选址—分配问题
(Location Allocation Problem)

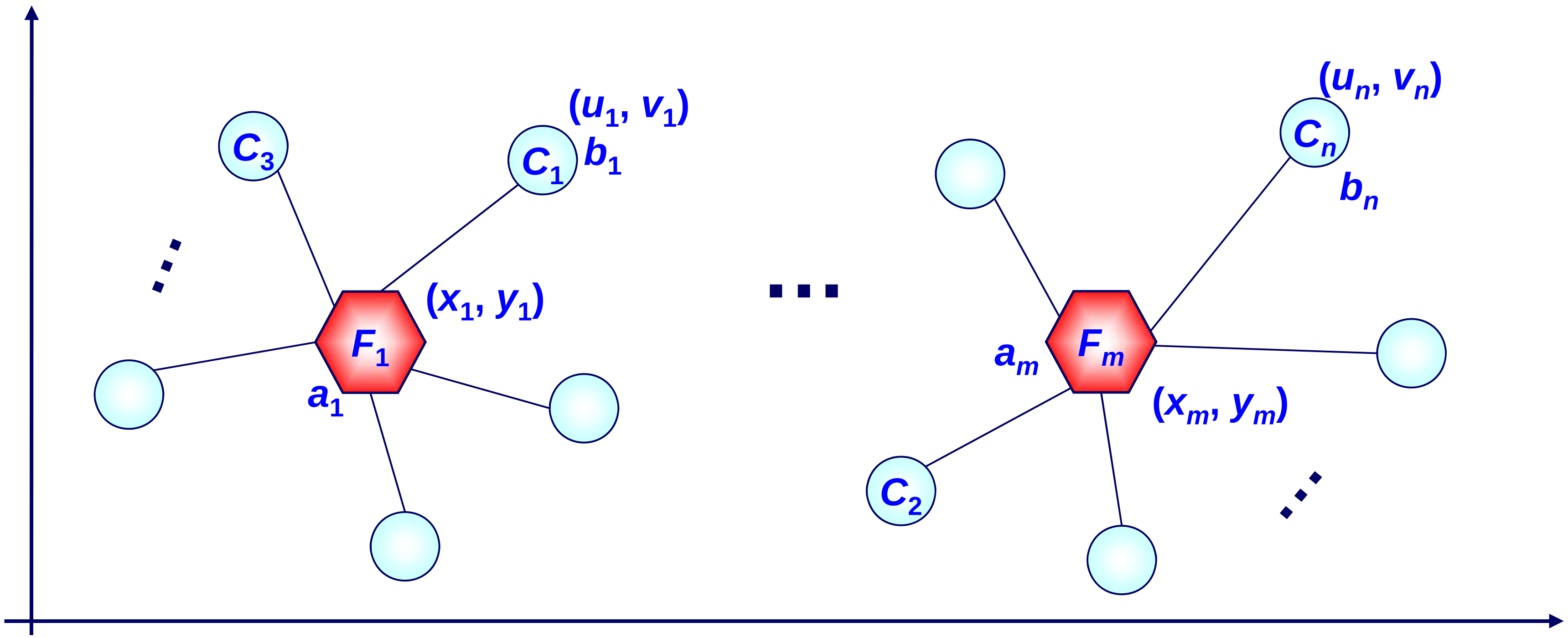
东北大学 系统工程研究所

2011.09

选址－分配问题

- 选址－分配 (location-allocation) 问题
  - 也称作多韦伯 (multi-Weber) 问题或 P 中位 (P-median) 问题。
- 单韦伯 (single Weber)问题
  - 在欧几里德空间上典型的单韦伯(single Weber) 问题是寻找一个位置，使从代表顾客位置的一些固定点到它的距离和最小。
- 问题描述：
  - 有 m 个“设施”需要选址，n 个已知位置的“顾客”分配给不同的设施，每个顾客的需求为a_j，j=1,2,…,n；每个设施具有的能力为b_i，i=1,2,…,m
  - 我们需要找到
    - 设施的位置（选址）
    - 顾客对设施的分配
  - 使顾客和服务他们的设施间的距离总和最小。

图形描述



m ：设施总数

n ：顾客总数

F_i: 第i 个设施, i=1,2,…,m

C_j: 第 j 个顾客, j=1,2,…,n

a_i: 第 i 个设施的能力

b_j: 第 j 个顾客的需求

$F_i=(x_i, y_i)$: 设备 i 的未知位置，决策变量

$C_j=(u_j, v_j)$: 顾客 j 的已知位置

数学模型

$$\min \quad f(F,z)=\sum_{i=1}^{m}\sum_{j=1}^{n} t(F_i,C_j)\cdot z_{ij}$$

$$\text{s.t.} \quad g_i(z)=\sum_{j=1}^{n} b_j z_{ij} \le a_i,\ i=1,2,\cdots,m$$

保证不超过每个设施的服务能力

$$g_{m+j}(z)=\sum_{i=1}^{m} z_{ij}=1,\quad j=1,2,\cdots,n$$

保证每个顾客只由一个设施服务

$$z_{ij}=0 \text{ or } 1,\quad i=1,2,\cdots,m,\ \ j=1,2,\cdots,n$$

- 变量:

  z_{ij}: 0-1 决策变量

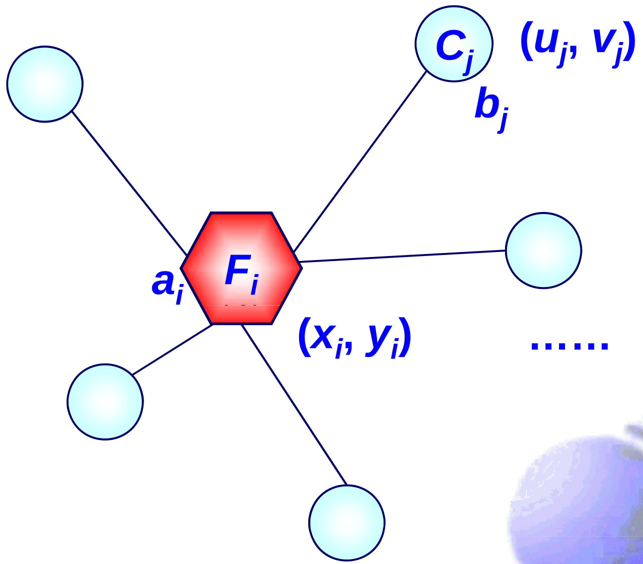
  z_{ij}=1, 顾客 ***j*** 由设施 ***i*** 服务；否则 **z_{ij}=0**

  F_i = (x_i, y_i) :设施 ***i*** 的未知位置，决策变量

- 参数:

  $t(F_i,C_j)$: 由设施 ***i*** 到顾客 ***j*** 的欧几里得距离。

$$t(F_i,C_j)=\sqrt{(x_i-u_j)^2+(y_i-v_j)^2}$$



特点

- 非线性规划问题
- 既有 **0-1** 变量，又有实数变量
- 分配子问题是一个一般的指派问题
- **NP-**难的问题
- **Cooper** 是正式认识并描述多韦伯问题的第一位学者。他证明目标函数既不是凹的也不是凸的，并存在许多局部最优解。

分类

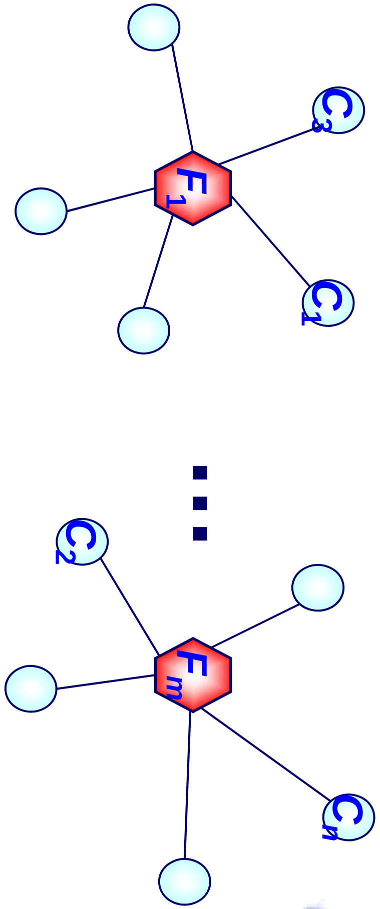
- 按能力分类：
  - 有能力约束的LAP
  - 无能力约束的LAP
- 按阶段分类：
  - 单阶段LAP
  - 多阶段LAP
- 其它分类：
  - 有障碍的LAP
  - 平衡的LAP

问题的扩展

- 一般的选址-分配模型致力于为这样一些基础设施寻找最佳选址，如：
  - 学校、消防站、公园等，
  - 此类基础设施对于“最佳选址”理解的共同点在于，使供需点之间的“总距离”或者“平均距离”最小。
- 除此以外，这个基本模型经过扩展还可以有更广泛的应用范围，如：
  - 优化城市零售商业网点空间分布(通过最大化惠顾人流量来实现)、
  - 优化制造业场所空间分布(通过最小化运输成本实现)、
  - 优化公共服务设施空间分布(通过最优化服务质量实现)、
  - ……

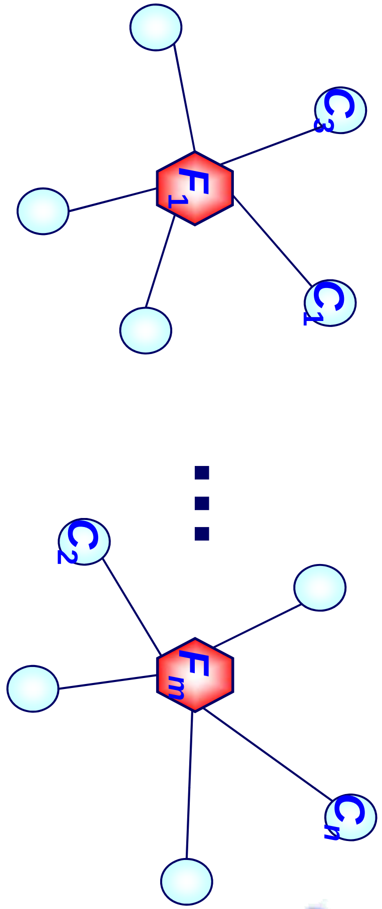
应用

- 物流配送中心选址问题
  - 物流网络设计中首先必须要解决的问题
  - 物流中心→设施；零售商→顾客；运费或距离最短
- 城市医院空间布局优化
  - 医院→设施；街区→顾客；服务质量, 距离最小等
- 城市邮政局所空间布局
  - 邮局→设施；建筑物的中心→顾客；邮政服务的平均出行距离最短
- 城市电网变电站选址问题
  - 变、配电站→设施；小区→顾客；输电线（距离）最短
- 海上溢油应急点选址优化
  - 应急服务点→设施；发生溢油较高的区域→顾客；到达发生事故点的距离之和最小
- 公路养护资源的选址和配置问题
  - 养护点→设施；路段→顾客；目标是工作量均衡



应用

- 社会考试考场选择问题
  - 例如英语等级考试、公务员考试等涉及的人数很多；
  - 考场→设施；生源地→顾客；距离最短、最便利等
- 零售商业点的选址问题
  - 商业点→设施；小区（街区）→顾客；惠顾人流量最大或距离最短
- 公共设施的最优选址问题
  - 如公园、学校、图书馆、加油站、污水处理厂、城市垃圾填埋场、消防设施等
- 突发事件的应急资源配置问题
  - 如地震、海啸、流行病、
- 设备选址
  - 如电力、通信、交通系统中相关设备或站址的选取和定位



一般求解方法

- **Cooper** 提出了一种称作选择选址－分配(**alternative location-allocation**) 的启发式，这是一种最好的启发方法
- 随着非线性规划技术的发展，通过松弛整数分配约束，同时考虑选址变量和分配变量，产生了一些新的方法。
  - **Murtagh** 和 **Niwattisyawong** 提出一种松弛 **0-1**分配约束的方法。允许在 **[0,1]** 区间上取值，然后用非线性规划软件包来解决选址和分配问题。
  - **Bongartz** 等提出的方法，也松弛 **0-1** 分配约束，同时还采用活动集合方法，并利用了选址－分配问题固有的特殊结构。
  - 这些方法可用于大规模的实际问题。但都不能保证全局和近全局最优。
- **Harris** 等发现当可能解的数量是一个类型 **II** 的 **Sterling** 数时，可行解的数量相当少。他们发现任何可行解中顾客的子集必须包括在非重叠凸域内，他们提出了一种算法，利用问题这一凸域特性产生所有可行解，并通过对这些解的完全枚举找到最优解。
- **Qstresh** 提出了基于通行线 (**passing line**) 解决两设施韦伯问题的不同算法。
- **Rosing** 基于 **Harris** 的思想提出了关于一般多韦伯问题的优化方法，然而，只能用于小规模问题。

GA求解

- **Gong, Gen, Xu 和 Yamazaki** 提出了一种解决具有能力约束的选址－分配问题的混和进化方法
- 这种方法将问题分为两层:
  - 分配层
    - 在分配层，因为这一子问题需经常解决，所以采用拉格朗日松弛法解决这个一般指派问题，以尽快获得解。
  - 选址层
    - 在选址层，采用进化方法搜索整个选址区域。通过这种方法，整个可选址区域可以被有效地搜索以找到全局或近全局最优解。

- 染色体表达
  - 因为选址变量是连续的，所以采用浮点值染色体。染色体表达如下：

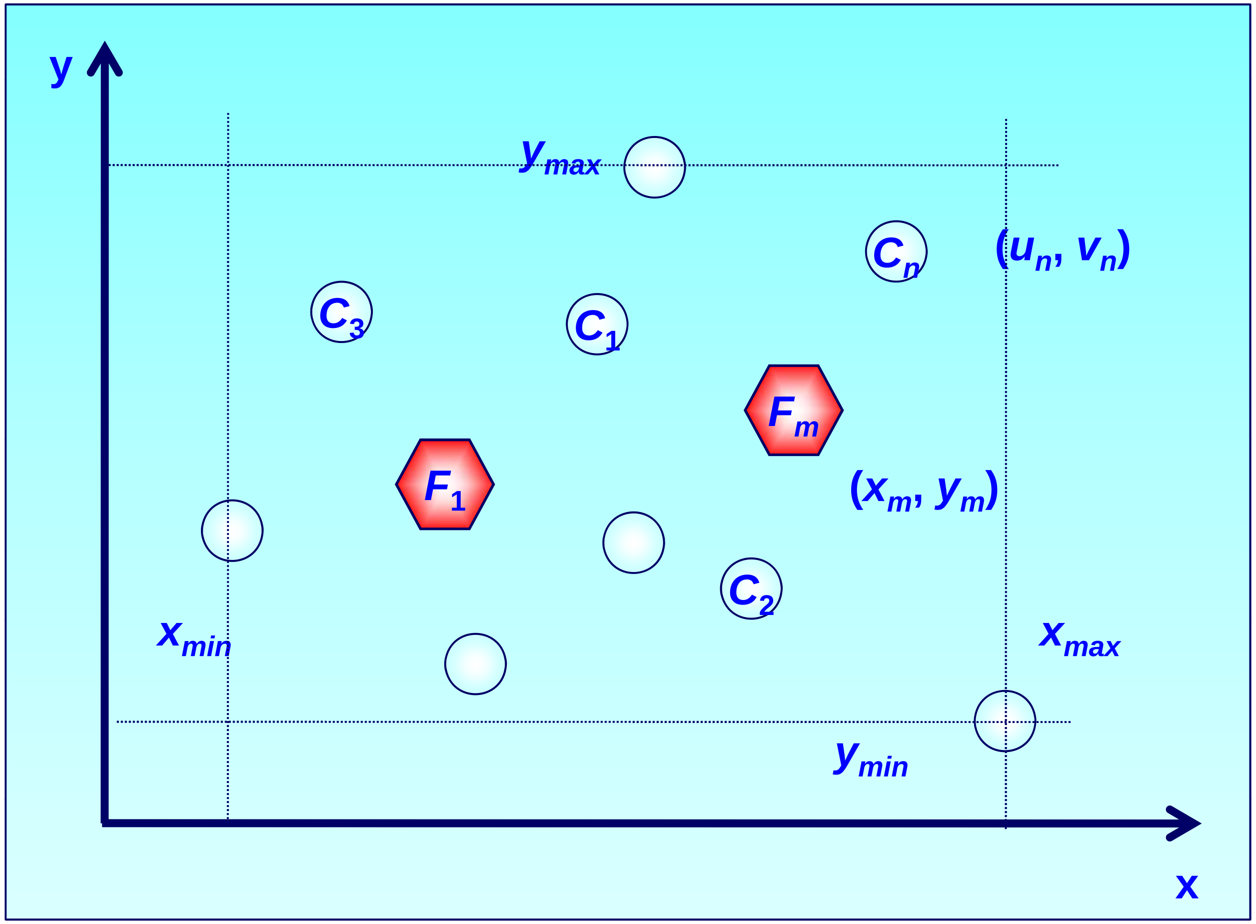
$$v_k = [(x_1^{\,k}, y_1^{\,k}), (x_2^{\,k}, y_2^{\,k}), \cdots, (x_i^{\,k}, y_i^{\,k}), \cdots, (x_m^{\,k}, y_m^{\,k})]$$

  - 例如：

$$v_1 = [(x_1^{\,1}, y_1^{\,1}), (x_2^{\,1}, y_2^{\,1}), (x_3^{\,1}, y_3^{\,1})] \;=\; [(1.3, 3.8), (2.5, 0.4), (3.1, 1.7)]$$

GA求解

- 初始化
  - 最优选址应在包括所有顾客的矩形区域内，因此设施初始选址可以从这个矩形区域内任取位置。



GA求解

- 交叉
  - 对于实数编码，可以采用线性凸组合交叉 (**Linear Convex Combination**)
    - 假设选择具有如下染色体的两个双亲产生后代

$$v_1=[(x_1^{\ 1},y_1^{\ 1}),(x_2^{\ 1},y_2^{\ 1}),\cdots,(x_i^{\ 1},y_i^{\ 1}),\cdots,(x_m^{\ 1},y_m^{\ 1})]$$

$$v_2=[(x_1^{\ 2},y_1^{\ 2}),(x_2^{\ 2},y_2^{\ 2}),\cdots,(x_i^{\ 2},y_i^{\ 2}),\cdots,(x_m^{\ 2},y_m^{\ 2})]$$

    - 后代染色体中的基因由下面等式确定

$$\bar{x}_i^{\ 1}=\alpha_i\cdot x_i^1+(1-\alpha_i)\cdot x_i^2$$

$$\bar{y}_i^{\ 1}=\alpha_i\cdot y_i^1+(1-\alpha_i)\cdot y_i^2$$

$$\bar{x}_i^{\ 2}=(1-\alpha_i)\cdot x_i^1+\alpha_i\cdot x_i^2$$

$$\bar{y}_i^{\ 2}=(1-\alpha_i)\cdot y_i^1+\alpha_i\cdot y_i^2$$

    - 其中， α_i 是 **(0, 1)**内的独立随机数 (i=1,2,⋯,m)。

GA求解

- 变异
  - 采用两类变异算子在进化过程中交替进行
    - 精细变异(subtle mutation)：双亲染色体进行较小的随机改变，产生一个新的后代染色体；
    - 强烈变异(violent mutation)：采用与初始化过程相同的方法产生后代

$$\bar{v}=\begin{cases} v+\delta & \text{Subtle mutation} \\ \Theta & \text{Violent mutation} \end{cases}$$

$\delta=[\delta_1,\delta_2,\ldots,\delta_{2m-1},\delta_{2m}]$,

δ_i 是 $(-\alpha,\alpha)(i=1,2,\ldots,2m)$之间的随机数

α 是一个较小的正数

$\Theta=[(\theta_1^x,\theta_1^y),(\theta_2^x,\theta_2^y),\ldots,(\theta_m^x,\theta_m^y)]$

θ_i^x 是 $[x_{\min},x_{\max}]$ 之间的随机数

θ_i^y 是 $[y_{\min},y_{\max}]$ 之间的随机数

GA求解

- 评价
  - 对于指定的染色体 v_k，设施的位置是固定的
  - 我们可将最优分配的距离和映射为 v_k 的适值
  - 为计算距离和，需要解决分配子问题。
    - 对于固定的设施选址问题，采用拉格朗日松弛法解决分配子问题
    - 调整松弛解使之可行 (考虑能力约束)
  - 指定染色体 v_k 的适值函数如下:

$$eval(v_k) = \frac{1}{f(F, z)}$$

GA求解

- 相对禁止的选择策略
  - (μ+λ) 选择策略，在双亲和他们的后代中选出较好的个体形成下一代。
  - 然而，这一策略通常导致进化过程退化，陷入局部最优。为了避免退化，提出了一种 称作相对禁止(**relative prohibition**)的选择策略。
    - 选择最好的个体作为下一代的第一个成员
    - 在剩余的个体中选择最好的个体
    - 如果它属于已选择的个体的禁止邻域，则忽略；否则作为下一代成员。
    - 重复上述过程直至所有成员都被选择。
    - 如果未满足种群规模要求，则采用随机选择进行补充。
  - 禁止邻域的定义：
    - 适值意义上的邻域

$$eval(v_k) - eval(v) \le \alpha \quad \alpha - \text{较小的正参数}$$

    - 选址意义上的邻域

$$\|v_k - v\| \le \gamma \quad \gamma - \text{较小的正参数}$$

其中 v_k已被选入下一代，v 尚未选入下一代的染色体。

无能力约束的选址—分配问题

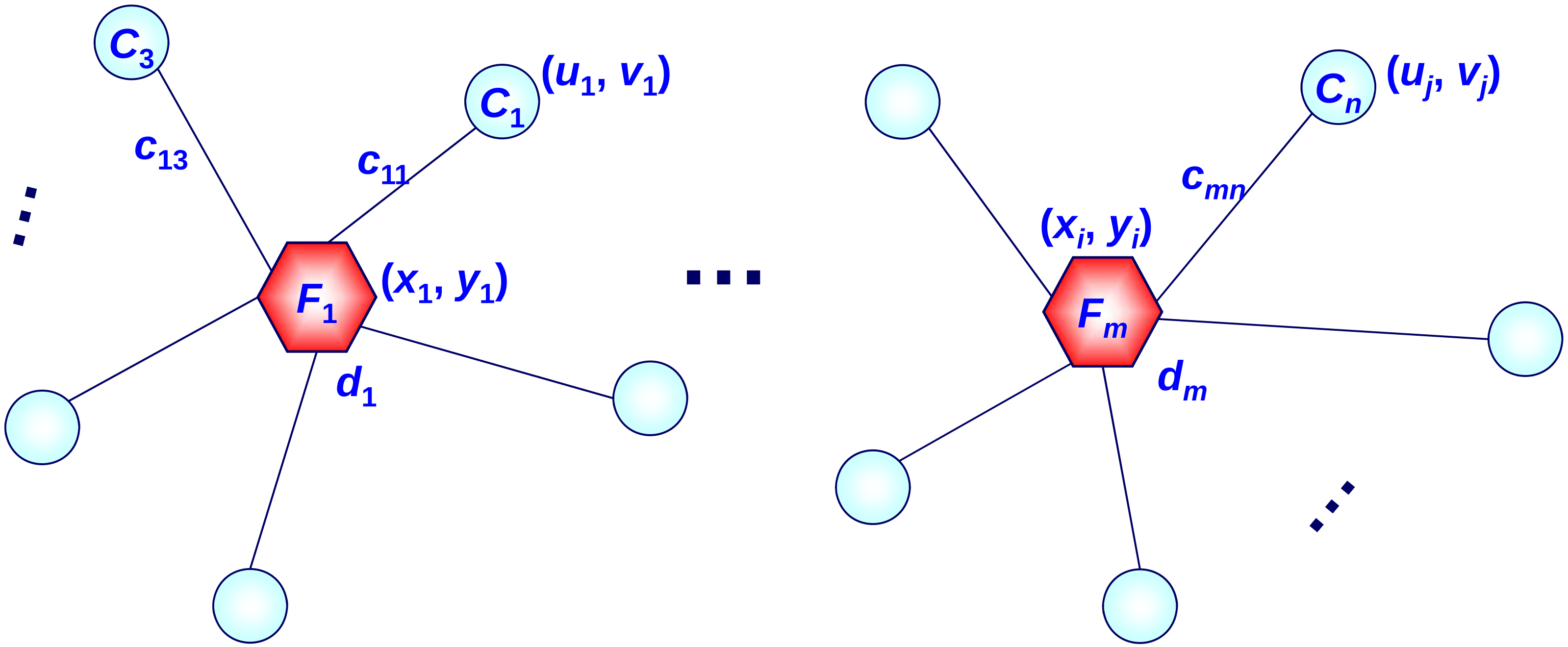
(Uncapacitated Location Allocation Problem)

无能力约束的选址－分配问题

- 如果设施在能力上没有限制，则问题被称为无能力约束的选址一分配问题（**uLAP**）
- 在 **uLAP** 问题当中，一些设施在 $\boldsymbol{n}$ 个可能的客户之间进行分配，目标是以最小的费用在 $\boldsymbol{m}$ 个*给定位置*建立设施，以满足所有客户的需求。
- 费用由建立设施的固定费用和完成需求的运输（或供应)费用组成。

图形描述

- 设施的选址和客户的分配



m ：设施总数

n ：客户总数

F_i: 第i 个设施, i=1,2,…,m

C_j: 第 j 个客户, j=1,2,…,n

d_i: 建立第 i 个设施的费用

c_{ij}: 第 i 个设施向第 j 个客户提供服务的费用

$F_i=(x_i, y_i)$: 设备 i 的已知位置

$C_j=(u_j, u_j)$: 客户 j 的已知位置

数学模型

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

$$\text{s.t.} \quad g(x)=\sum_{i=1}^{n}x_{ij}=1 \qquad \forall j\in J \qquad (2)$$

保证每个顾客只由一个设施服务

$$x_{ij}\le y_i \qquad \forall i\in I,\ j\in J \qquad (3)$$

如果设施不被建立，则不提供服务

$$x_{ij},\ y_i\in\{0,\ 1\} \qquad \forall i\in I,\ j\in J \qquad (4)$$

- **变量:**

  x_{ij}: **0-1 决策变量**

  x_{ij}=1, 顾客 *j* 由设施 *i* 服务；否则 x_{ij}=0

  y_i: **0-1 决策变量**

  y_i=1,设施 *i* 被建立，否则 y_i=0

GA求解

- 基本思路：
  - 采用子种群的概念增加染色体的多样性
  - 每一代中的染色体被分为2个子种群
  - 在每个子种群中，采用不同种类的遗传算子。

- 染色体表达
  - **Syarif, A., Y.S. Yun, & M. Gen** 提出一种表达方法。
  - 双表达模式，分为两个“**Phase**”：
    - 基于设施的表达：**0-1** 整形变量用于表达设施的关闭 **(=0)** 和开放 **(=1)**；
    - 基于客户的表达：**1-*m*** 之间的随机数。
  - 例如：一个 **5** 个设备 **10** 个客户的染色体表达如下：

facility id	1	2	3	4	5	1	2	3	4	5	6	7	8	9	10	customer id
bit string	1	0	1	0	1	4	5	5	3	4	3	2	3	2	5	encode value

phase 1 Location　　phase 2 Allocation

GA求解

- 交叉
  - Phase1采用 1 点交叉；Phase2采用 2 点交叉。

	phase 1					phase 2									
	1	2	3	4	5	1	2	3	4	5	6	7	8	9	10
v_1	1	0	1	0	1	4	5	5	3	4	3	1	3	2	5
v_2	1	1	0	1	0	3	4	2	3	3	4	5	3	1	1
v_1'	1	0	1	1	0	4	5	5	3	3	4	5	3	2	5
v_2'	1	1	0	0	1	3	4	2	3	4	3	1	3	1	1
	one-cut point					two-point									

- 变异
  - Phase1采用反转变异；Phase2采用位移变异。

	phase 1					phase 2									
	1	2	3	4	5	1	2	3	4	5	6	7	8	9	10
Parent	1	0	1	0	1	4	5	5	3	4	3	1	3	2	5
Offspring	1	1	0	1	1	4	3	4	3	1	5	5	3	2	5
	inversion					displacement									

GA求解

- 评价
  - 以目标函数为适值
  - 在染色体解码的过程当中计算适值。

input: chromosomes, c_{ij}, d_i
output: total cost
step1: Determine the opened/closed facilities.
step2: Each customer is assigned to the opened facility that gives the least cost.
step3: Repeat the procedure for all individuals.

- 选择
  - 精英选择方法
    - 用于选择第一个子种群的下一代
  - 轮盘赌选择方法
    - 用于选择第二个子种群的下一代

GA求解

- 求解流程

```
begin
        t ← 0;
        initialize P(t) by facilities/customer-based encoding;
        fitness eval(P) by facilities/customer-based decoding;
        while (not termination condition) do
                crossover P(t) to yield C(t) by one-cut point/two-cut point crossover;
                mutation P(t) to yield C(t), by inversion/displacement mutation;
                fitness eval(C) by facilities/customer-based decoding;
                select P(t+1) from P(t) and C(t) by elitist selection and roulette wheel method;
                t ← t+1;
        end
      output best facility locations and customer allocations;
end
```

- 值得借鉴的地方：
  - 分为多个子种群，不同的种群采用不同的遗传操作，以提高种群的多样性
  - 针对问题不同的方面，设计不同编码，采用不同的算子

障碍选址—分配问题

(Obstacle Location Allocation Problem)

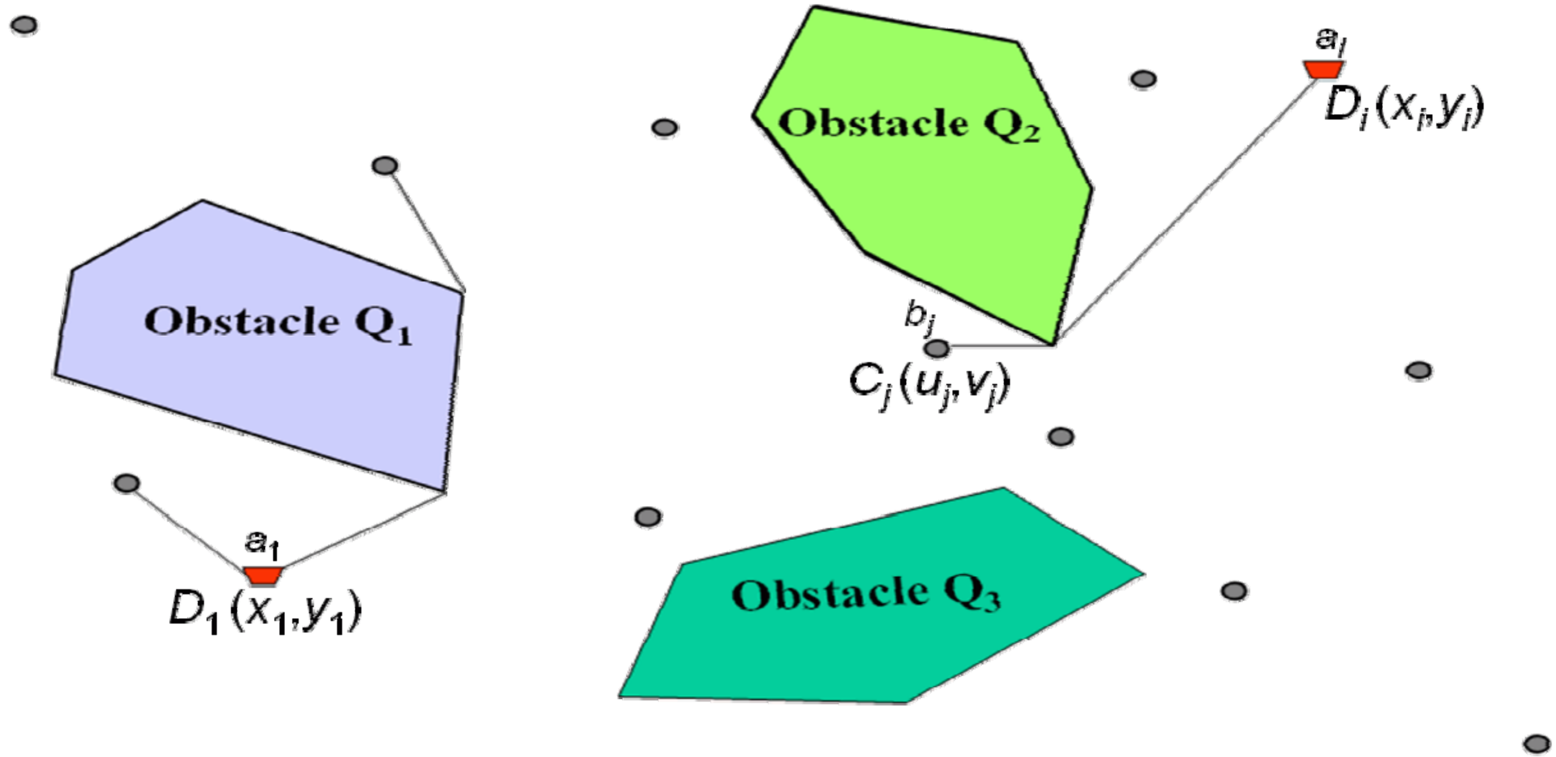
障碍选址－分配问题

- 选址－分配模型是在没有考虑障碍的情况下对设施选址进行设计的理想情况。
- 实际问题中，通常需要考虑障碍或禁止区域约束。
  - 例如湖泊、公园、建筑、河流等对设施选址设计形成的障碍。
- 障碍有两类:
  - 禁止选址;
    - 新设施不能布置在障碍区域内，这一问题也称作可憎的选址 **(obnoxious location)**。
  - 禁止连接路径。
    - 顾客和设施的连接路径不能通过障碍区域，这一问题也称作移动障碍选址。
- 当考虑障碍约束时，选址－分配问题的求解变得更为复杂和困难。
- 由于障碍选址－分配问题应用广泛，许多学者都集中于这一问题的研究。

问题描述

- 设有 n 个位置已知的顾客和 m 个供应中心，供应中心对所有顾客提供某种服务，例如，提供物料或能源，此外还存在 q 个代表禁止区域的障碍。
- 为进行数学描述，做如下假设：
  - 顾客 j 的服务需求为 a_j，$j=1,2,\cdots,n$
  - 供应中心 i 的服务能力 b_i，$i=1,2,\cdots,m$
  - 每个顾客只由一个设施服务
  - 障碍可描述为凸多边形
  - 供应中心不能建在任何障碍区内
  - 供应中心和顾客间的连接路径不允许穿过任何障碍
- 问题是：
  - 要选择供应中心的最好选址，使顾客与他们的供应中心间的距离总和为最小。

图形描述



m ：设施总数

n ：客户总数

k ：障碍物总数

D_i: 第 *i* 个供应中心, $i=1,2,\dots,m$

C_j: 第 *j* 个客户, $j=1,2,\dots,n$

Q_k: 第 *k* 个障碍物, $k=1,2,\dots,q$

a_i: 第 *i* 个供应中心的能力

b_j: 第 *j* 个客户的需求

$D_i=(x_i, y_i)$:供应中心*i* 的未知位置，决策变量

$C_j=(u_j, u_j)$: 客户 *j* 的已知位置

数学模型

$$\min \quad f(D,z)=\sum_{i=1}^{m}\sum_{j=1}^{n} t(D_i,C_j)\cdot z_{ij} \qquad (1)$$

$$\text{s.t.} \quad g_i(z)=\sum_{j=1}^{n} b_j z_{ij} \le a_i, \quad i=1,2,\cdots,m \qquad (2)$$

保证不超过每个设施的服务能力

$$g_{m+j}(z)=\sum_{i=1}^{m} z_{ij}=1, \quad j=1,2,\cdots,n \qquad (3)$$

保证每个顾客只由一个设施服务

$$D_i=(x_i,y_i)\notin Q_k \qquad i=1,2,\cdots,m, k=1,2,\cdots,q \qquad (4)$$

设施不落入障碍物内

$$(x_i,y_i)\in R_T\ , \qquad i=1,2,\cdots,m \qquad (5)$$

在选址和分配区域内

$$x_i, y_i \in R \qquad i=1,2,\cdots,m \qquad (6)$$

$$z_{ij}=1 \text{ or } 0, \qquad i=1,2,\cdots,m, j=1,2,\cdots,n \qquad (7)$$

- 变量:
  - **z_{ij}: 0-1** 决策变量
    - **z_{ij}=1,** 顾客 ***j*** 由供应中心 ***i*** 服务；否则 **z_{ij}=0**
  - **D_i = (x_i, y_i) :**供应中心 ***i*** 的未知位置，决策变量，不应该落入障碍物内
- 参数：
  - **$t(D_i,C_j)$:**在避开障碍物的情况下，由供应中心 ***i*** 到顾客 ***j*** 的路径集合中的最短距离。
  - **R_T :**考虑选址和分配问题的区域

特点

- 非线性0-1混合整数规划问题
- 目标函数和约束非常复杂，有些不能用数学方程式直接表达。
- NP-难的问题

GA求解

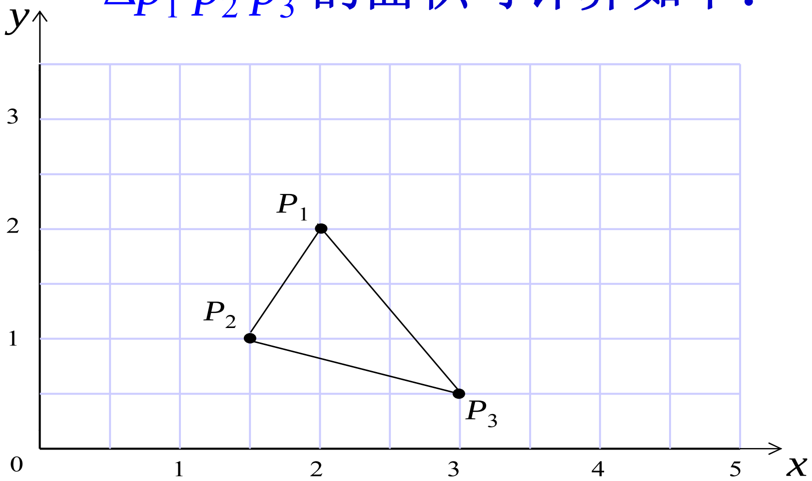
- 染色体表达
  - 问题具有两类决策变量
    - 一个是连续选址变量 (x_i^k, y_i^k)
    - 另一个是 **0-1** 分配变量 z_{ij}
  - 采用供应中心（**DC**）的坐标选址描述如下：

$$v_k = [(x_1^k, y_1^k), (x_2^k, y_2^k), \cdots, (x_i^k, y_i^k), \cdots, (x_m^k, y_m^k)]$$

- 初始化
  - 在包括所有顾客的矩形区域内，随机选择点，作为某些供应中心的初始选址。
- 检查可行性
  - 一个新的设施选址产生以后，应该知道选址是否是可行的；
  - 对不可行的染色体，一个可计算的有效修复算法，是非常重要同时也是必须的。

GA求解

- 检查可行性
  - 命题1：
    - 给定坐标平面上三个点 $p_1(x_1, y_1)$，$p_2(x_2, y_2)$，$p_3(x_3, y_3)$，则三角形 $\Delta p_1 p_2 p_3$ 的面积可计算如下：



$$S_{\Delta p_1 p_2 p_3} = \frac{1}{2}\left|\det\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}\right|$$

  - 命题2：
    - 令：$P_{k1}, P_{k2}, \ldots, P_{kp_k}$ 是多边形 p_k 的顶点，D_i 是一个被选址的点。
    - S_{Q_k} 是多边形 $P_{k1}, P_{k2}, \ldots, P_{kp_k}$ 的面积
    - $S_{\Delta D_i P_{kr} P_{k(r+1)}}$ 是由 D_i 和多边形的边组成的三角形 $\Delta D_i P_{kr} P_{k(r+1)}$ $(r = 1, 2, \cdots, p_k)$ 的面积，这里考虑 $P_{k(p_k+1)} \simeq P_{k1}$

GA求解

- 检查可行性
  - 点 D_i 在多边形 $P_{k1},P_{k2},\ldots,P_{kp_k}$ 之外，当且仅当

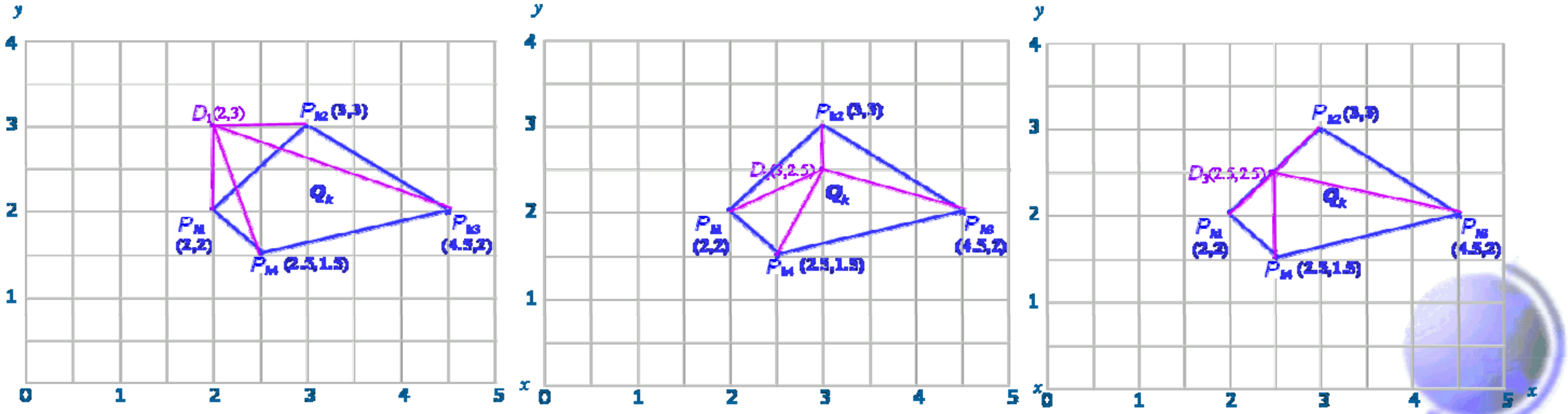
$$S_{Q_k} < \sum_{r=1}^{p_k} S_{\Delta D_i P_{kr} P_{k(r+1)}}$$

  - 点 D_i 严格在多边形之内，当且仅当

$$S_{Q_k} = \sum_{r=1}^{p_k} S_{\Delta D_i P_{kr} P_{k(r+1)}};\quad S_{\Delta D_i P_{kr} P_{k(r+1)}} > 0,\qquad r = 1,2,\ldots,p_k$$

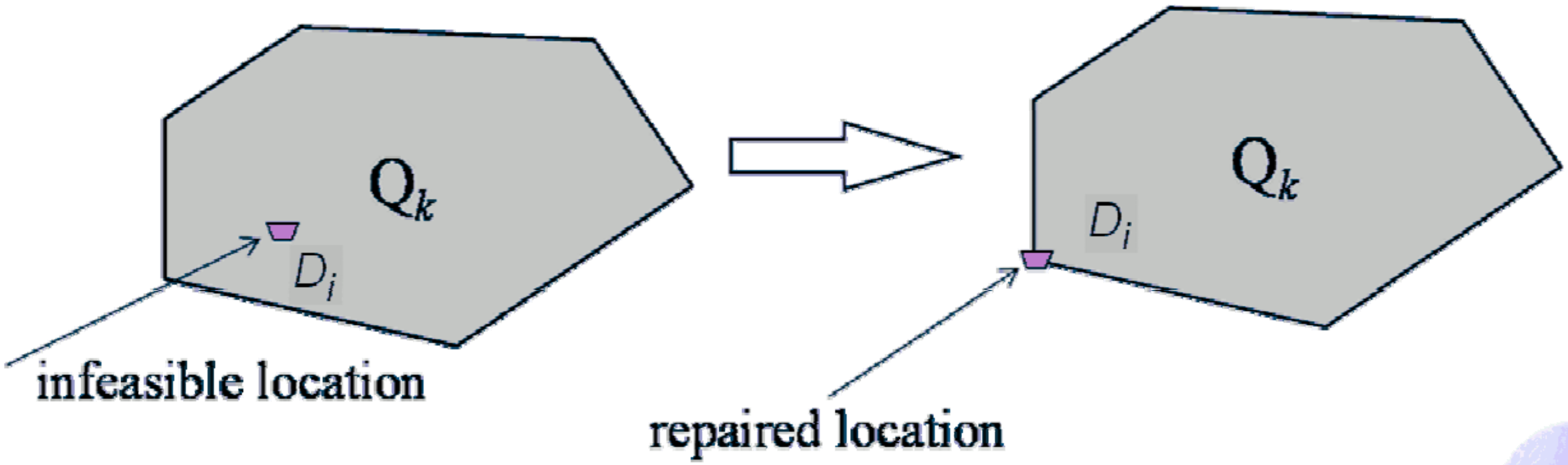
  - 点 D_i 在多边形的边上，当且仅当

$$S_{Q_k} = \sum_{r=1}^{p_k} S_{\Delta D_i P_{kr} P_{k(r+1)}};\ S_{\Delta D_i P_{kr_0} P_{k(r_0+1)}} = 0,\quad \exists r_0 \in \{1,2,\ldots,p_k\}$$



GA求解

- 修复不可行染色体
  - 如果一个 ***DC*** 的选址是不合理的，例如，落入到任意一个障碍物之内
  - 应该根据修复算法，用障碍物上离 ***DC*** 最近的顶点替换障碍物之内的不可行点。



GA求解

- 染色体的评价
  - 采用目标函数值，既用客户到供应中心的距离和最小，来确定染色体的适应值。

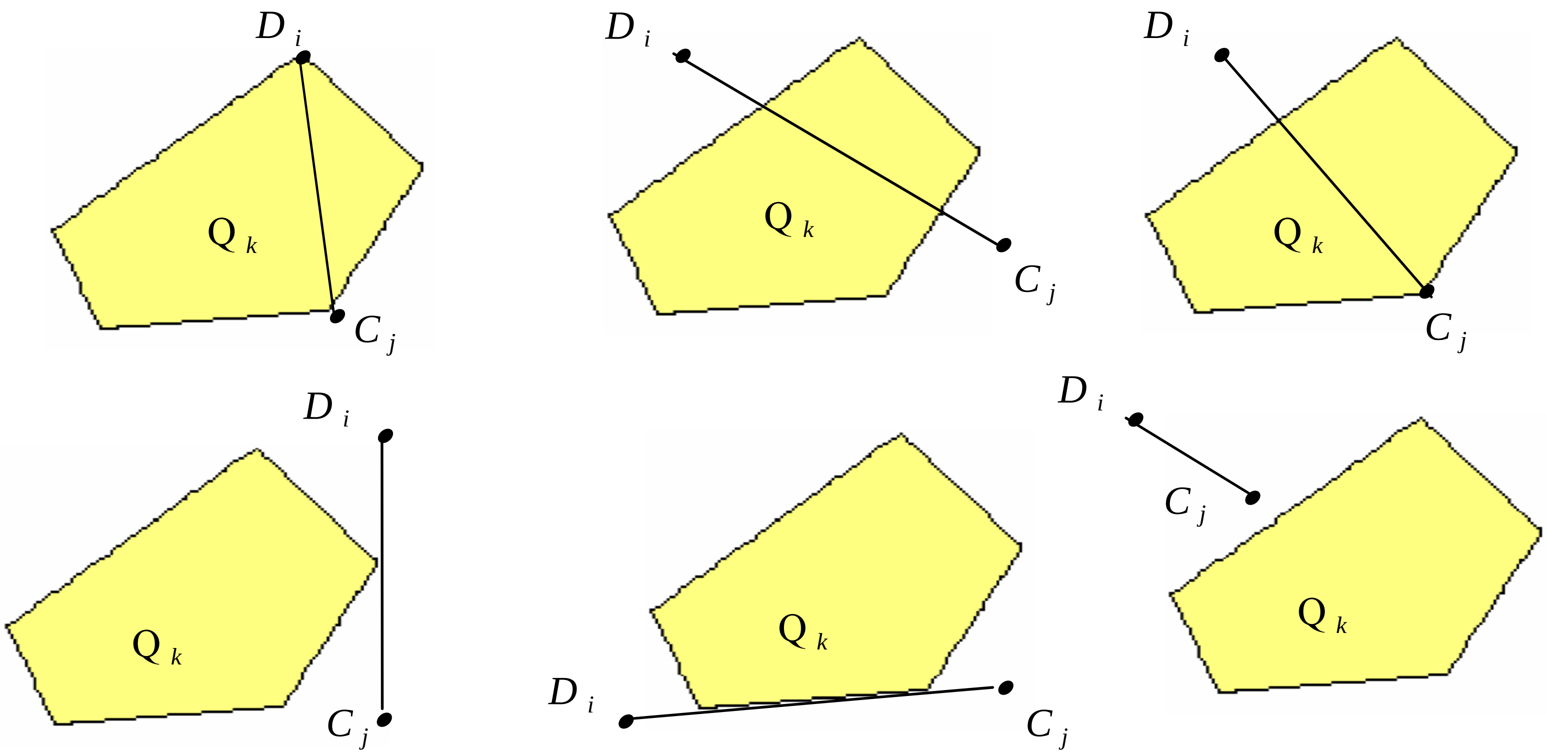
$$eval(v_k) = \frac{1}{f(F, z)}$$

  - 为了有效评价适值，在满足障碍约束的条件下，必须要有对分配问题和路径选择子问题的高效求解过程。
    - 首先确定每个客户到 **DC** 的分配、连接路径，并确定距离。
    - 检查路径的可行性（不跨越任何障碍物）
    - 如果路径不可行（跨越一个或多个障碍物），修改路径并更新距离。
- 分配
  - 一旦 **DC** 的选址已经确定并忽略障碍物，则分配问题可以采用拉格朗日松弛法有效地解决
  - 若存在过载的**DC**，将客户调整到其它的 **DC**，尽可能地使距离增加最小。

GA求解

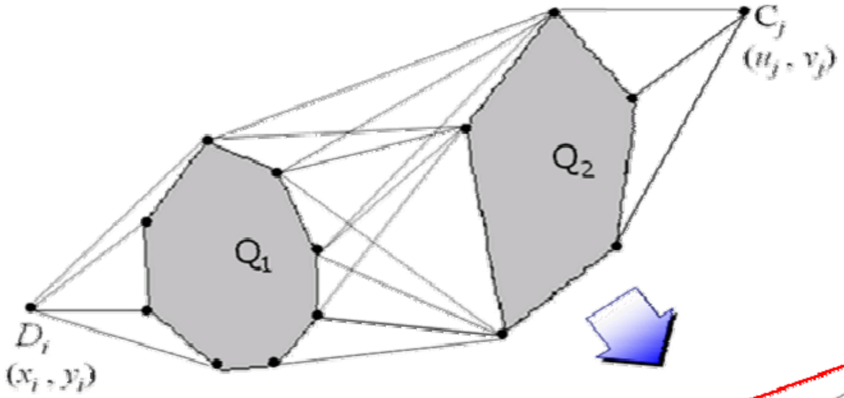
检查路径的可行性

- 一种避免冲突的方法是：检查连接路径的线段是否与定义多边形的线段交叉。



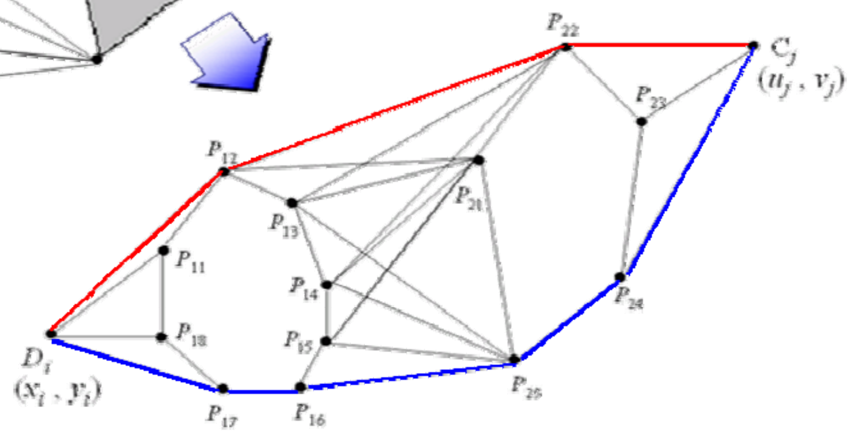
GA求解

- 寻找跨越障碍的最短路
  - 可视网络包括:
    - 与 **DC** 相应的选址点, 客户点, 以及障碍物的所有顶点所组成的节点
    - 所有不穿过障碍物的一对点所构成的线段（边）。
  - 最短路的获取
    - 采用 **Dijkstra** 最短路算法获得最短路



最短路: $D_i - P_{12} - P_{22} - C_j$

可行路: $D_i - P_{17} - P_{16} - P_{25} - P_{24} - C_j$



GA求解

- 交叉
  - 采用线性凸组合交叉
- 变异
  - 采用两类变异算子在进化过程中交替进行
    - 精细变异：双亲染色体进行较小的随机改变，产生一个新的后代染色体；
    - 强烈变异：采用与初始化过程相同的方法产生后代
- 相对禁止的选择策略

End