运输问题

(Transportation Problem)

东北大学 系统工程研究所

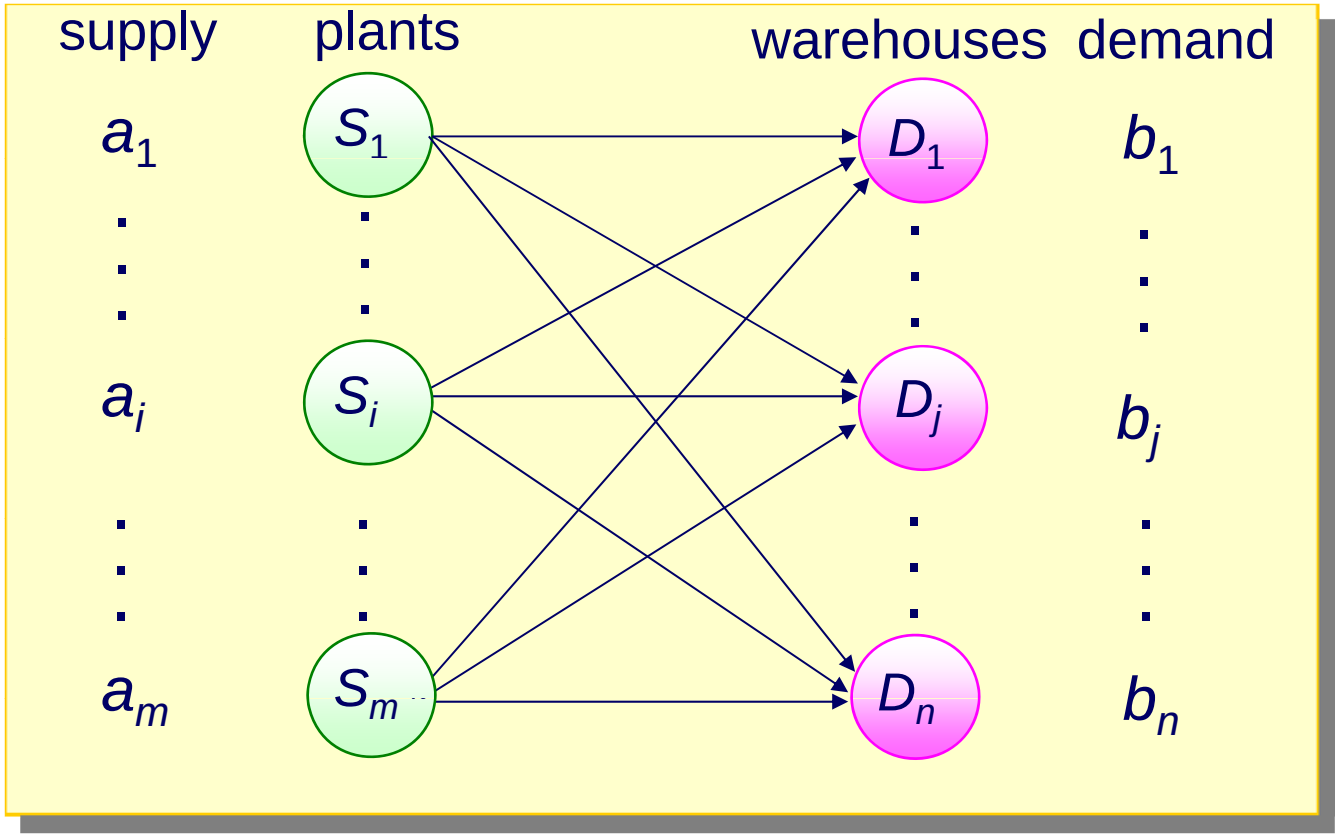
2011.09

基本运输问题

- 运输问题最早是由**Hitchcock**在**1941**年提出。
- 此后，人们对这一问题的研究给予了极大的关注并对各类运输问题进行了研究。
- 基本的运输问题是“线性单目标”运输问题。
  - 从不同的供给起点或来源向不同的终点运送同一种物品，每个终点需要特定数量的物品。
  - 问题是如何分配在每个起点的供给，以便在满足每个终点需求的条件下，优化某个目标。
  - 常用的目标函数有：运输费用最小、加权距离最小或利润最大 等。
- 例如：
  - 从工厂向仓库运送产品
  - 运输问题的目标就是要找到成本最小的运输模式

运输问题的网络模型

- 运输问题可以采用如下网络模型进行描述。
  - 这样的图由起始节点和终止节点及连接他们的边构成。
  - 图中一个集合中的所有边直接与另一个集合相连，构成完全二分图(**complete bipartite graph**)。



模型描述

- 给定 m 个起点和 n 个终点, 运输问题可描述为:

$$\min \quad z(x)=\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}x_{ij}$$

$$\text{s.t.} \quad \sum_{j=1}^{n}x_{ij}\le a_i,\ \ i=1,2,...,m$$

$$\sum_{i=1}^{m}x_{ij}\ge b_j,\ \ j=1,2,...,n$$

$$x_{ij}\ge 0,\qquad \forall i,j$$

where,

x_{ij} : 从起点 i 到终点 j 的运输量;
c_{ij} : 从起点 i 到终点 j 的单位运输费用;
a_i : 起点 i 的供给量;
b_j : 终点 j 的需求量。

分类

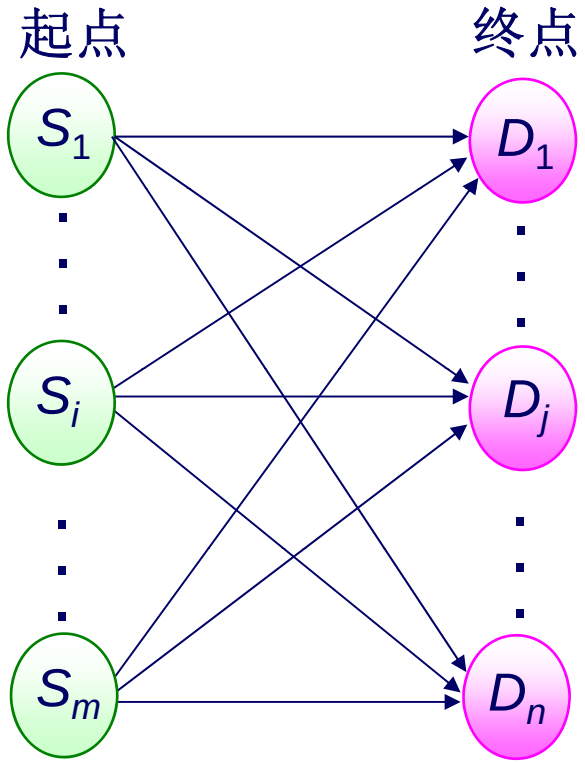
- 按目标的类型划分
  - 线性问题或非线性问题;
  - 单目标问题或多目标问题。
- 按约束的类型划分
  - 二维(planar)问题或三维(solid)问题;
  - 平衡问题或非平衡问题。
- 各类运输问题
  - 线性运输问题
  - 双准则线性、双准则三维运输问题
  - 固定费用运输问题
  - 容量限制的运输问题
  - 广义运输问题
  - 多目标运输问题
  - 模糊多准则三维运输问题
  - 带模糊系数的双目标运输问题
  - 两阶段运输问题

特点

- 是一个基本的网络问题
- 问题的约束具有特殊的结构
- 线性或非线性问题
- 单目标或多目标问题

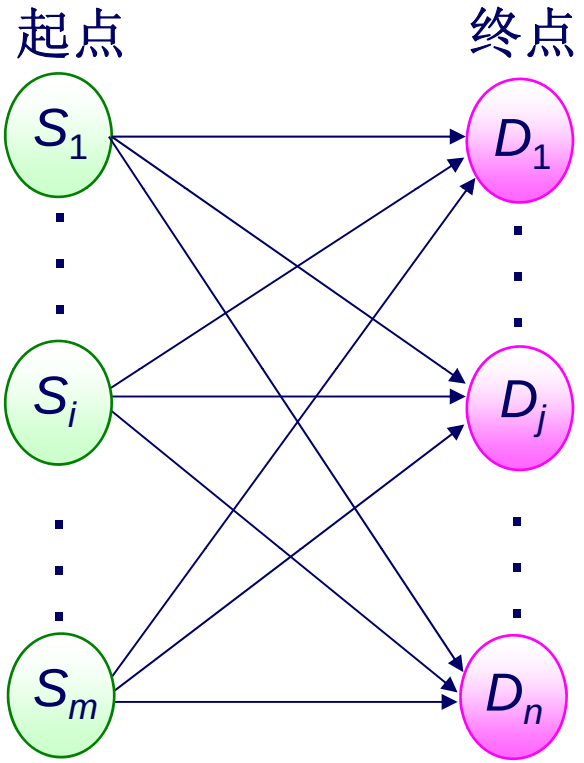
应用

- 工厂到仓库的产品运输问题
- 仓库/零售商供应链问题
  - 物品从仓库运出,到达客户,以满足客户的需求。如何分配运输量能使运输成本最低。
- 盐业配送问题
  - 要将食用盐从食盐供应站，配送到下属工厂或者用盐单位，如何配送能使运输费用最少。
- 应急物资的紧急调运问题
  - 当发生自然灾害或一些突发事件时（例如：地震、洪水等），经常会出现需要从多个供应点向多个需求点运输救援物资的问题。这时的目标通常考虑的是运输时间最短。
  - 在军事上的应用就是军械物资调运问题。



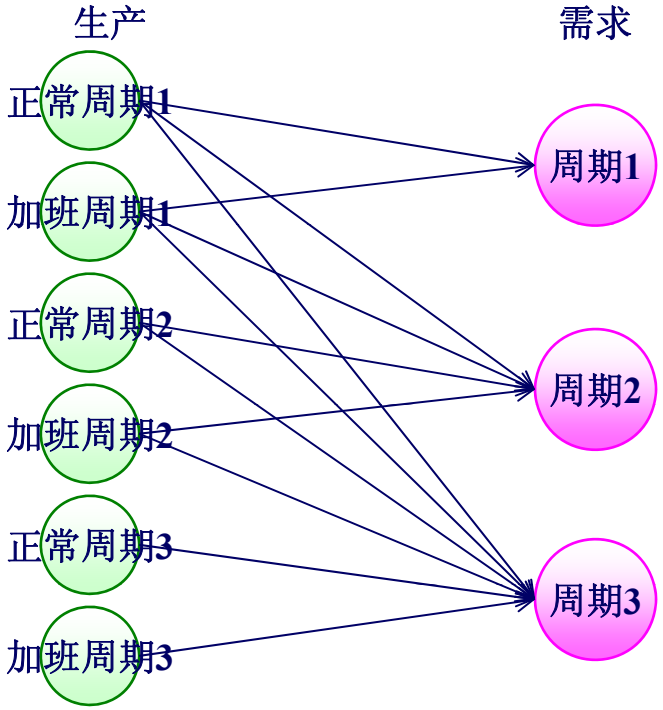
应用

- 多热源供热管网优化问题
  - 例如一个企业有多个热源（热水或蒸汽热），同时还有一些需要热能的车间或工厂，问题是如何调配热源，既能满足要求，又能使运费最小。
- 土地平整中的土方调配问题
  - 从不同的挖方区向不同的填方区运送土方,每个填方区需要特定数量的土方。问题是如何分配每个挖方区的土方供给,以便在满足每个填方区需求的条件下,使得全部费用最小。
  - 公路工程、房屋建设等，都涉及这个问题。
- 原油进口航线优化问题
  - 我国的原油进口**90%**以上采用海上航运。有不同的原油进口地，和接纳港口，如何优化运输航线，才能使总的运输费用最少。
- 军事上的兵力展开问题
  - 由数个具有一定数量兵力的集结地或基地将兵力以“最短时间”展开到多个阵地，每个阵地应到达一定数量的兵力。
- ……



多阶段的生产计划和库存控制问题

- 考虑季节性销售模式的生产商，假设生产保持完全的稳定，或生产的变化能满足销售模式 。
- 考虑3个周期，各周期的预计需求分别为：45，64，75个单位。正常时间单位成本是4，加班时间单位成本是6，每个周期单位存储成本是1.5。
- 假设每个周期的成本相同。而且在每个周期，正常生产能力是50个单位，加班生产能力是20个单位。
- 如何安排生产计划才能使总成本最小。

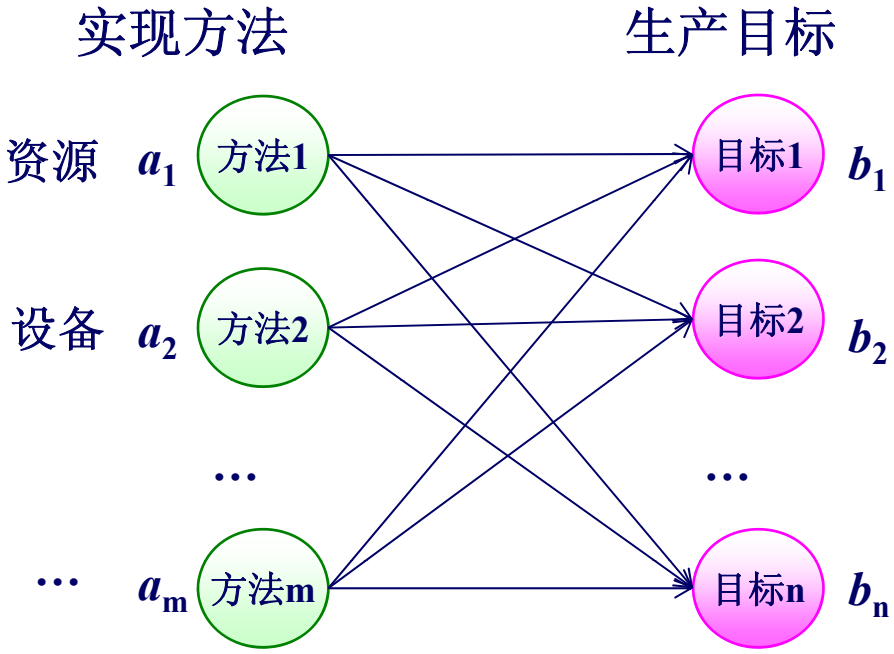


运输表

源点(生产源)	终点(周期需求)			供应(生产能力)
	1	2	3	
1. 正常周期 1	4.0	5.5	7.0	50
2. 加班周期 1	6.0	7.5	9.0	20
3. 正常周期 2	—	4.0	5.5	50
4. 加班周期 2	—	6.0	7.5	20
5. 正常周期 3	—	—	4.0	50
6. 加班周期 3	—	—	6.0	20
需求	45	64	75	

生产计划问题

- 给定 n 个生产目标，m 种用于实现这些目标的方法（固定的资源、设备等）。
- 假设每个目标有若干方法可以实现，则目的是如何分配各种方法以完成各种目标，达到总成本最小。



$$\min \quad z(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ij} \le a_i, \ \ i = 1,2,...,m$$

$$\sum_{i=1}^{m} p_{ij}x_{ij} \ge b_j, \ \ j = 1,2,...,n$$

$$x_{ij} \ge 0, \qquad \forall i, j$$

- 这里的决策变量 x_{ij} 是各方法(例如资源)的使用量(或设备的使用时间)；
- c_{ij} 是使用方法（资源或设备）的单位成本；
- p_{ij} 是方法的数量单位； $p_{ij} * x_{ij}$ →表示完成目标的量。

求解方法

- 一般的求解方法
  - 线性规划的方法
  - 多目标线性规划的方法
- GA求解
  - 人们应用遗传算法，针对各类不同的运输问题进行了多种尝试
  - 实践证明，用遗传算法求解运输问题效果很好

线性运输问题

(Linear Transportation Problem)

线性运输问题

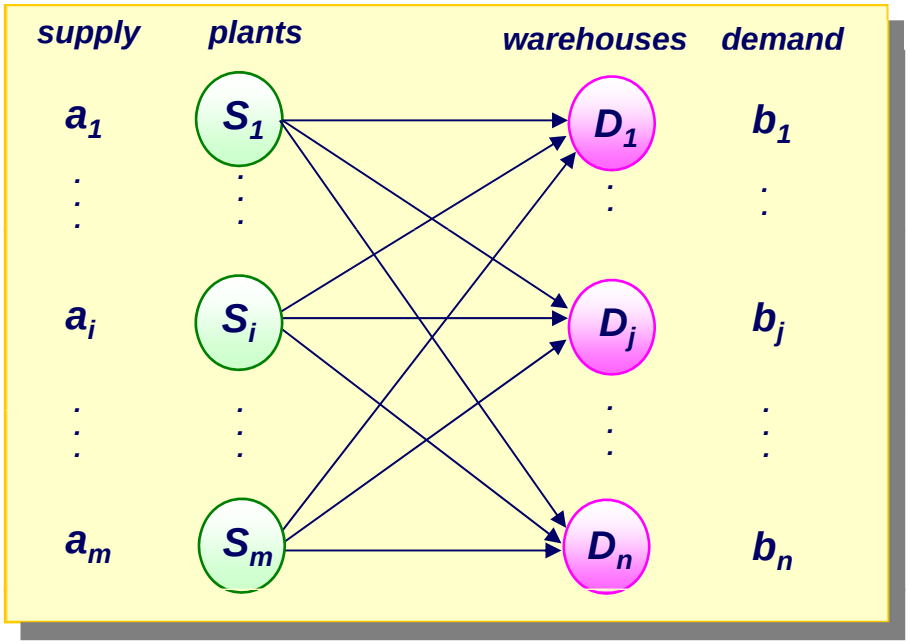
- 从不同的供给起点或来源向不同的终点运送同一种物品，每个终点需要特定数量的物品。
- 问题是：如何分配在每个起点的供给，以便在满足每个终点需求的条件下，优化某个目标，例如运输费用最小。

$$\min \quad z(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ij} \le a_i, \quad i = 1,2,\ldots,m$$

$$\sum_{j=1}^{m} x_{ij} \ge b_j, \quad j = 1,2,\ldots,n$$

$$x_{ij} \ge 0, \qquad \forall i, j$$



where,

x_{ij}：从起点 i 到终点 j 的运输量；

c_{ij}：从起点 i 到终点 j 的单位运输费用；

a_i：起点 i 的供给量；

b_j：终点 j 的需求量。

运输表

- 运输问题通常可用运输表来表示
- 其中行代表起点，列代表终点，*i* 行 *j* 列格中的内容代表决策变量 x_{ij}，相应的费用系数 c_{ij} 放在 (i, j) 格的右上角处。
- 起点的供给量放在最后一列；终点的需求量放在最后一行。

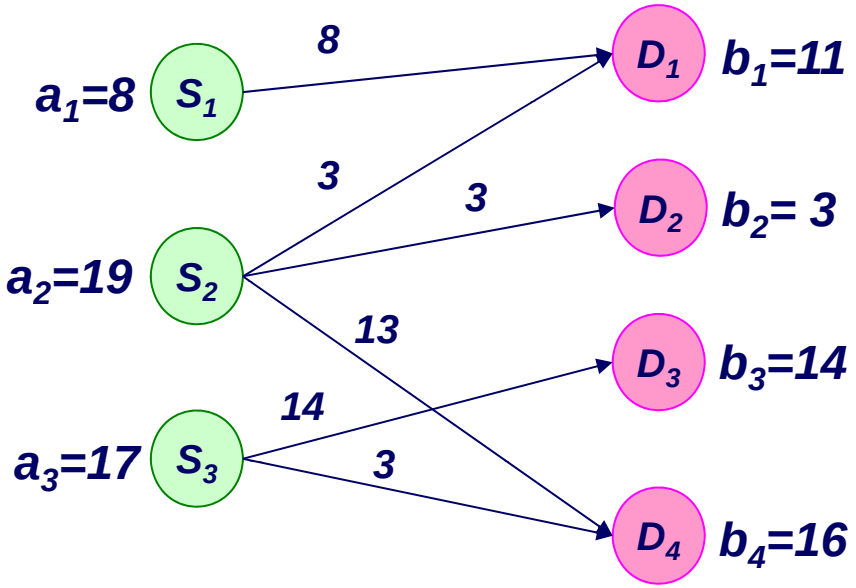
Destination

Origin: from \ to	1	2	…	n	Supply
1	x_{11} c_{11}	x_{12} c_{12}		x_{1n} c_{1n}	a_1
2	x_{21} c_{21}	x_{22} c_{22}		x_{2n} c_{2n}	a_2
…					…
m	x_{m1} c_{m1}	x_{m2} c_{m2}		x_{mn} c_{mn}	a_m
Demand	b_1	b_2	…	b_n	

运输问题解的共同特征

- 有 $m+n-1$ 个基变量，每个变量对应运输表中的一格。
- 基必须是一棵*运输树*，即，在运输表的每一行和每一列中至少存在一个基本单位（即：每个起点都至少要运出物品，每个终点也至少要有物品运入）。
- 基不能包含圈

	D_1	D_2	D_3	D_4	a_i
S_1	B				8
S_2	B	B		B	19
S_3			B	B	17
b_j	11	3	14	16	



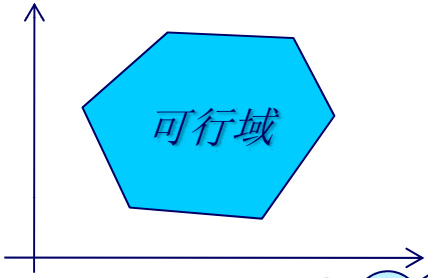
线性规划问题的基础可行解

线性规划的标准形式

$$\max f = \sum_{j=1}^{n} c_j x_j$$

$$s.t. \sum_{j=1}^{n} a_{ij} x_j = b_j, \quad i = 1,2,\cdots,m \quad (b_j \geq 0)$$

$$x_j \geq 0, \quad j = 1,2,\cdots,n$$



约束条件的矩阵表示

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & & \ddots & \\ a_{m1} & a_{m2} & & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \\ b_m \end{bmatrix}$$

通过矩阵变换，可将其中一些列的组合变换为单位矩阵

$$\begin{bmatrix} 1 & 0 & \cdots & 0 & a_{1m+1} & a_{1m+2} & \cdots & a_{1n} \\ 0 & 1 & & 0 & a_{2m+1} & a_{2m+2} & & a_{2n} \\ \vdots & & \ddots & \vdots & & & \ddots & \\ 0 & & & 1 & a_{mm+1} & a_{mm+2} & & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b'_1 \\ b'_2 \\ \\ b'_m \end{bmatrix} \geq 0$$

基础可行解

非零变量->基

零变量->非基

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \\ x_{m+1} \\ x_{m+2} \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b'_1 \\ b'_2 \\ \\ b'_m \\ 0 \\ 0 \\ \\ 0 \end{bmatrix}$$

运输问题的可行性

- 在运输模型的描述当中，假设全部供给和全部需求是相等的（平衡假设），即存在：

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

- 在平衡假设条件下，运输问题通常有可行解。例如：

$$x_{ij} = \frac{a_i b_j}{\sum_i a_i}, \quad i = 1,2,\ldots, m, \quad j = 1,2,\ldots, n$$

- 对于每个可行解，每个运输量 x_{ij}，要受如下限制

$$0 \le x_{ij} \le \min \{a_i, b_j\}$$

- 具有可行解的受限线性规划肯定存在最优解

GA求解

- 染色体的表达方式
  - 基于矩阵的表达
    - 矩阵或许是运输问题的解的最适当的表达方式。
  - 基于生成树的表达
    - 作为一种特殊类型的网络问题，在它们的解中表现为运输树。因此可以采用生成树的方法进行表达。

$$\min \quad z(x) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\text{s. t.} \quad \sum_{j=1}^{n} x_{ij} \le a_i, \quad i = 1,2,..., m$$

$$\sum_{j=1}^{m} x_{ij} \ge b_j, \quad j = 1,2,..., n$$

$$x_{ij} \ge 0, \qquad \forall i, j$$

GA求解—基于矩阵的表达

- 染色体编码
  - 运输问题的分配矩阵可描述为：

$$[X_p]=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

  - 其中 X_p 代表第 p 个染色体，x_{ij} 是相应的决策变量。
- 种群初始化
  - 初始化时，要考虑非负条件和平衡条件，产生满足所有约束的种群。
  - 初始化过程：
  - 初始化过程的基本思想：
    - 从分配矩阵中任选一个决策变量 x_{ij}；
    - 赋给 x_{ij} 尽可能多的可用量；
    - 修改需求和供给数据以保证平衡条件。

begin

$\pi \leftarrow \{1, 2, \cdots, mn\}$;

repeat

从集合 π 中任选一个数 k;

计算相应的行和列;

$i \leftarrow \lfloor (k-1)/n + 1 \rfloor$;

$j \leftarrow (k-1) \bmod n + 1$;

将可用量赋给 x_{ij};

$x_{ij} \leftarrow \min\{a_i, b_j\}$;

修改数据;

$a_i \leftarrow a_i - x_{ij}$;

$b_j \leftarrow b_j - x_{ij}$;

$\pi \leftarrow \pi \setminus \{k\}$;

until π 为空

end

GA求解—基于矩阵的表达

- 交叉操作
  - 假设 a_i, b_j, x_{ij} 都是整数
  - 假设矩阵 $X_1=[x^1_{ij}]$， $X_2=[x^2_{ij}]$ 被选作交叉运算的双亲，交叉操作由三步完成。
  - 步骤1:建立两个临时矩阵 $D=(d_{ij})$ 和 $R=(r_{ij})$

$$d_{ij}=\left\lfloor (x^1_{ij}+x^2_{ij})/2 \right\rfloor, \quad r_{ij}=(x^1_{ij}+x^2_{ij}) \bmod 2$$

    - 矩阵 D 存放两个双亲均值的取整，矩阵 R 记录是否需要取整。两个矩阵的关系由下列等式给出:

$$a_i-\sum_{j=1}^{n} d_{ij}=\frac{1}{2}\sum_{j=1}^{n} r_{ij}, \quad i=1,\ldots,m; \quad b_j-\sum_{i=1}^{m} d_{ij}=\frac{1}{2}\sum_{i=1}^{m} r_{ij}, \quad j=1,\ldots,n$$

    - 矩阵 R 每行每列中 1 的数目是偶数，即行与列的边际和$\sum r_{ij}$, $i=1,\ldots m$, $\sum r_{ij}$, $j=1,\ldots n$ 都是偶整数。
    - 矩阵 R 行（或列）边际和的值等于矩阵 D 行（或列）边际和与相应供给（或需求）之差的两倍。

GA求解—基于矩阵的表达

- 交叉操作
  - 矩阵 ***R*** 行边际和的值等于矩阵 ***D*** 行边际和与相应供给之差的两倍

$$a_i - \sum_{j=1}^{n} d_{ij} = \frac{1}{2}\sum_{j=1}^{n} r_{ij}, \quad i = 1,2,\ldots, m$$

  - 证明：

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

GA求解—基于矩阵的表达

- 交叉操作
  - 步骤2:将矩阵 R 分解为两个矩阵 $R^1=(r^1_{ij})$ 和 $R^2=(r^2_{ij})$

$$R = R^1 + R^2$$

$$\sum_{j=1}^{n} r_{ij}^1 = \sum_{j=1}^{n} r_{ij}^2 = \frac{1}{2}\sum_{j=1}^{n} r_{ij}, \quad i = 1,2,...,m$$

$$\sum_{i=1}^{m} r_{ij}^1 = \sum_{i=1}^{m} r_{ij}^2 = \frac{1}{2}\sum_{i=1}^{m} r_{ij}, \quad j = 1,2,...,n$$

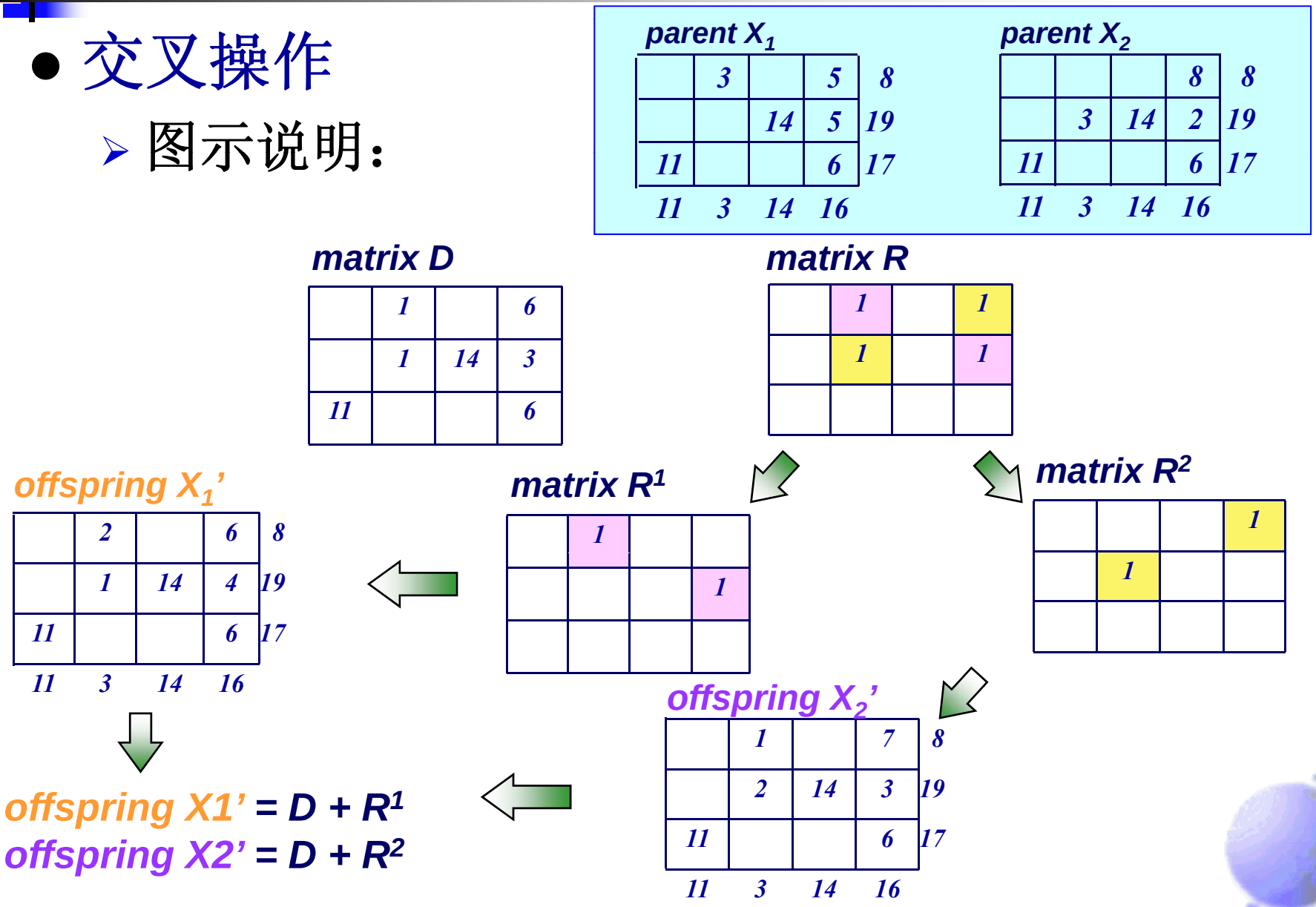
    - 在满足上述条件下，将 R 分成 R^1 和 R^2 有许多可能情况
  - 步骤3:产生两个如下后代 X_1' 、 X_2'

$$X_1' = D + R^1$$

$$X_2' = D + R^2$$

GA求解—基于矩阵的表达

- 交叉操作
  - 图示说明：

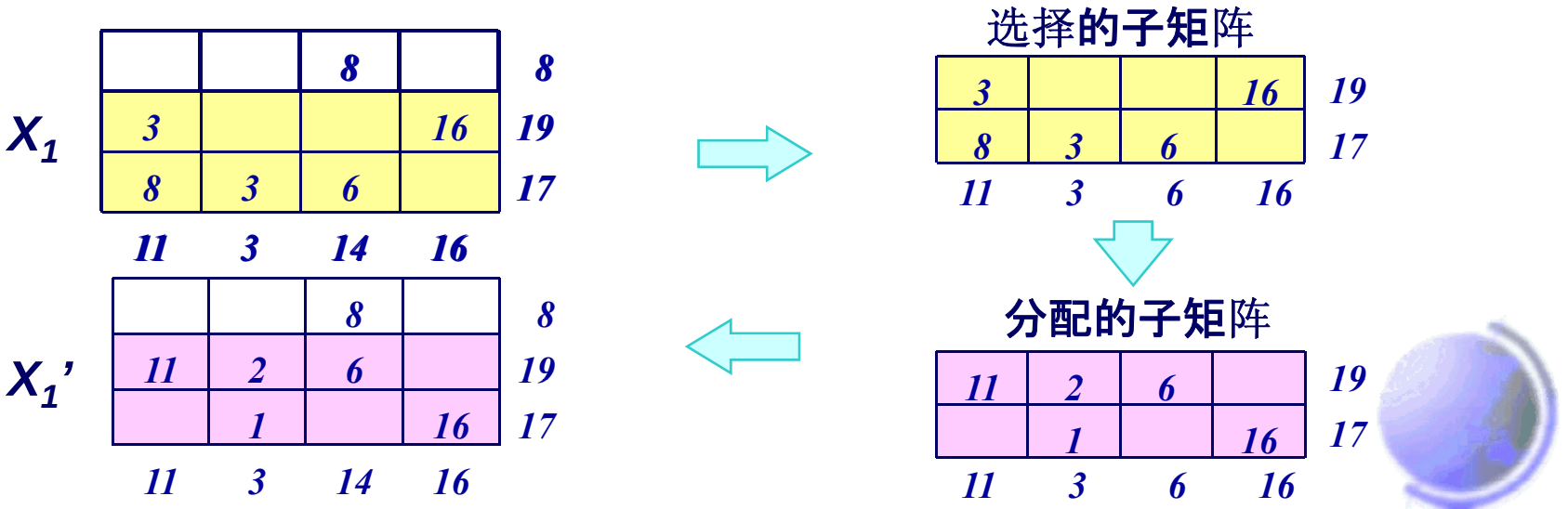


GA求解—基于矩阵的表达

- 变异操作
  - 步骤1：从双亲矩阵中任选 p (≥2) 行，q (≥2)列，建立一个 $(p*q)$ 子矩阵 $Y=(y_{ij})$，其中 y_{ij} 选自双亲矩阵中所选行列的交叉点处的值
  - 步骤2：重新分配子矩阵的物品（运输量）。子矩阵中供给 a_i^y和需求 b_j^y，由如下公式确定：

$$a_i^y = \sum_{j\in\{j_1,\ldots,j_q\}} y_{ij}, \quad i = i_1,\ldots,i_p; \quad b_j^y = \sum_{i\in\{i_1,\ldots,i_p\}} y_{ij}, \quad j = j_1,\ldots,j_q$$

    - 可以采用初始化过程对子矩阵赋值，以满足上述约束
  - 步骤3：用重新分配的子矩阵 Y 中的新元素替换双亲矩阵中的适当元素。



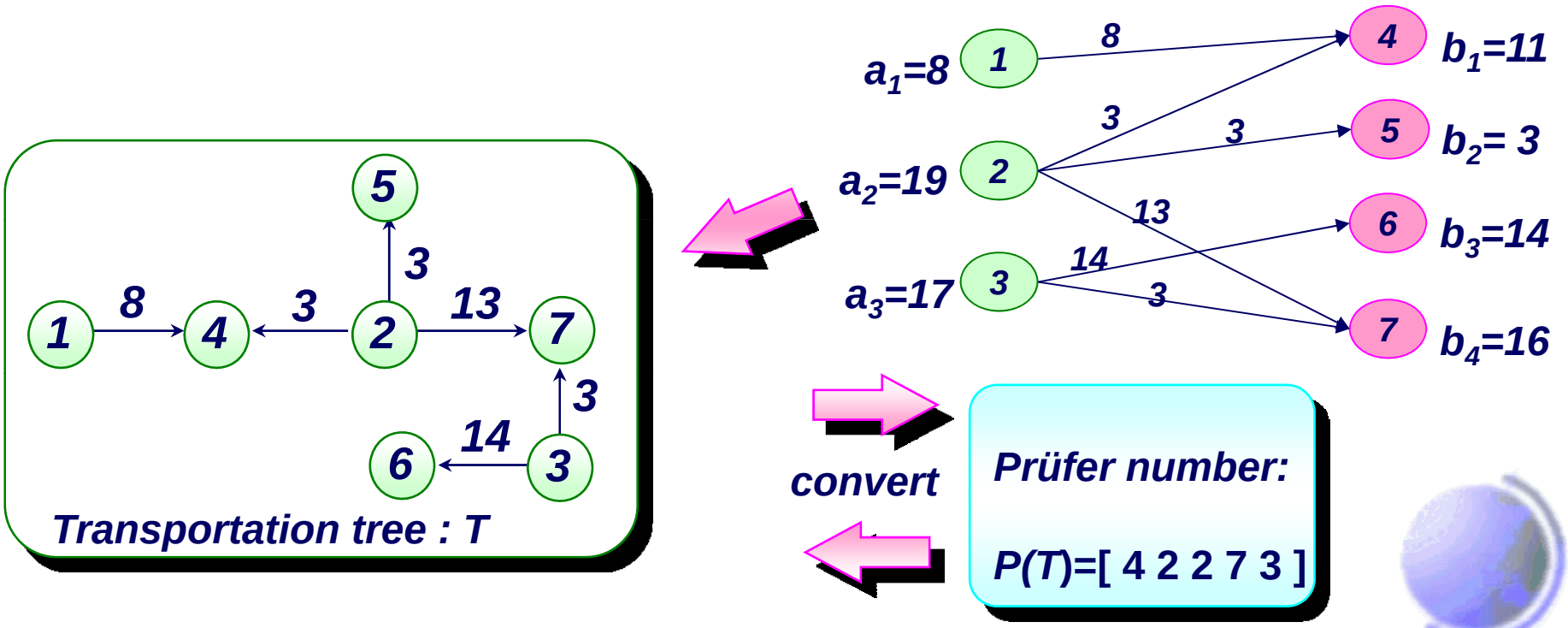
GA求解—基于生成树的表达

- 作为一种特殊类型的网络问题，它们的基础解表现为运输树。因此可以采用生成树的方法进行表达。
- 使用基于生成树的 **Prüfer** 数编码，对于运输问题中的每个染色体仅需要 $m+n-2$ 个存储单元，比矩阵表示法节省大量存储空间。
- 因为运输树是一种特殊类型的树，所以 **Prüfer** 数可能对应不可行的解，因此要给出检验染色体可行性的准则。

- 基于生成树的遗传算法能找到在解空间中运输问题的最优或近似最优解。

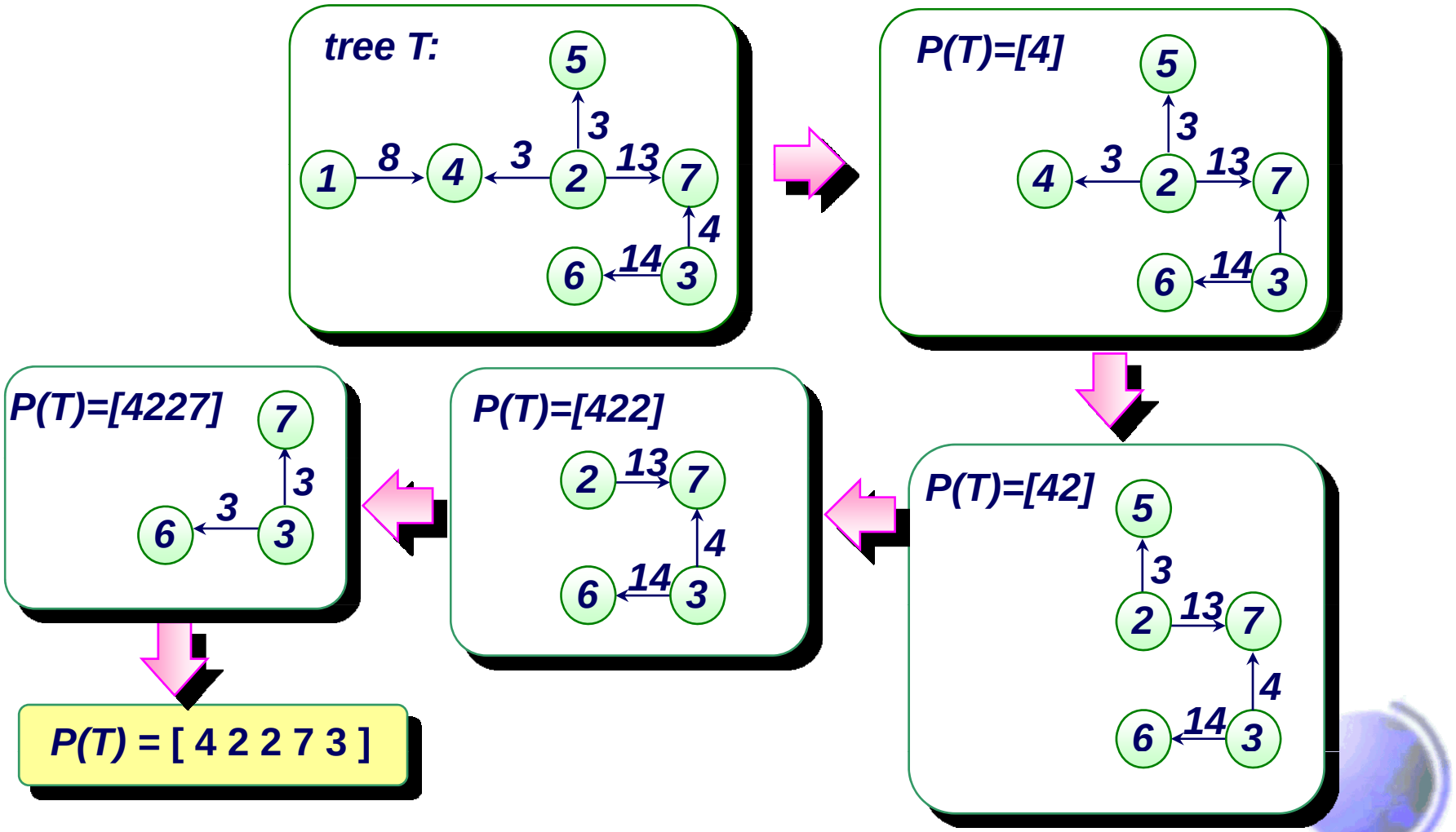
GA求解—基于生成树的表达

- 将运输树转化为 **Prüfer** 数的过程（编码）：
  - 令 *i* 是树 *T* 中数字最小的叶子节点，*j* 为 *i* 的关联节点，则将 *j* 作为 **Prüfer**数，添加到 *P*(*T*) 的右边。
  - 移去节点 *i* 和边 (*i*, *j*)。
  - 重复上述过程，直到只剩下一条边。



GA求解—基于生成树的表达

- 运输树转化为 **Prüfer** 数的图示：



GA求解—基于生成树的表达

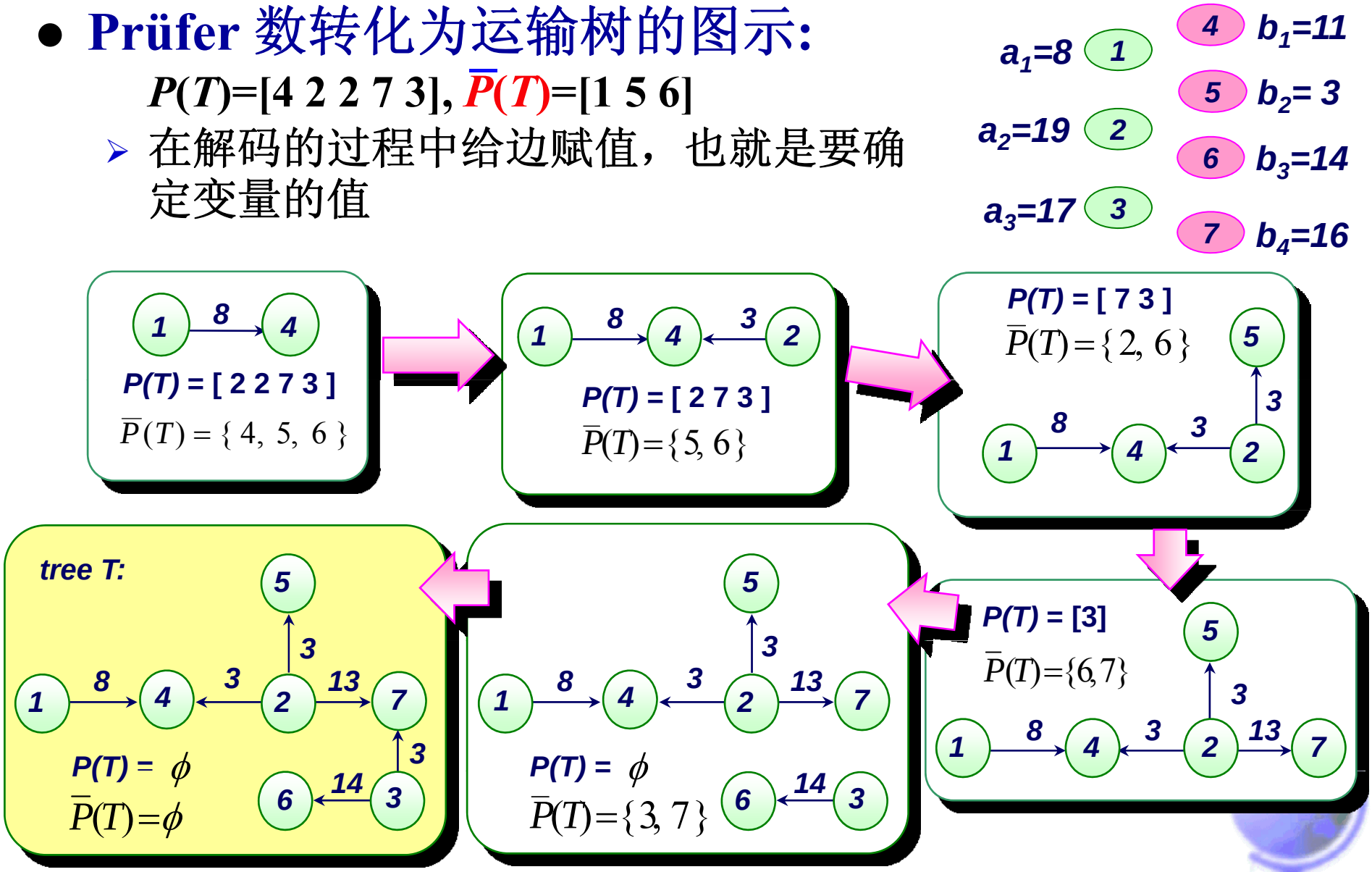
- 将 Prüfer 数转化为运输树的过程（解码）：
  - 令 $P(T)$ 是原 Prüfer数，$\bar{P}(T)$ 是所有不包括在 $P(T)$ 的端点集合
  - 重复下述过程，直至 $P(T)$ 为空为止
    - 令 i 为 $\bar{P}(T)$ 中标号最小的点。令 j 为 $P(T)$ 的最左边的数字。
    - 如果 i 和 j 不同在 S 或 D 中，则添加边 (i,j) 到树 T。否则，从 $P(T)$ 中选择下一个数字 k，它与 i 不在同一个集合中，用 k 交换 j，添加边 (i, k) 到树 T。
    - 从 $P(T)$ 中移去 j (或 k)，从 $\bar{P}(T)$ 中移去 i。如果节点 j (或 k)不再在 $P(T)$ 剩余部分的任何位置出现，则将其放人 $\bar{P}(T)$。
    - 指定边 (i, j) (或(i, k)) 的运输量为 $x_{ij}=\min\{a_i,b_j\}$ ($x_{ik}=\min\{a_i,b_k\}$)，其中 $i\in S$, $j,k\in D$。
    - 更新可用的运输量$a_i=a_i-x_{ij}$, $b_j=b_j-x_{ij}$ (或 $b_k=b_k-x_{ik}$)。
  - 如果在 $P(T)$ 中不再有数字，$\bar{P}(T)$ 中正好有两个点 i 和 j，添加边 (i,j) 到树 T，形成 $m+n-1$ 条边的树。
  - 如果没有可用的运输量可分配，则停止；否则，保留供应点 r 和需求点 s。添加边 (r, s)到树 T，指定边的可用运输量为 $x_{rs}=a_r=b_s$。如果此时存在圈，则移去流量为 0 的边。从而形成具有 $m+n-1$ 条边的树。

GA求解—基于生成树的表达

- **Prüfer 数转化为运输树的图示:**

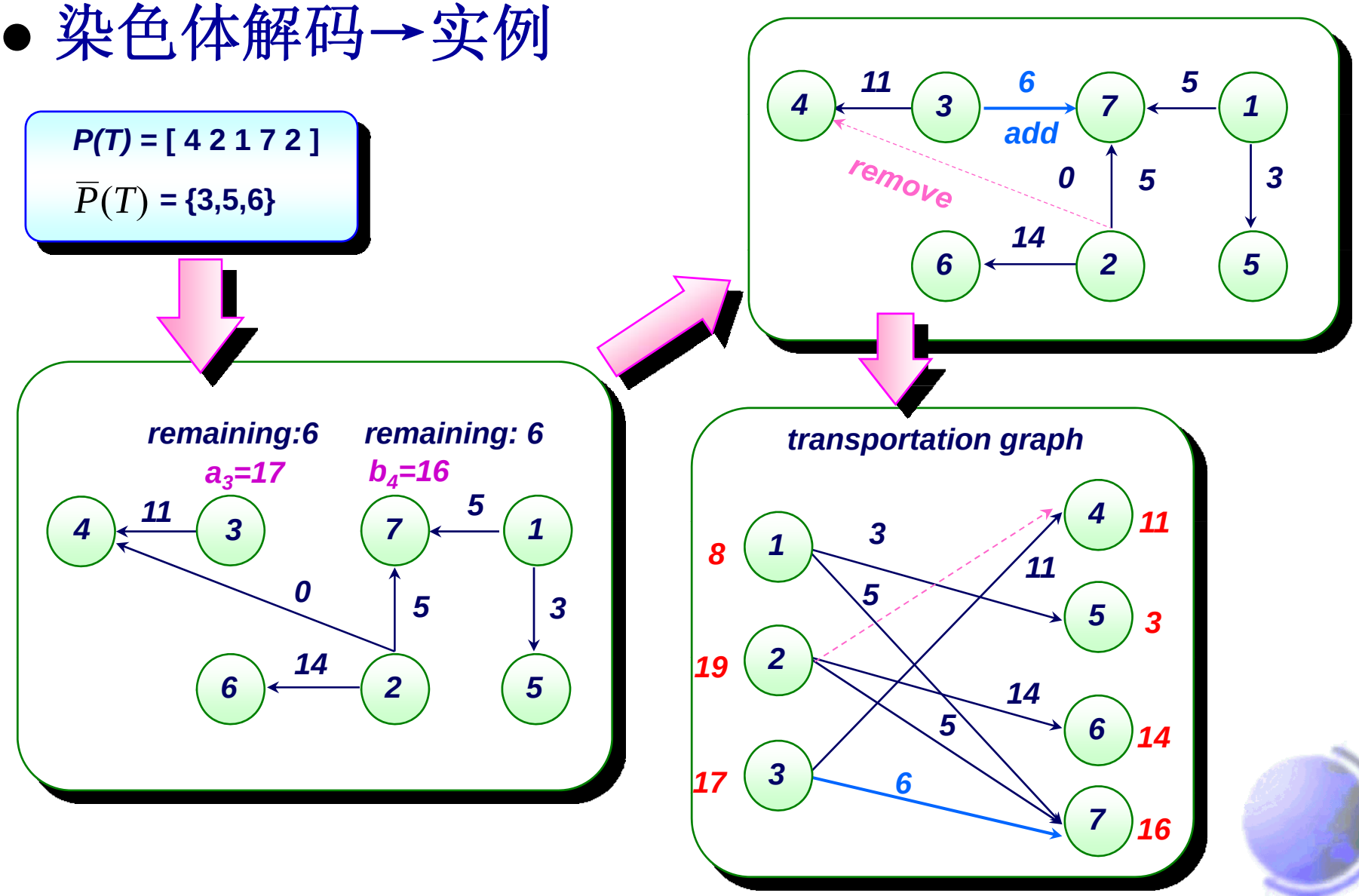
  $P(T)$=[4 2 2 7 3], $\bar{P}(T)$=[1 5 6]

  - 在解码的过程中给边赋值，也就是要确定变量的值



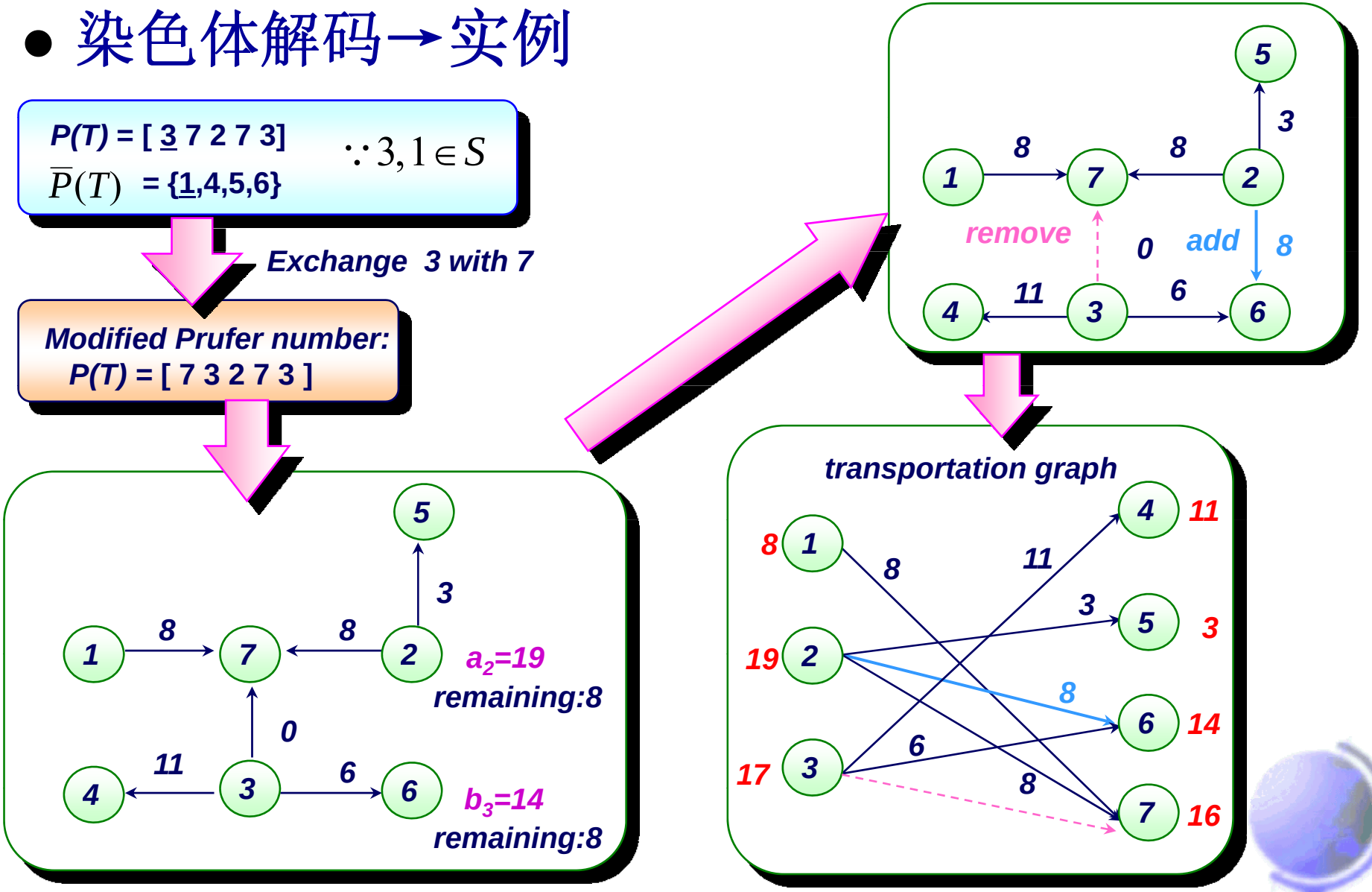
GA求解—基于生成树的表达

- 染色体解码→实例



GA求解—基于生成树的表达

- 染色体解码→实例



GA求解—基于生成树的表达

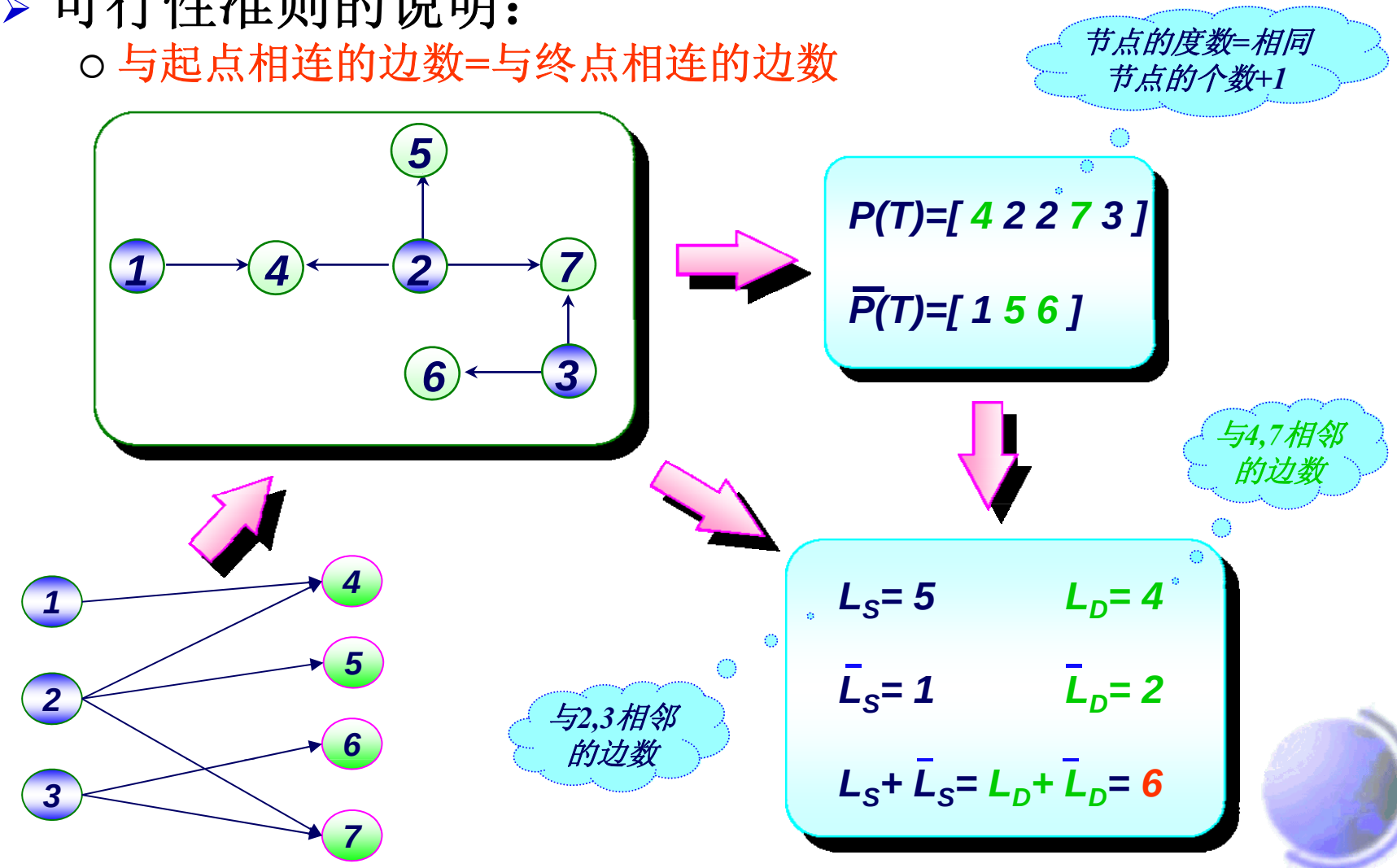
- 种群初始化
  - 在 [1, *m*+*n*] 的范围内, 随机产生 *m*+*n*-2个随机数, 构成 Prüfer 数
  - 问题：
    - 有可能产生不可行解，即非运输树
- 可行性的准则
  - Gen 和 Li 设计了一种用于检验染色体可行性的准则

$$L_S + \overline{L}_S = L_D + \overline{L}_D$$

  - 其中：
    - L_S: 由 Prüfer 数定义的生成树中,与 Prüfer 数中的起点相关的边的总和
    - L_D :由 Prüfer 数定义的生成树中,与 Prüfer 数中的终点相关的边的总和
    - $\overline{L}_S$: 包含在 $\overline{P}(T)$中的起点数 （=与 $\overline{P}$ 中的起点相关的边的总和）
    - $\overline{L}_D$: 包含在 $\overline{P}(T)$中的终点数（=与 $\overline{P}$ 中的终点相关的边的总和）
  - 准则的实质：与起点相连的边数=与终点相连的边数

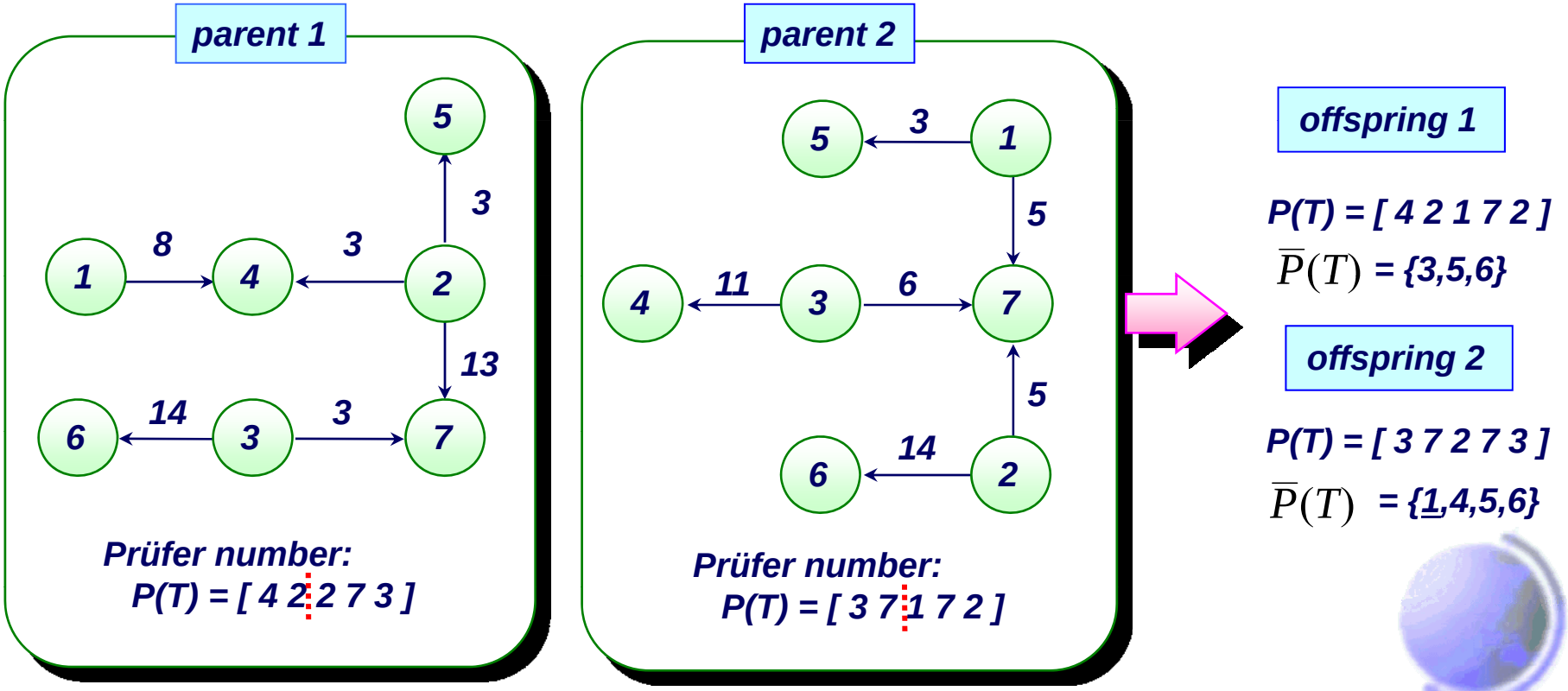
GA求解—基于生成树的表达

- 种群初始化
  - 可行性准则的说明：
    - 与起点相连的边数=与终点相连的边数



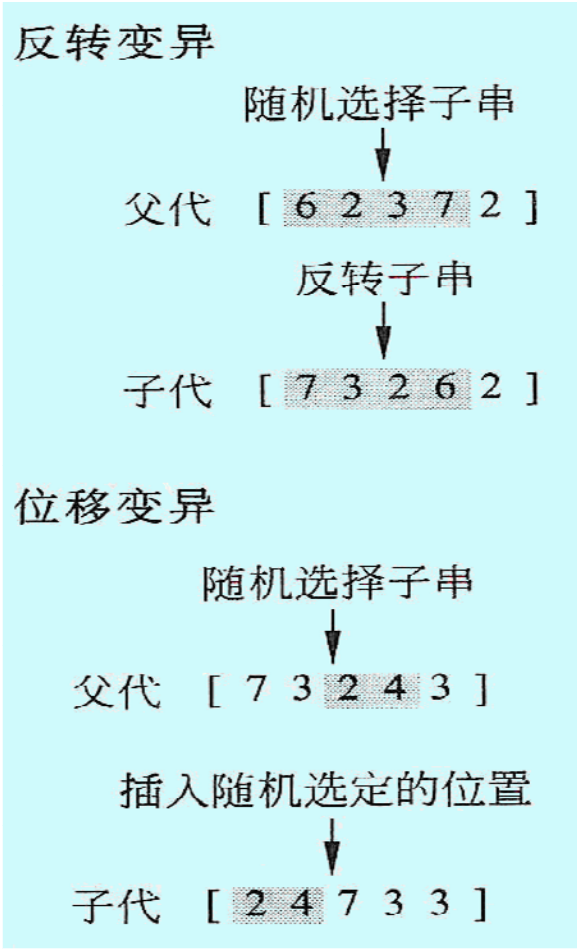
GA求解—基于生成树的表达

- 交叉操作
  - 单点交叉
    - 交叉运算有可能产生不可行的后代，如果将可行性检验嵌入到交叉运算当中，可以保证产生可行的后代。
  - 交叉运算实例：



GA求解—基于生成树的表达

- 变异操作
  - 反转变异
    - 随机地选择染色体中的两个位置，然后将两个位置间的子串反转
  - 位移变异
    - 随机选择位移子串，插入随机选定的位置
  - 如果父代是可行的，则这两种变异永远产生可行的染色体，因为在运算后，可行性准则是不变的



双准则线性运输问题

(Bicriteria Linear Transportation Problem)

双准则线性运输问题

- 实际运输问题中，通常需要考虑费用以外的其它准则。
- 例如:
  - 物品的平均交货时间，
  - 运输的可靠性，
  - 用户可接近性 (accessibility)，
  - 产品损失 (deterioration)等
- 这些运输问题实际上是多目标问题，因而，可表述为多目标线性规划问题。
- 双准则线性运输问题 (BLTP) 是多目标运输问题的一个特例，只有两个目标。

模型描述

- 考虑下述两个目标：最小化全部费用和全部损失。给定 ***m*** 个起点和 ***n*** 个终点, **BLTP**问题可描述为：

$$\min \quad z_1 = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}^{1} x_{ij}$$

$$\min \quad z_2 = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}^{2} x_{ij}$$

$$\text{s. t.} \quad \sum_{j=1}^{n} x_{ij} \le a_i, \quad i = 1,2,\ldots, m$$

$$\sum_{j=1}^{m} x_{ij} \ge b_j, \quad j = 1,2,\ldots, n$$

$$x_{ij} \ge 0, \qquad \forall i, j$$

where,

x_{ij}：从起点 *i* 到终点 *j* 的运输量；

c^1_{ij}：从起点 *i* 到终点 *j* 的单位运输费用；

c^2_{ij}：从起点 *i* 到终点 *j* 的单位运输损失；

a_i：起点 *i* 的供给量；

b_j：终点 *j* 的需求量。

- 一个是供给约束，一个是需求约束
- 变量和约束与**LTP**是完全一样的，区别仅在于有两个目标。

多目标优化的基本概念

- 多目标优化问题（***multiple objective optimization problem***）起源于许多实际复杂系统的设计、建模与规划问题，几乎每个重要的现实生活中的决策问题都要在考虑不同约束的同时处理若干相互冲突的目标，这些问题大大增加了问题的复杂程度。
- 自20世纪60年代以来，多目标优化问题吸引了许多研究人员的注意，许多学者为之做出了重要的贡献，其中Pareto被公认为是本领域的开创者之一。
- 模型描述
  - 不失一般性，模型可以描述如下：

$$\begin{aligned} \max \ & \{z_1 = f_1(\boldsymbol{x}), z_2 = f_2(\boldsymbol{x}), \cdots, z_q = f_q(\boldsymbol{x})\} \\ \text{s. t.} \ & g_i(\boldsymbol{x}) \leq 0, \quad i = 1, 2, \ldots, m \end{aligned}$$

  - 决策空间（***decision space***）

$$S = \{\boldsymbol{x} \in R^n | g_i(\boldsymbol{x}) \leq 0, \quad i = 1, 2, \ldots, m, \quad \boldsymbol{x} \geq 0\}$$

  - 判据空间（***criterion space***）

$$Z = \{\boldsymbol{z} \in R^q | z_1 = f_1(\boldsymbol{x}), z_2 = f_2(\boldsymbol{x}), \cdots, z_q = f_q(\boldsymbol{x}), \quad \boldsymbol{x} \in S\}$$

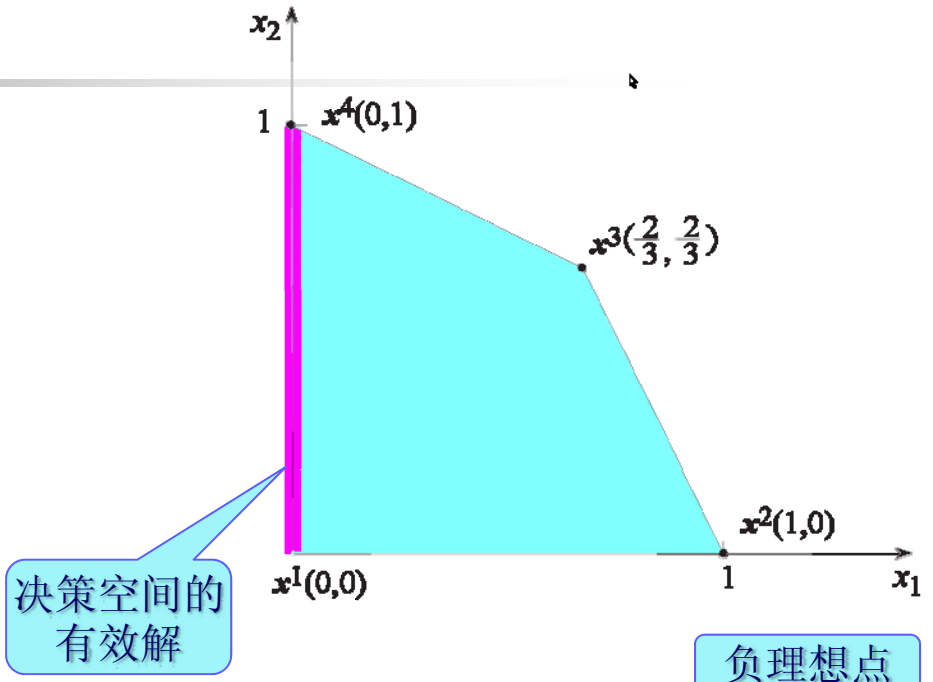
Pareto最优解

- 多目标问题的非支配解（*nondominated solutions*）：
  - 单目标问题通常可以找到一个最优解，对于多目标问题来讲，由于目标之间的冲突（矛盾），严格意义上的最优解通常是不存在的。
  - 多目标问题通常存在一个“解的集合”，而不是一个解，他们之间不能简单地比较好坏，
  - 对这种解来说：对于任何目标函数，在不牺牲其他目标的情况下，解就不能改进。
  - 换句话说：不存在一个比该解的各个目标函数值都好的解
  - 这样的解在判据空间通常称为：
    - 非支配解（或非控解）、
    - **Pareto最优解（*Pareto optimal solutions*）**
  - 在决策空间通常称为：
    - 有效解（*efficient solutions*）、
    - 非劣解（*noninferior solutions*）

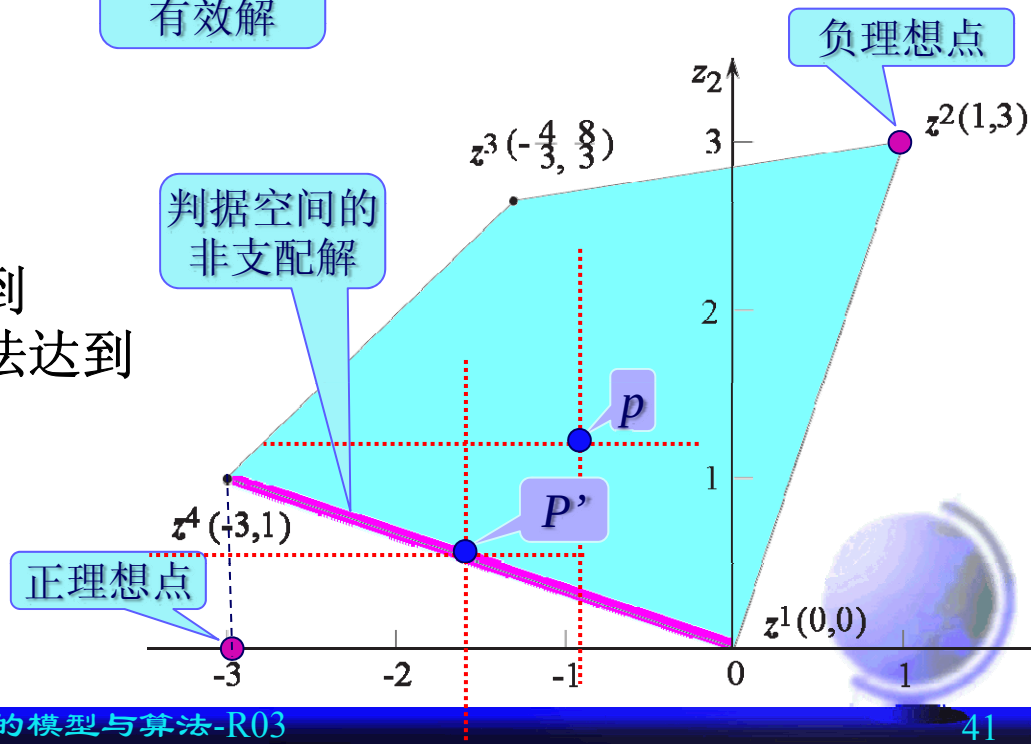
Pareto最优解

- 实例说明：

$$\begin{aligned}
&\min\ z_1(x_1,x_2) = x_1 - 3x_2\\
&\min\ z_2(x_1,x_2) = 3x_1 + x_2\\
&\text{s. t. } g_1(x_1,x_2) = x_1 + 2x_2 - 2 \le 0\\
&\qquad g_2(x_1,x_2) = 2x_1 + x_2 - 2 \le 0\\
&\qquad x_1, x_2 \ge 0
\end{aligned}$$



- 正理想点（解）
  - $Z^+=(Z_1^+,\ldots,Z_q^+)$
  - 由各目标的最好值构成
  - 单个目标的最优相对容易找到
  - 多个目标的同时最优通常无法达到
- 负理想点（解）
  - $Z^-=(Z_1^-,\ldots,Z_q^-)$
  - 由各目标的最坏值构成
  - 负理想点通常也无法达到



偏好结构

- 在实际的决策过程中，通常需要从无法比较好坏的非支配解中选择一个作为给定问题的最终解。然而，如果不提供对于不同目标附加的偏好信息，很可能无法从解中进行选择。
- 偏好(preferencc)的概念：
  - 偏好是通过采用某人对目标的价值判断来对有效集合中无法进行比较的解给出排序(order)。
  - 偏好反映了某人根据对问题事先掌握的知识对所有目标进行的折中或者对某个目标进行的强调。
- 给定了偏好就可将非支配集中可选的解进行排序，然后获得最终解，这就是通常决策过程的结果。这个最终解称作最优妥协解。
- 偏好通常用二元关系表示，对于一组给定对u和v，可能且仅可能发生下面一种关系：
  - u比v好，或对u的偏好大于v，用$u \succ v$来表示
  - u比v差，或对u的偏好小于v，用$u \prec v$来表示
  - u与v相当，或对u与v同等偏好，用$u \sim v$来表示
  - u与v的偏好关系未定义，用 u ？ v来表示
- 决策空间中所有点对的偏好关系，都可以通过上述4个算子中的一个来指定。

基本求解方法

- 对于多目标问题，一般希望决策过程：
  - 要么获得妥协解或偏好解（一个解）
  - 要么确定所有非支配解（一组解）
- 因此，主要存在两类求解方法：
  - 基于偏好的方法（**preference-based approach**）
    - 试图获得妥协解或者偏好解
    - 需要决策者能够用正式和有结构的方式来清晰表述其偏好
    - 如果有对于目标相对重要性的某些想法，就可以对偏好进行限定。有了偏好信息就可以确定妥协解或偏好解。
  - 产生式方法（**generating approach**）
    - 用于确定整个**Pareto**解集或者解集的近似
    - 需要决策者从整个**Pareto**解中做选择来进行必要的价值判断
    - 在高维问题时，选择将变得非常困难与复杂。
    - 如果没有对目标偏好结构的先验知识，就只能采用产生式方法来检验所有的非支配解。

基本求解方法

- 权重和方法（**weighted-sums approach**）：
  - 为每个目标函数分配权重并将其组合成为一个目标函数的基本思想由**Zadeh**首先提出。

$$\max\ z(\boldsymbol{x}) = \sum_{k=1}^{q} w_k f_k(\boldsymbol{x})$$
$$\text{s. t. } \boldsymbol{x} \in S$$

  - 权重可以理解为目标和目标之间的相对重要性或价值，可以看做是对目标的偏好。因此问题的最优解涉及了一个特殊的偏好结构。
  - 该问题的最优解，在所有权重都是正数的前提下，是一个非支配解。
  - 由于权重和方法采用的数值次序，不存在偏好比较中的不明确情况。对于任意两点来说，要么一点好于另一点，要么一点差于另一点，要么二者相当。

基本求解方法

- 效用函数方法（utility function approach）：
  - 效用函数是偏好结构的一种数学表示，它将判据空间中的点映射为实数，数值越大则表明其对应的判据空间中点的偏好程度越高。
  - 对于给定的偏好结构，如果可以构造效用函数$U(\cdot)$，则求解多目标优化问题的理想方法就是求解效用函数规划：

$$\max\{U(z_1,\ldots,z_q)|z_1=f_1(\boldsymbol{x}),\ldots,z_q=f_q(\boldsymbol{x}),\ \ \boldsymbol{x}\in S\}$$

  - 与权重和方法相同的是，由于效用函数给出的是数值次序，不存在偏好比较中的不明确情况。
  - 实际上，很难获得偏好结构的特定效用函数，即使获得一般也是非线性的，求解困难。
  - 然而效用方法从概念上讲是很有用的。

基本求解方法

- 妥协方法（compromise approach）：
  - 妥协方法可以看作一种根据距离函数进行目标搜索行为的数学表示。
  - 妥协方法寻找与理想点最近的解。鉴于理想点通常无法达到，如果没有其他可选点来解决目标的冲突，就必须进行妥协。
  - 为选取妥协解 z 而不是找到理想点，通常设置一个后悔函数(regret function)，可以用下面的距离函数来近似：

$$r(z) = \|z - z^*\|$$

    - 其中 $\|z\text{-}z^*\|$ 是在某种特定范数意义上从 z 到 z^* 的距离。
    - 通常采用的 L_p 范数比较清晰，当 $p \geq 1$ 时，具有如下的形式：

$$r(z;p) = \|z - z^*\|_p = \left[\sum_{j=1}^{q} |z_j - z_j^*|^p\right]^{1/p}$$

    - L_p 范数意义下的妥协解，就是最小化后悔函数 $r(z;p)$，而获得的点 $z \in Z$

  - 与权重和方法和效用函数方法相同的是，由于后悔函数给出的是数值次序，不存在偏好比较中的不明确情况。

基本求解方法

- 字典顺序方法（lexicographic ordering approach）：
  - 在这种类型的偏好中，次序比目标更重要。
  - 对于 $z=(z_1, z_2,\ldots, z_q)$进行索引，使得对所有 $k=1,2,\ldots,q-1$，都有第k个元素压倒性地比第$(k+1)$元素重要。
  - 字典顺序偏好定义如下：
    - 对点z^1的偏好大于z^2的偏好，当且仅当下面两种情况有一种发生：
    - ① $z_1^1 > z_1^2$，或者
    - ② 存在某些 $r \in \{1,2,\ldots,q\}$使得$z_r^1 > z_r^2$，同时满足对于 $i=1,2,\ldots,r-1$都有$z_i^1 = z_i^2$。
  - 不同的两点的字典顺序一定不同，因此不存在偏好比较中的不明确情况。

基本求解方法

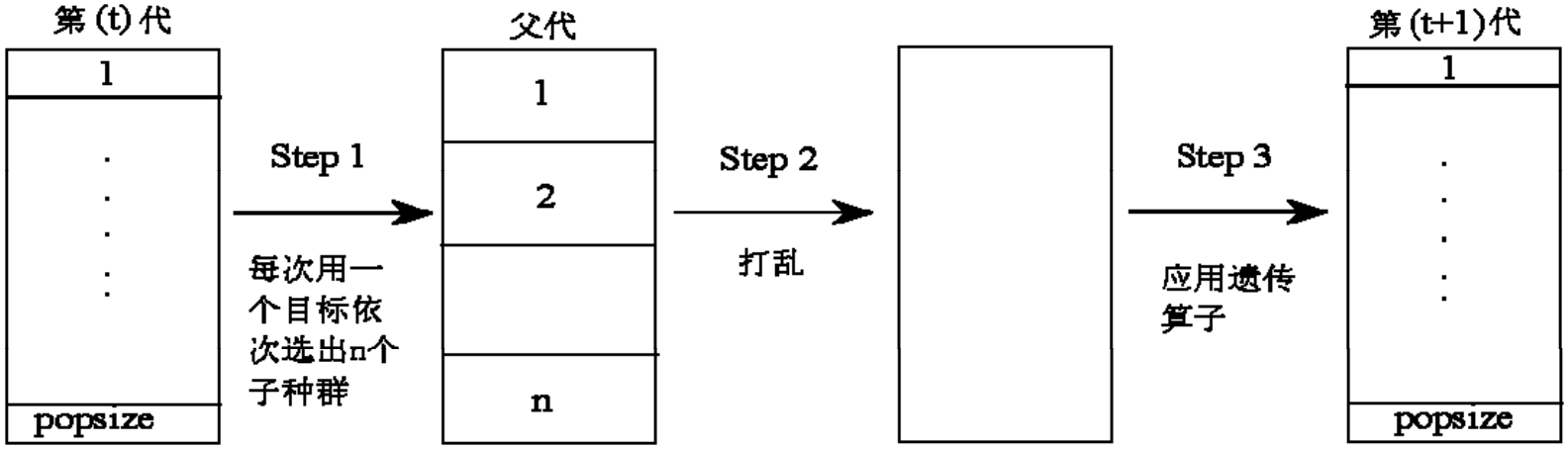
- **Pareto方法：**
  - **Pareto**方法假设关于对目标偏好的任何信息都不存在，
  - 我们知道的所有信息就是对于每个目标 z_j，其值越大越得到偏好。
  - **Pareto**偏好定义如下：
    - 对于判据空间 Z 上任意两点z^1和z^2，点z^1偏好于点z^2，
    - 当且仅当$z^1 \geq z^2$，
    - 即至少存在一个元素 r 满足$z_r^1 > z_r^2$，其他元素满足$z_k^1 \geq z_k^2$，$k=1,2,\ldots,q,\ k \neq r$
  - 采用**Pareto**方法，我们将处理偏好比较中的不明确情况。
    - 而其他方法，一旦得到恰当的后悔函数、效用函数或权重系数，就不存在偏好比较中的不明确情况，问题成为一维比较或数学规划问题。

遗传多目标优化

- 用遗传算法求解多目标优化问题的关键，就是如何根据多个目标来确定个体的适应值，从而实现进化操作。
- 有关适应值分配机制，得到了广泛的研究，粗略分类如下：
  - 向量评价方法
  - 权重和方法
  - 基于**Pareto**的方法
  - 妥协方法
  - 目标规划方法
- 根据偏好信息和适应值函数结合程度的不同，
  - 有的方法给定全部偏好信息，
  - 有的方法不给任何偏好信息。

遗传多目标优化

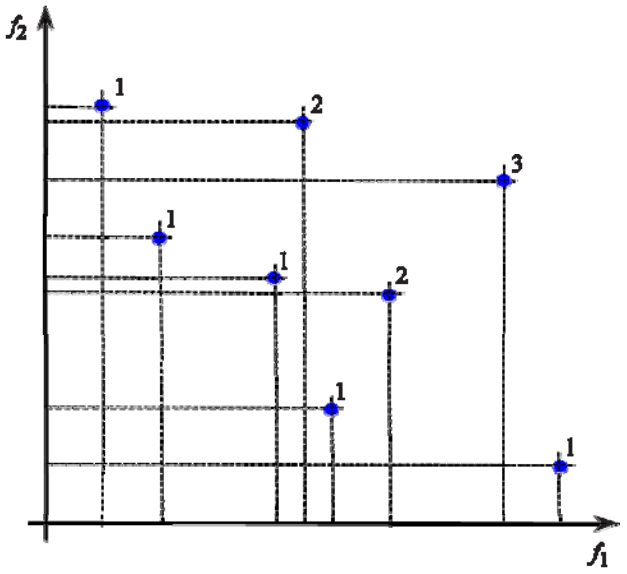
- 向量评价方法(vector evaluation approach)
  - 该方法采用向量形式而不是标量适应值形式来评价染色体
  - 每代中，选择过程是一个循环，在每次循环的过程中，根据每个目标选出若干下一代中的优秀个体（即子种群），然后整个种群被完全打乱并执行交叉和变异操作。目的是促进不同子种群个体之间的交配。



  - 这种方法保护了单个目标上最优个体的生存，同时为那些在多于一个目标上好于平均适应值的个体提供了合理的被选择概率。

遗传多目标优化

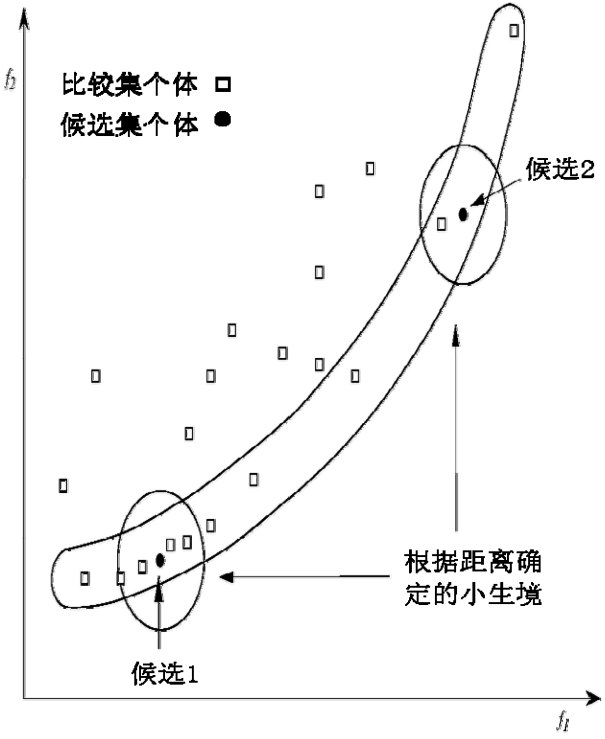
- **Pareto排序方法**
  - 这种方法对当前种群中的非支配个体分配次序1，并将其移去；然后从当前种群中选出非支配个体并对其分配次序2，依次类推。
    （两个求极小问题的排序结果如图）
  - 然后根据排序对个体分配选择概率。次序号越高，选择概率越小。



  - **Pareto解的概念**
    - 在严格意义上，GA中使用的术语Pareto解，与传统方式的含义不同。在某些GA中，每代都需要确定Pareto解。
    - 由于每代中仅包含原始问题的部分解，因此Pareto解仅表示关于所有当前获得的解的非支配点的含义。
    - 某一代中的非支配点，可能被以后产生的新解所支配。
    - 因此GA给定的某一代，从中获得的Pareto解可能是真正的Pareto解，也可能不是。虽然无法保证GA能够产生给定问题的Pareto解，但GA会提供对Pareto解很好的近似。

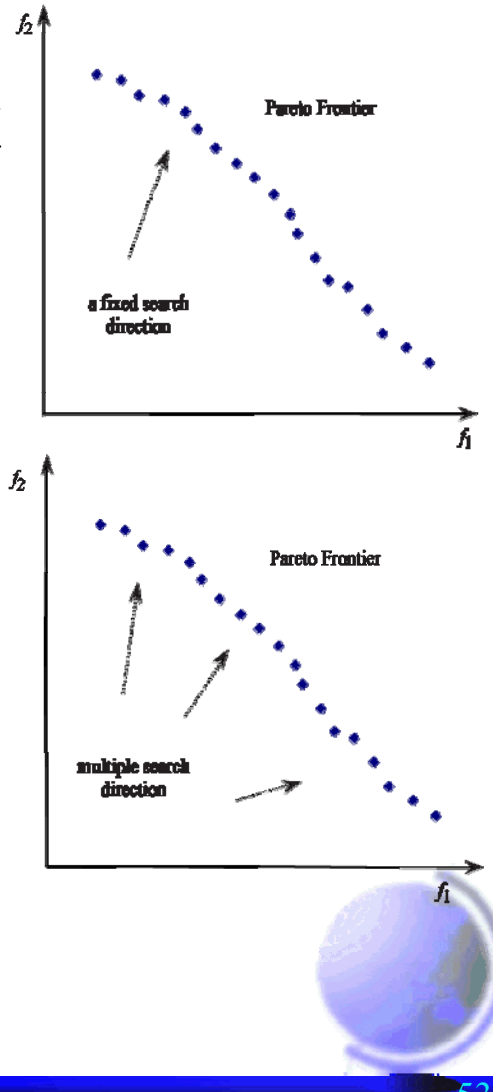
遗传多目标优化

- **Pareto竞争方法**
  - 这种方法从种群中随机选出两个候选个体。同时还从种群中随机选出一组比较个体集。然后每个候选个体与比较集中的每个个体进行比较。可能产生两种结果：
    - 如果一个候选个体被比较集支配，而另一个不被支配，则非支配候选个体被选出来进行复制。
    - 如果两个候选个体都被比较集支配或都支配比较集，则选择具有较小小生境数（周围一定距离内的个体数量较少）的个体选为优胜者。
  - 最大化第一个目标，最小化第二个目标的实例如图所示。
    - 两个候选个体都是非支配点，但第二个候选点的小生境数较少，故选第二个点
    - 选择具有较小小生境数的个体，有利于维持种群的多样性



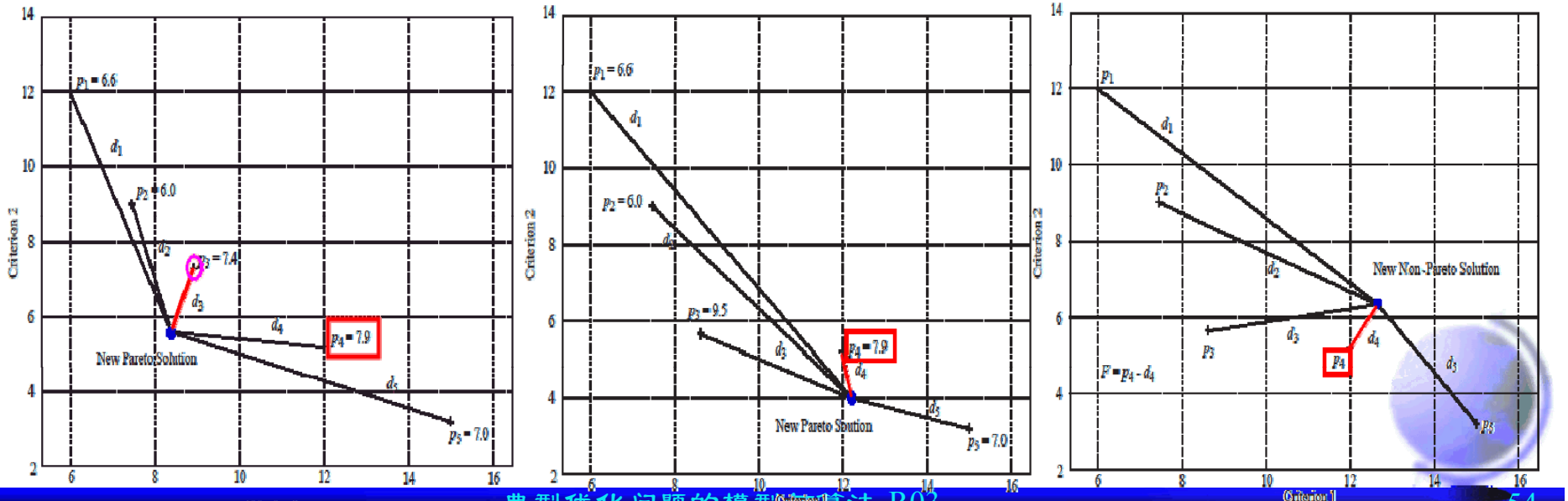
遗传多目标优化

- 权重和方法
  - 从概念上讲，权重和方法可以看作是将多目标优化中采用的方法向遗传算法进行的扩展。
  - 但是，在遗传算法中使用的方法与传统方法在本质上有很大不同。
    - 传统方法用来获得妥协解，需要给定合适的权重向量，而对于给定的问题，通常很难获得一组合适的权重。
    - 在遗传算法中，并不一定需要良好的权重向量来使遗传算法运行。
  - 最近提出了3种权重设置的方法：
    - 固定权重方法
      - 具有向着判据空间中一个固定的点所在的区域进行搜索的趋势
    - 随机权重方法
      - 使遗传算法具有可变搜索方向，具有在Pareto前沿面上进行均匀搜索的能力
    - 适应性权重方法



遗传多目标优化

- 距离方法
  - 基本思想是根据每个个体与前一代Pareto解之间的距离来分配其适应值。该方法采用潜在值的概念，为每个Pareto解分配一个潜在值。
  - 现存Pareto解可能有不同的潜在值，新产生的个体有3种情况：
    - 是一个Pareto解，支配一些或所有当前的Pareto解；
      - 适应值=最大潜在值+最小距离，移去Pareto解集中被支配解加入新解（潜在值=适应值）
    - 是一个Pareto解，但不支配任何当前的Pareto解；
      - 适应值=最近解的潜在值+最小距离，新解加入Pareto解集中（潜在值=适应值）
    - 是一个解，至少被一个现存的Pareto解支配
      - 适应值=最近解的潜在值-最小距离，
  - 每代更新结束后，将所有Pareto解的潜在值更新为最大潜在值。



遗传多目标优化

- 妥协方法
  - 此方法根据某种距离度量方式来确定与理想解最近的解
  - 加权范数 L_p 常被用作距离度量方式。

$$r(z;p,w) = ||z - z^*||_{p,w} = \left[\sum_{j=1}^{q} w_j^p |z_j - z_j^*|^p\right]^{1/p}$$

  - 其中的Z^*表示判据空间Z中的正理想点，权重 w 被分配给目标并用来强调其不同的重要程度。
  - 理想解实际无法达到，但却可以很好地成为评价可达到的非支配解的标准。
  - 然而，对于许多实际问题来说，寻找理想点也是困难的，
  - 为克服这个困难，在遗传迭代过程中可采用代理理想点来替代理想点

$$z_i^{\min} = \min\{z_i(\boldsymbol{x}) \mid \boldsymbol{x} \in P\}$$ $i=1,\ldots,q$, P 表示当前种群

  - 代理理想点在每代中很容易获得，随着进化过程，代理理想点会最终接近真实理想点。

遗传多目标优化

- 目标规划方法（**goal programming approach**）
  - 目标规划方法的基本思想是给每个目标函数建立明确的数值目标，然后寻找一个其目标函数值到对应目标的偏差之加权和最小的解。
  - 本质上存在两种目标规划问题：
    - 无优先权目标规划，基本上所有目标都具有相同的重要性；
    - 优先权目标规划，目标中存在优先级别的层次。
  - 非线性目标规划的一般形式如下：

$$\min \quad z_0 = \sum_{k=1}^{q} \sum_{i=1}^{m_0} p_k (w_{ki}^+ d_i^+ + w_{ki}^- d_i^-)$$

$$\text{s. t.} \quad f_i(x) + d_i^- - d_i^+ = b_i, \qquad i = 1, 2, \cdots, m_0$$

$$g_i(x) \le 0, \qquad i = m_0 + 1, \cdots, \bar{m}_1 (= m_0 + m_1)$$

$$h_i(x) = 0, \qquad i = \bar{m}_1 + 1, \cdots, m (= \bar{m}_1 + m_2)$$

$$d_i^-, d_i^+ \ge 0, \qquad i = 1, 2, \cdots, m_0$$

where

p_k：第 k 个抢先优先权 ($p_k \gg p_{k+1}$ for all k)

d_i^+：第 i 个目标超过预期的正偏差变量

d_i^-：第 i 个目标不足预期的负偏差变量

w_{ki}^+：在优先权 p_k 上分配给 d_i^+ 的正权重

w_{ki}^-：在优先权 p_k 上分配给 d_i^- 的负权重

b_i：目标 i 的目标值或期望级别

q：优先权个数

m_0：目标约束个数

……

双准则线性运输问题的GA求解

- 双准则线性运输问题与一般的线性运输问题的区别仅在于目标函数，
  - 因此，应用GA求解，从染色体编码到交叉变异都可以采用一般线性运输问题的方法。
- 在双目标情况下：
  - 两目标通常本质上是相互矛盾的，
  - 因此我们不是寻找一个最优解，而是要寻找一个 Pareto解集。
- 求解 BLTP 问题，以获取Pareto解集的关键：
  - 在于如何对染色体进行评价，
  - 也就是如何根据两个目标来确定染色体适值的问题。
- Yang 和 Gen 采用“自适应移动线”技术建立了一种求加权和的方法，可以迫使遗传搜索去探索 BLTP 在目标空间中 Pareto 解的集合。

GA求解

- 适值函数：
  - 令 E 代表迄今为止找到的非支配解集合。E 中有两个特殊点：一个点包括 z^1_{min} 而另一个点包括z^2_{min}。这两个点分别表示为(z^1_{min}, z^2_{max})和(z^1_{max}, z^2_{min})，其中：

$$z^1_{\min} = \min \{z^1(\mathbf{x}) \mid \mathbf{x} \in E\}$$
$$z^1_{\max} = \max \{z^1(\mathbf{x}) \mid \mathbf{x} \in E\}$$
$$z^2_{\min} = \min \{z^2(\mathbf{x}) \mid \mathbf{x} \in E\}$$
$$z^2_{\max} = \max \{z^2(\mathbf{x}) \mid \mathbf{x} \in E\}$$

  - 依据特殊点构造新的目标函数：

$$z = \alpha z^1 + \beta z^2 \quad \text{其中} \quad \begin{aligned} \alpha &= | z^2_{\max} - z^2_{\min} | \\ \beta &= | z^1_{\max} - z^1_{\min} | \end{aligned}$$

  - 将目标函数映射为适值函数，可以描述为：

$$eval(x^k) = C - z^k = C - \sum_{i=1}^{m} \sum_{j=1}^{n} (\alpha c^1_{ij} + \beta c^2_{ij}) x^k_{ij}$$

GA求解

- 关于目标的解释：
  - 在判据（目标）空间中，连接点 (z^1_{min}, z^2_{max}) 和点 (z^1_{max}, z^2_{min}) 的线将判据空间分为两部分：
    - 一部分包含正的理想解 Z^+；（目标最好的情况）
    - 一部分包含负的理想解 Z^-。（目标最坏的情况）
  - 可行解空间 F 也相应地被分为两部分：
    - $F^+ = F \cap Z^+$
    - $F^- = F \cap Z^-$
  - 容易证明 F^+ 中的解比 F^- 中的解具有较高的适值。所以 F^+中的染色体具有更多的机会进入下一代。
  - 在每代中 **Pareto** 解集被更新，两个特殊点也要被更新。这表明随着进化过程的进行，由两个特殊点构成的直线将逐步沿着从负理想点向正理想点的方向移动。
  - 换言之：适值函数将迫使遗传搜索去探索目标空间中的那些非支配解。

GA求解

- 求解步骤：
  - ① 参数设置：种群大小、交叉率、变异率、最大代数，集合E 的大小等；
  - ② 初始化：产生初始种群，确定集合 E；
  - ③ 交叉、变异操作；
  - ④ 修改集合 E：
    - a) 计算双亲和后代的每个染色体的目标值；
    - b) 通过对目标值的比较，确定非支配解
    - c) 向集合 E 中增加新的非支配解和从集合 E 中删除支配解
    - d) 计算 E 中新的特殊点
  - ⑤ 评估：计算每个染色体的适值；
  - ⑥ 选择：在双亲和后代中选择最好的染色体进入下一代；
  - ⑦ 终止检查：如果不满足终止条件，转步骤 3
  - ⑧ 将集合 E 中的解作为最终解输出。

双准则三维运输问题

(Bicriteria Solid Transportation Problem)

双准则三维运输问题

- 三维运输问题 **STP (Solid Transportation Problem)** 是基本运输问题的一般化。
- 当存在多种运输工具运送物品时，就有必要考虑这类特殊的运输问题。
- 三维运输问题经常用在公共分配系统并经常考虑一个以上的目标。

模型描述

- 假设存在 m 个起点和 n 个终点、 K 种运输工具，两个目标的双准则三维运输问题 **BSTP** 可描述为：

$$\min \quad z^q = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{K} c_{ijk}^q x_{ijk}; \quad q = 1,2$$

$$\text{s.t.} \quad \sum_{j=1}^{n}\sum_{k=1}^{K} x_{ijk} = a_i; \quad i = 1,2,\cdots,m$$

$$\sum_{i=1}^{m}\sum_{k=1}^{K} x_{ijk} = b_j; \quad j = 1,2,\cdots,n$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ijk} = e_k; \quad k = 1,2,\cdots,K$$

$$x_{ijk} \geqslant 0; \quad \forall\, i,j,k$$

（平衡条件）

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j = \sum_{k=1}^{K} e_k$$

平衡条件可作为问题存在可行解的充分必要条件。

其中 x_{ijk}：*用第 k 种工具，从起点 i 到终点 j 的运输量；*

c^q_{ijk}：*用第 k 种工具，从起点 i 到终点 j 的，第 q 个目标的系数（单位运输费用、产品损失等）；*

a_i：*起点 i 的供给量；*

b_j：*终点 j 的需求量。*

e_k：*可用第 k 种工具运输的物品的重量*

GA求解

- 染色体编码
  - 对于三维运输问题，一个染色体可以用一个具有 $m\times n\times K$ 个元素的三维矩阵描述，$X=[x_{ijk}]$。
  - 其中 x_{ijk} 是染色体中相应的决策变量。
- 种群初始化、交叉、变异
  - 可将二维线性运输问题的矩阵表示法，在三维空间中进行扩展，详略。
- 适值函数的计算
  - 采用双准则线性运输问题的计算方法

固定费用运输问题

(Fixed Charge Transportation Problem)

固定费用运输问题

- 固定费用运输问题 (fixed-charged transportation problem) (fcTP) 是运输问题的扩展。
- 许多实际运输和分配问题，可以用公式表示为固定费用运输问题。
  - 例如具有固定物流费用的最小费用网络流问题，
  - 又如运输问题中，在给定的工厂和仓库之间每次运输存在一个固定成本的问题。

模型描述

- ***m*** 家工厂 ***n*** 个仓库的固定费用的运输问题可描述为：

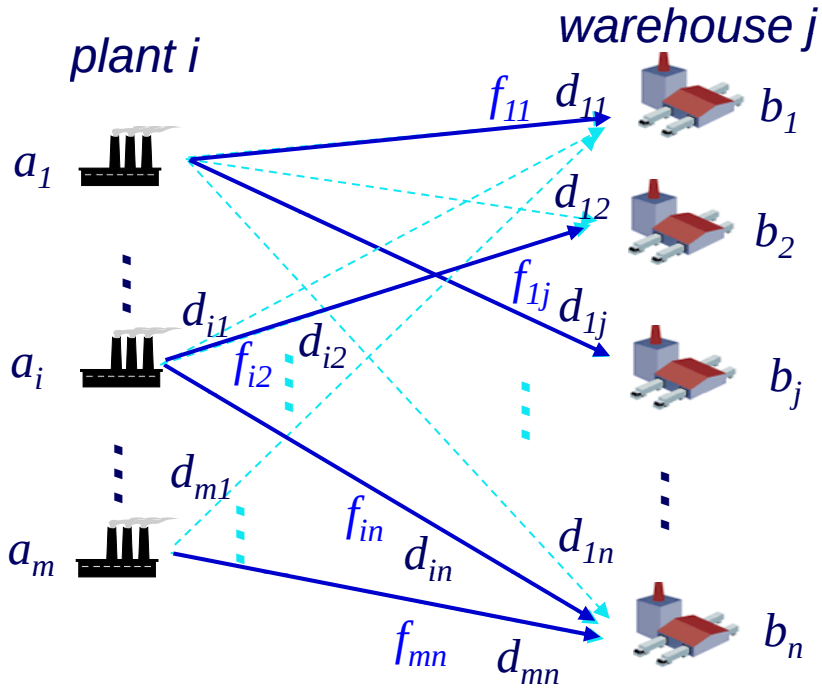
$$\min \quad f(x)=\sum_{i=1}^{m}\sum_{j=1}^{n}\left[f_{ij}(x)+d_{ij}g_{ij}(x)\right]$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ij} \le a_i, \qquad i=1,2,\ldots,m$$

$$\sum_{i=1}^{m} x_{ij} \ge b_j, \qquad j=1,2,\ldots,n$$

$$x_{ij} \ge 0, \qquad \forall i,j$$

$$g_{ij}(x)=\begin{cases}1, & \text{if } x_{ij}>0\\ 0, & \text{otherwise}\end{cases}$$



where,

x_{ij}：从工厂 ***i*** 到仓库 ***j*** 的运输量；

d_{ij}：从工厂 ***i*** 到仓库 ***j*** 的固定运输成本；

a_i：工厂 ***i*** 的供给量；

b_j：仓库 ***j*** 的需求量；

$f_{ij}(x)$：从工厂 ***i*** 到仓库 ***j*** 的运输费用。如果是线性的则有：$f_{ij}(x)=c_{ij}x_{ij}$

特点

- 目标函数不连续（凹的）
- **NP** 难题
- 经常被作为混合整数规划问题构造并求解
- 最优解必定出现在可行域的一个极点上
  - 是这类问题的一个重要属性。
  - 也就是说，它必定是与凸多边形的基本可行解相关。这意味着寻找一个全局最优解可以限制于仅需考虑一个凸多边形的极点。
  - 这个结果与线性规划的结果相似。
  - 尽管如此，因为目标函数是凹的，所以相关的算法更为复杂。
- 固定费用为零时，即为一般的运输问题

GA求解

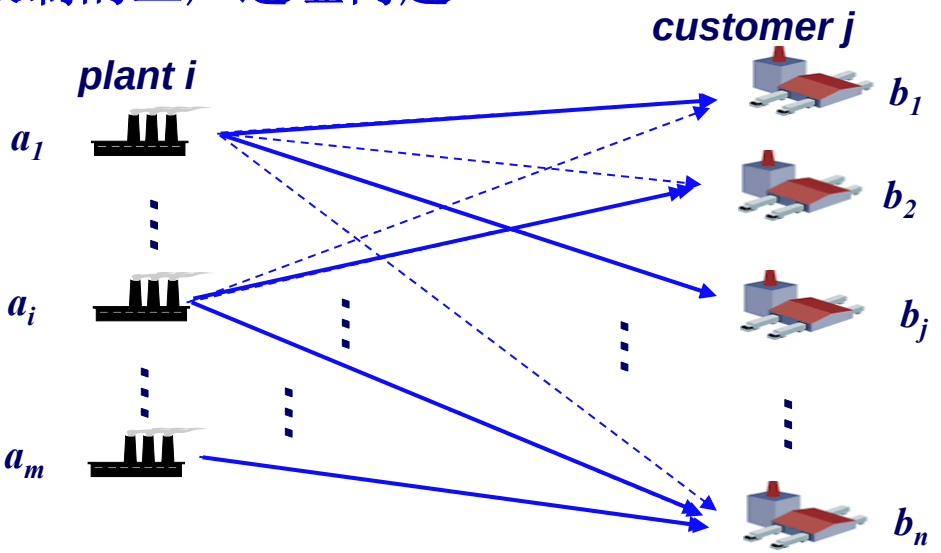
- 除目标函数不同于一般运输问题外，网络结构、约束构成等与一般运输问题的一样
- 所以，可以采用求解一般运输问题的遗传算法
  - 例如采用 **Prüfer** 数编码，单点交叉、反转和位移变异等。

容量限制的工厂选址问题

(Capacitated Plant Location Problem)

工厂选址问题

- 工厂选址问题实际上是固定费用模型的变型。
  - 假设有 n 个顾客，供给来自 m 家*潜在*（或备选）的工厂，以满足这些顾客的需求。通常 n 要比 m 更大。
  - 为了满足需求，有必要确定每家工厂的位置和生产能力。
  - 目标是最小化总成本，包括工厂建设的固定成本和工厂到顾客之间的运输成本。
  - 工厂选址问题可进一步分为：
    - 有容量限制的工厂选址问题
    - 无容量限制的工厂选址问题



容量限制的工厂选址问题

- **Capacitate Plant Location Problem (cPLP)**
  - 在工厂的生产能力有限的情况下，m 个潜在工厂 为 n 位顾客生产单一的产品，每位顾客需求量为 b_j 个单位。
  - 如果某个工厂 i 是开放的（或被建立)，则它存在固定成本 $d_i \geq 0$ 和相关的生产能力 $a_i > 0$。从工厂 i 到顾客 j 也有一个正的单位运输成本 c_{ij}。
  - 问题是如何确定工厂的位置，使得成本最小，并且容量不能超过、需求必须满足。

模型描述

- **cPLP可以构造为如下的混合整数规划:**

$$\min \quad z(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} d_i y_i$$

$$\text{s.t.} \quad \sum_{i=1}^{m} x_{ij} = b_j, \qquad j = 1,2,\cdots,n$$

$$\sum_{j=1}^{n} x_{ij} \leqslant a_i y_i, \qquad i = 1,2,\cdots,m$$

$$x_{ij} \geqslant 0, \qquad \forall\, i,j$$

$$y_i = 0 \text{ 或 } 1, \qquad i = 1,2,\cdots,m$$

固定费用运输模型

$$\min \quad f(x) = \sum_{i=1}^{m}\sum_{j=1}^{n}\left[f_{ij}(x) + d_{ij}g_{ij}(x)\right]$$

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ij} \leq a_i, \qquad i = 1,2,...,m$$

$$\sum_{i=1}^{m} x_{ij} \geq b_j, \qquad j = 1,2,...,n$$

$$x_{ij} \geq 0, \qquad \forall i, j$$

$$\text{with} \quad g_{ij}(x) = \begin{cases} 1, & \text{if } x_{ij} > 0 \\ 0, & \text{otherwise} \end{cases}$$

where,

- x_{ij} : *从工厂 i 到顾客 j 的运输量;*
- y_i :*工厂是开放的(或定址)(=1)或是关闭的(=0)*
- d_i : *工厂 i 开放的固定成本;*
- a_i : *工厂 i 的供给量（有限制）;*
- b_j : *顾客 j 的需求量;*
- c_{ij}: *从工厂 i 到顾客 j 的单位运输费用。*

无容量限制的工厂选址问题

- 当每个工厂能满足所有顾客需求时（$a_i \geq \sum b_j$, $i=1,\cdots,m$），容量限制可以不再考虑。问题转化为无容量限制的工厂选址问题 **(uPLP)**
- 定义 g_{ij} 表示工厂 i 满足顾客 j 的需求的份额(%)， $g_{ij}=x_{ij}/b_j$
- x_{ij} 用 $g_{ij}b_j$ 代替，令：$t_{ij}=c_{ij}b_j$，则模型可描述如下：

$$\text{uPLP:}\quad \min \quad z(x)=\sum_{i=1}^{m}\sum_{j=1}^{n}t_{ij}g_{ij}+\sum_{i=1}^{m}d_iy_j$$

$$\text{s.t.}\quad \sum_{i=1}^{m}g_{ij}=1, \qquad j=1,2,\cdots,n$$

$$\sum_{j=1}^{n}g_{ij}\leqslant ny_i, \qquad i=1,2,\cdots,m$$

$$g_{ij}\geqslant 0, \qquad \forall\, i,j$$

$$y_i=0\text{ 或 }1, \qquad i=1,2,\cdots,m$$

特点

- 目标函数不连续（凹的）
- **NP** 难题
- 混合整数规划
- 最优解必定出现在可行域的一个极点上

GA求解

- 除了不同的评价函数外，针对容量限制的工厂选址问题，可以采用：
  - 基于生成树的遗传算法

End