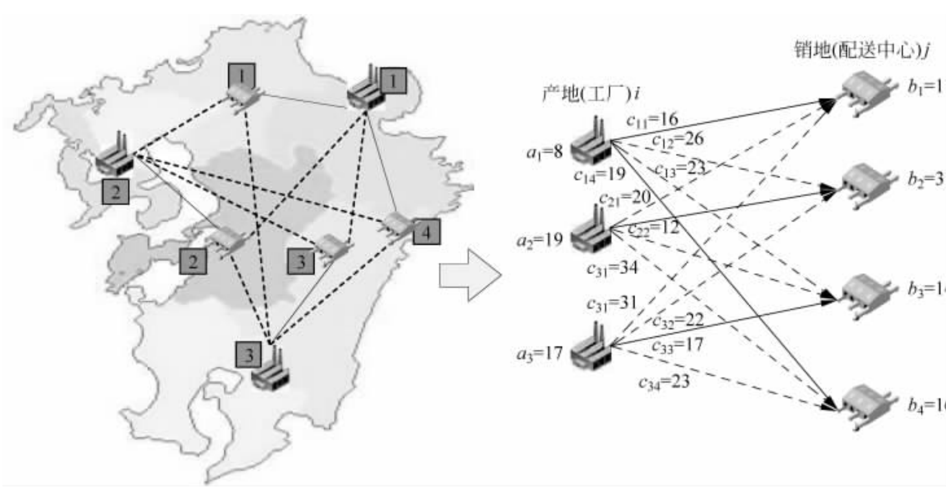

图 3.1　从 3 个产地往 4 个销地的产品配送计划

3.1.1　配送计划模型

a. 基本配送计划模型

如例 3.1 所示的配送计划模型，是从产地(工厂)$i=1,2,\cdots,I$ 往销地(家庭、配送中心或仓库)$j=1,2,\cdots,J$ 配送产品时，在满足产地的供应量和销地的需求量这一约束条件下，制定配送总成本最低的配送计划。为了建立该配送计

划网络的数学模型，定义所需记号如下。

编号：

i：产地（$i=1,2,\cdots,I$）

j：销地（$j=1,2,\cdots,J$）

参数：

I：产地数量

J：销地数量

a_i：产地 i 的供应量（$a_i \geqslant 0$）

b_j：销地 j 的需求量（$b_j \geqslant 0$）

c_{ij}：从产地 i 向销地 j 配送的单位产品配送成本

决策变量：

x_{ij}：从产地 i 到销地 j 的配送量

因此，配送计划网络的基本数学模型如下：

$$\min z = \sum_{i=1}^{I} \sum_{j=1}^{J} c_{ij} x_{ij} \tag{3.1}$$

$$\text{s.t.} \quad \sum_{j=1}^{J} x_{ij} \leqslant a_i, \quad i = 1,2,\cdots,I \quad \text{（供应量的约束条件）} \tag{3.2}$$

$$\sum_{i=1}^{I} x_{ij} \geqslant b_j, \quad j = 1,2,\cdots,J \quad \text{（需求量的约束条件）} \tag{3.3}$$

$$x_{ij} \geqslant 0, \quad \forall\, i,j \tag{3.4}$$

此外，该基本数学模型以公式（3.5）所示的总供应量和总需求量相等的这一平衡条件为前提。

$$\sum_{i=1}^{I} a_i = \sum_{j=1}^{J} b_j \tag{3.5}$$

假如平衡公式不成立，例如供应量过剩，则建立虚拟的销地，将过剩部分设定为虚拟销地的需求量，并将往虚拟销地的配送成本设定为 0，这样平衡条件（balanced condition）就能够成立。在所得结果中，各产地发往虚拟销地的配送量，则表示各产地未能供应出去的部分。对于需求量过剩的情况，也可以用同样的方法建立虚拟产地。

b. 基于容量约束的配送计划模型

基于容量约束的配送计划（capacitated transportation planning，cTP）模型，是指在各配送线路（i,j）上有配送量的容量约束 u_{ij} 的扩充模型，形式如下。

$$\min z = \sum_{i=1}^{I} \sum_{j=1}^{J} c_{ij} x_{ij} \tag{3.6}$$

$$\text{s.t.}\quad \sum_{j=1}^{J} x_{ij} \leqslant a_i,\quad i = 1,2,\cdots,I \quad (\text{供应量的约束}) \tag{3.7}$$

$$\sum_{i=1}^{I} x_{ij} \geqslant b_j,\quad j = 1,2,\cdots,J \quad (\text{需求量的约束}) \tag{3.8}$$

$$0 \leqslant x_{ij} \leqslant u_{ij},\quad \forall\, i,j \quad (\text{配送量的容量约束}) \tag{3.9}$$

c. 基于固定费用的配送计划模型

基于固定费用的配送计划(fixed-charge transportation planning,fcTP)模型,可以由基本配送计划模型扩充而来(图3.2)。在物流系统中,如同带有固定费用的最小费用流模型一样,实际配送计划模型大多以带固定费用的配送计划来建立数学模型。例如,在配送计划模型中,有时固定费用被包含在给定的仓库间的各配送成本里,或者被包含在工厂或仓库的设备费用里。由此,原配送计划模型可以看做是整个线路的固定费用为0的fcTP模型。只是在fcTP模型中,由于目标函数中存在固定费用,目标函数是不连续的,所以求可行解将更加困难。

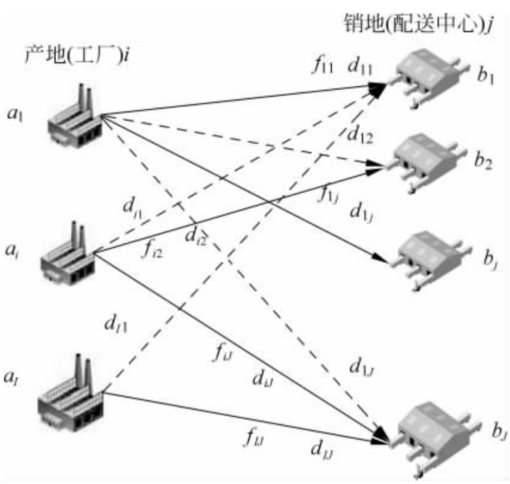

图3.2　从产地往销地的固定费用配送模型

在带有固定费用的配送计划模型中,选择最佳实施计划时,同时考虑以下两类成本:(1)产地到销地间的运输费用,(2)固定费用。另外,fcTP模型也可制定从多个工厂往多个仓库配送的最低成本配送计划。

基于固定费用的配送计划网络数学模型如下:

$$\min f(\boldsymbol{x}) = \sum_{i=1}^{I}\sum_{j=1}^{J}[f_{ij}(\boldsymbol{x}) + d_{ij}g_{ij}(\boldsymbol{x})] \tag{3.10}$$

$$\text{s.t.} \quad \sum_{j=1}^{J} x_{ij} \leqslant a_i, \quad i = 1,2,\cdots,I \tag{3.11}$$

$$\sum_{i=1}^{I} x_{ij} \geqslant b_j, \quad j = 1,2,\cdots,J \tag{3.12}$$

$$x_{ij} \geqslant 0, \quad \forall\, i,j \tag{3.13}$$

这里，$f_{ij}(x)$是从产地 i 往销地 j 的一般运输成本，d_{ij}是从产地 i 往销地 j 的固定费用。从产地 i 往销地 j 配送的决策变量定义如下。

$$g_{ij}(\boldsymbol{x}) = \begin{cases} 1, & x_{ij} > 0 \\ 0, & \text{其他} \end{cases} \tag{3.14}$$

d. 基于排斥约束的配送计划模型Ⅰ

物流系统在实际问题的应用中，经常会以附加约束条件来扩充配送计划模型。本节将介绍基于排斥约束的配送模型Ⅰ（exclusionary side constrained transportation planning，escTP）。

该 escTP 模型是在配送计划模型上添加了不允许由多个产地往某个特定销地同时发货的这一附加约束条件。在实际物流系统中，由于企业内的产品的种类系列不同，经常遇到此类问题。例如，有时在同一条线路上，化学药品不可以与食品等其他产品在同一个集装箱或者车辆中一起配送。在这种状况下，escTP 模型的目标是在考虑在产地不能同时出货的前提下，制定满足供应、需求约束条件的可行配送计划。该问题的难度随着附加约束条件而增加，而在实际问题中的应用会更加复杂。另外，由于该约束条件为非线性函数，所以不能使用以往的 LP 软件包来求解。

从 I 个产地往 J 个销地配送不同类产品的配送计划 escTP 模型的数学模型如下：

$$\min z = \sum_{i=1}^{I} \sum_{j=1}^{J} c_{ij} x_{ij} \tag{3.15}$$

$$\text{s.t.} \quad \sum_{j=1}^{J} x_{ij} \leqslant a_i, \quad i = 1,2,\cdots,I \tag{3.16}$$

$$\sum_{i=1}^{I} x_{ij} \geqslant b_j, \quad j = 1,2,\cdots,J \tag{3.17}$$

$$x_{ij} x_{kj} = 0 \quad \forall\,(i,k) \in D_j, \quad j = 1,2,\cdots,J \tag{3.18}$$

$$x_{ij} \geqslant 0, \quad \forall\, i,j \tag{3.19}$$

这里，$D_j=\{(i,k)\mid$产地 i 和产地 k 不能同时向销地 j 配送$\}$，式(3.18)即为产地 i 和产地 k 不允许同时向销地 j 配送的约束条件。

e. 基于排斥约束的配送模型Ⅱ

基于排斥约束的配送模型Ⅱ(nonlinear side-constrained transportation planning,nscTP)与前面提到的 escTP 模型类似。escTP 模型是由多个产地往一个销地配送时出现的约束条件,而 nscTP 则是由一个产地往多个销地配送时出现的约束条件。也就是说,考虑对于某个产地来说有可以配送的销地和不可以配送的销地的情况。

$$\min z = \sum_{i=1}^{I}\sum_{j=1}^{J} c_{ij} x_{ij} \tag{3.20}$$

$$\text{s.t.} \quad \sum_{j=1}^{J} x_{ij} \leqslant a_i, \quad i = 1,2,\cdots,I \tag{3.21}$$

$$\sum_{i=1}^{I} x_{ij} \geqslant b_j, \quad j = 1,2,\cdots,J \tag{3.22}$$

$$x_{ij} x_{il} = 0 \quad (j,l) \in S_i, \quad i = 1,2,\cdots,I \tag{3.23}$$

$$x_{ij} \geqslant 0, \quad \forall\, i,j \tag{3.24}$$

这里,$S_i=\{(j,l)\,|$不能由产地 i 向销地 j 和销地 l 同时配送$\}$,公式(3.23)表示不可以从产地 i 向两个不同的销地 j 和 l 同时配送。

3.1.2　基于矩阵的遗传算法解法

Michalewicz 等人[1,2]针对线性和非线性的配送计划模型,提出了基于矩阵的遗传算法。

a. 基于矩阵的染色体设计

编码

在这里,以矩阵来构造满足系统约束条件的某一配送计划的染色体,表示如下:

$$\boldsymbol{X}_p = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1J} \\ x_{21} & x_{22} & \cdots & x_{2J} \\ \vdots & \vdots & & \vdots \\ x_{I1} & x_{I2} & \cdots & x_{IJ} \end{bmatrix} \tag{3.25}$$

这里,矩阵 $\boldsymbol{X}_p$ 是第 p 个染色体,各元素 x_{ij} 是染色体的遗传基因,表示问题的决策变量(配送量)。采用基于矩阵的遗传算法,生成配送计划模型配送计划初始解的编码流程如下。

基于矩阵的染色体编码流程

```
procedure: matrix-based encoding
input: number of plants I, number of customers J,
       capacity of i th plant [a_i], demand of j th customer [b_j]
output: chromosome X_P=[x_ij]
begin
   S← {1 , 2 , ... , IJ};
   repeat
      select a random number k from set S;
      i ← ⌊(n-j) / J+1⌋;        // calculate corresponding row i;
      j ← k mod J;              // calculate corresponding column j;
      x_ij ← min{a_i, b_j};     // assign available amount of units of x_ij ;
      a_i ← a_i − x_ij ;        // update the following data
      b_j ← b_j − x_ij ;
      S ← S / {k};
   until(S =∅)
   output [x_ij];
End
```

解码

通常作为配送计划模型的解码方法，在这里采用如下的矩阵码的解码方法。

基于矩阵的染色体解码流程

```
procedure : matrix-based decoding
input: chromosome [x_ij],shipping cost [c_ij]
output: total shipping cost z
begin
   z← 0;
   for i=1 to I
      for j=1 to J
         z=z+ c_ij x_ij;
   output total shipping cost z;
end
```

b. 遗传操作

交叉：首先选择两个父代染色体 $\boldsymbol{X}_1=[x_{ij}^1]$与 $\boldsymbol{X}_2=[x_{ij}^2]$。基于矩阵的交叉(matrix-based crossover)过程由以下 3 个步骤完成。

步骤 1：基于矩阵的交叉

Step 1：由父代染色体 $\boldsymbol{X}_1$ 与 $\boldsymbol{X}_2$ 临时组成如下所示的两个矩阵 $\boldsymbol{D}=(d_{ij})$与 $\boldsymbol{R}=(r_{ij})$。

$$d_{ij} = \lfloor (x_{ij}^1 + x_{ij}^2)/2 \rfloor \tag{3.26}$$

$$r_{ij} = (x_{ij}^1 + x_{ij}^2) \bmod 2 \tag{3.27}$$

这两个矩阵具有如下的从属关系。

$$a_i - \sum_{j=1}^{J} d_{ij} = \frac{1}{2}\sum_{j=1}^{J} r_{ij} \quad i = 1,2,\cdots,I \tag{3.28}$$

$$b_j - \sum_{i=1}^{I} d_{ij} = \frac{1}{2}\sum_{i=1}^{I} r_{ij} \quad j = 1,2,\cdots,J \tag{3.29}$$

该关系可以证明如下。

$$\begin{aligned}\sum_{j=1}^{J} d_{ij} &= \sum_{j=1}^{J}\lfloor (x_{ij}^1 + x_{ij}^2)/2 \rfloor \\ &= \sum_{j=1}^{J} \frac{1}{2}((x_{ij}^1 + x_{ij}^2) - (x_{ij}^1 + x_{ij}^2)\bmod 2) \\ &= \frac{1}{2}\sum_{j=1}^{J}(x_{ij}^1 + x_{ij}^2) - \frac{1}{2}\sum_{j=1}^{J}((x_{ij}^1 + x_{ij}^2)\bmod 2) \\ &= a_i - \frac{1}{2}\sum_{j=1}^{J} r_{ij}\end{aligned}$$

Step 2：将矩阵 $\boldsymbol{R}$ 分解为满足以下条件的两个矩阵 $\boldsymbol{R}_1=[r_{ij}^1]$与 $\boldsymbol{R}_2=[r_{ij}^2]$。

$$\boldsymbol{R} = \boldsymbol{R}_1 + \boldsymbol{R}_2 \tag{3.30}$$

$$\sum_{j=1}^{J} r_{ij}^1 = \sum_{j=1}^{J} r_{ij}^2 = \frac{1}{2}\sum_{j=1}^{J} r_{ij} \quad i = 1,2,\cdots,I \tag{3.31}$$

$$\sum_{i=1}^{I} r_{ij}^1 = \sum_{i=1}^{I} r_{ij}^2 = \frac{1}{2}\sum_{i=1}^{I} r_{ij} \quad j = 1,2,\cdots,J \tag{3.32}$$

Step 3：根据父代染色体 $\boldsymbol{X}_1$ 与 $\boldsymbol{X}_2$ 组成的临时矩阵 $\boldsymbol{D}$ 和 $\boldsymbol{R}_1$、$\boldsymbol{R}_2$，通过以下公式生成两个染色体 $\boldsymbol{X}_1'$，$\boldsymbol{X}_2'$。

$$\boldsymbol{X}_1' = \boldsymbol{D} + \boldsymbol{R}_1 \tag{3.33}$$

$$\boldsymbol{X}_2' = \boldsymbol{D} + \boldsymbol{R}_2 \tag{3.34}$$

交叉操作实例

考虑有 4 个产地和 5 个销地的配送计划问题，供应量和需求量如下所示。

$$a_1 = 8,\quad a_2 = 4,\quad a_3 = 12,\quad a_4 = 6$$

$$b_1 = 3,\quad b_2 = 5,\quad b_3 = 10,\quad b_4 = 7,\quad b_5 = 5$$

用基于矩阵的编码方法，生成两个该配送计划模型满足约束条件的可行配送计划 $\boldsymbol{X}_1$ 与 $\boldsymbol{X}_2$，分别如表 3.1 与表 3.2 所示。

表 3.1 $\boldsymbol{X}_1$ 对应的配送计划

产地 \ 销地	1	2	3	5	5	产量
1	X_{11} 1	X_{12} 0	X_{13} 0	X_{14} 7	X_{15} 0	8
2	X_{21} 0	X_{22} 4	X_{23} 0	X_{24} 0	X_{25} 0	4
3	X_{31} 2	X_{32} 1	X_{33} 4	X_{34} 0	X_{35} 5	12
4	X_{41} 0	X_{42} 0	X_{43} 6	X_{44} 0	X_{45} 0	6
消费量	3	5	10	7	5	30 \ 30

表 3.2 $\boldsymbol{X}_2$ 对应的配送计划

产地 \ 销地	1	2	3	5	5	产量
1	X_{11} 0	X_{12} 0	X_{13} 5	X_{14} 0	X_{15} 3	8
2	X_{21} 0	X_{22} 4	X_{23} 0	X_{24} 0	X_{25} 0	4
3	X_{31} 0	X_{32} 0	X_{33} 5	X_{34} 7	X_{35} 0	12
4	X_{41} 3	X_{42} 1	X_{43} 0	X_{44} 0	X_{45} 2	6
消费量	3	5	10	7	5	30 \ 30

将这些表分别写成对应的矩阵，则以下的两个矩阵 $\boldsymbol{X}_1$，$\boldsymbol{X}_2$ 成为父代染色体。

$$\text{父代染色体 } \boldsymbol{X}_1 = \begin{bmatrix} 1 & 0 & 0 & 7 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 2 & 1 & 4 & 0 & 5 \\ 0 & 0 & 6 & 0 & 0 \end{bmatrix}, \quad \text{父代染色体 } \boldsymbol{X}_2 = \begin{bmatrix} 0 & 0 & 5 & 0 & 3 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 5 & 7 & 0 \\ 3 & 1 & 0 & 0 & 2 \end{bmatrix}$$

计算这两个矩阵的和。

$$\boldsymbol{X}_1+\boldsymbol{X}_2=\begin{bmatrix}1&0&5&7&3\\0&8&0&0&0\\2&1&9&7&5\\3&1&6&0&2\end{bmatrix}$$

则可求得临时矩阵 $\boldsymbol{D}$ 与 $\boldsymbol{R}$，如下所示：

$$\boldsymbol{D}=\begin{bmatrix}0&0&2&3&1\\0&4&0&0&0\\1&0&4&3&2\\1&0&3&0&1\end{bmatrix},\quad \boldsymbol{R}=\begin{bmatrix}1&0&1&1&1\\0&0&0&0&0\\0&1&1&1&1\\1&1&0&0&0\end{bmatrix}$$

将矩阵 $\boldsymbol{R}$ 分解为矩阵 $\boldsymbol{R}_1$ 与 $\boldsymbol{R}_2$：

$$\boldsymbol{R}_1=\begin{bmatrix}0&0&1&0&1\\0&0&0&0&0\\0&1&0&1&0\\1&0&0&0&0\end{bmatrix},\quad \boldsymbol{R}_2=\begin{bmatrix}1&0&0&1&0\\0&0&0&0&0\\0&0&1&0&1\\0&1&0&0&0\end{bmatrix}$$

通过以上的操作，生成以下两个子染色体 $\boldsymbol{X}_1'$，$\boldsymbol{X}_2'$：

$$\boldsymbol{X}_1'=\boldsymbol{D}+\boldsymbol{R}_1=\begin{bmatrix}0&0&2&3&1\\0&4&0&0&0\\1&0&4&3&2\\1&0&3&0&1\end{bmatrix}+\begin{bmatrix}0&0&1&0&1\\0&0&0&0&0\\0&1&0&1&0\\1&0&0&0&0\end{bmatrix}=\begin{bmatrix}0&0&3&3&2\\0&4&0&0&0\\1&1&4&4&2\\2&0&3&0&1\end{bmatrix}$$

$$\boldsymbol{X}_2'=\boldsymbol{D}+\boldsymbol{R}_2=\begin{bmatrix}0&0&2&3&1\\0&4&0&0&0\\1&0&4&3&2\\1&0&3&0&1\end{bmatrix}+\begin{bmatrix}1&0&0&1&0\\0&0&0&0&0\\0&0&1&0&1\\0&1&0&0&0\end{bmatrix}=\begin{bmatrix}1&0&2&4&1\\0&4&0&0&0\\1&0&5&3&3\\1&1&3&0&1\end{bmatrix}$$

变异：该操作由以下3个步骤构成。

步骤2：基于矩阵的变异

Step 1：根据父染色体矩阵构造子矩阵。随机选取行 $\{i_1,i_2,\cdots,i_p\}$ 和列 $\{j_1,j_2,\cdots,j_q\}$，生成 $p\times q$ 型的子矩阵 $\boldsymbol{Y}=[y_{kl}]$。其中，由于 $\{i_1,i_2,\cdots,i_p\}$ 在 $\{1,2,\cdots,I\}$ 的范围内，$\{j_1,j_2,\cdots,j_q\}$ 在 $\{1,2,\cdots,J\}$ 的范围内，所以满足 $p\in[1,2,\cdots,I]$，$q\in[1,2,\cdots,J]$。y_{kl} 的值为父染色体矩阵元素 (i_k,j_l) 的值。

Step 2：重新分配子矩阵的元素。被重分配的元素总和 a_i^y 和 b_i^y 表示如下。

$$a_i^y=\sum_{j\in\{j_1,j_2,\cdots,j_q\}}y_{ij},\quad i=i_1,i_2,\cdots,i_p$$

$$b_i^y=\sum_{i\in\{i_1,i_2,\cdots,i_q\}}i_{ij},\quad j=j_1,j_2,\cdots,j_p$$