# 1 绪论

## 1.1 研究背景

运筹学(opearations research)是应用数学的一个分支，主要是运用统计学、数学模型和算法等知识，研究实际生活中的复杂问题，得到复杂问题的最优或近似最优解，进而达到整体最优或近似最优。概括的说，运筹学就是研究解决复杂问题的众多方案中，什么样的方案是最优方案以及怎么样找到最优方案。所以，运筹学也称为广义最优化理论。

我们通常认为运筹学起源于第二次世界大战，但实际上，在二战之前就有运筹学思想的萌芽了。1736年，Euler运用图论思想研究了哥尼斯堡七桥问题，这被认为是图论的起源；1738年，Bernoulli提出了效用这一概念，这是决策论的基础；1777年，Buffon提出了利用随机投针法来计算*π*的方法，这是蒙特卡洛法的最早应用；1896年，Pareto提出了Pareto最优的概念，这是多目标优化问题的基础；1909年，Erlang运用概率论研究了电话服务问题，这被认为是排队论的起源；1912年，Zermelo最早研究了对策论问题；1915年，Harris研究了商业中的库存问题，这被认为是库存论研究的起源；1916年，Borel提出了对策论中的最优策略的概念，证明了一些对策论问题存在最优策略；1928年，Von Nuemann深入研究了对策论问题，并且出版了著作《对策论和经济行为》；1932年，Weibull研究了维修问题，这被认为是可靠性数学理论的起源；1939年，Kantorovich提出了线性规划，并且因此获得了1975年的诺贝尔经济学奖。

二战期间，英国组织了一批科学家成立了军事数学小组，来对军事作战计划进行研究。例如，如何组织护航编队可以使得运输船队遭遇敌方攻击时损失最小；如何改进搜寻方法，更好的发现敌方潜艇；如何改进深水炸弹的爆炸深度，更好的毁伤敌方潜艇等等。当时，英国称这些研究为“opearation research”，也就是“运筹学”。二战之后，这些研究成果逐渐得到了发表，并且逐渐被应用到了生产、生活、经济和管理等领域当中，这被认为是运筹学的起源。

运筹学迎来大发展是在20世纪40年代到60年代。1947年，丹齐格研究了线性规划问题的一种解法——单纯形法，在1949年，线性规划理论建立，标志着运筹学大发展的到来。之后，1951年，非线性规划理论建立；1954年，网络流理论建立；1955年，随机规划理论建立；1958年，整数规划理论建立。同时，排队论、存储论、决策论和对策论也都得到了巨大的发展。

这期间，计算机的发明大大促进了运筹学的发展，使得运筹学中许多复杂的模型得以求解；同时，运筹学的发展也反过来促进了计算机的发展，使得计算机向更大规模更快速度的方向快速发展。

如今，运筹学已经广泛应用在生产、生活、经济、管理、政治和军事的各个领域中，创造了巨大的社会财富。

20世纪50年代中后期，科学家钱学森、许国志将运筹学从西方引入到我国，将“operation research”译作“运筹学”，取自古语“运筹帷幄之中，决胜千里之外”。

1956年，在钱学森、许国志的推动下，中科院力学所成立了中国第一个运筹学小组。1959年，中科院数学所成立了中国第二个运筹学小组。1960年，力学所运筹学小组和数学所运筹学小组合并成为了数学所的一个研究室，主要负责研究非线性规划、动态规划、排队论、图论方向。50年代后期，“打麦场的选址问题”、“中国邮递员问题”相继提出。60年代中后期，“优选法”、“统筹法”在华罗庚先生的大力推广下大大普及，同时，运筹学思想随之广泛传播，大大推动了运筹学的发展。

之后的几十年中，运筹学在我国发展迅速。如今，运筹学应用在生产、生活、经济、管理、政治和军事的各个领域中。

## 1.2 研究现状

在生产、生活、交通、物流、网络中我们经常遇到这样一类问题–选址问题(工厂、物流中心、网络服务中心等设施位置的选取问题)。设施选址问题就是运筹学研究的一类问题。

设施选址的好坏会影响服务的质量、服务的效率、服务的成本等方面，进而影响服务取得的利润和服务的竞争力。并且，选址的决策的影响是深远的，所以选址问题是重要的长期决策。所以，研究选址问题具有十分重要的经济和社会意义[1]。

1909年，Weber研究了如何在平面上选取一个仓库的位置使得仓库与多个顾客的总距离最小的问题，这被认为是选址问题研究的开始。1964年，Hakimi研究了网络的*p*−中值问题和*p*−中心问题，这篇文章推动选址问题的研究。自此之后，选址问题的相关研究论文数量急剧增多，选址问题的研究也开始活跃了起来[1]。

选址问题大致可以分为中值问题、覆盖问题和中心问题三种经典问题。

### 1.2.1 *p*−中值选址问题

*p*−中值选址问题研究选取*p*个设施的位置来服务若干个需求点，使得总的成本最小的问题。*p*−中值问题的目标可以是使需求点到最近的设施的总距离最短，可以是总运输时间最少，也可以是总运输费用最少。*p*−中值问题最早由Hakimi于1964年提出。

*p*−中值问题可以分为绝对*p*−中值问题和顶点*p*−中值问题，绝对*p*−中值问题是指可以选取网络上的任意位置作为设施的选址，而顶点*p*−中值问题则是指只能选取顶点作为设施的选址。

根据Hakimi提出的顶点最优性质: *p*−中值问题至少有一个最优解完全是顶点。绝对*p*−中值问题与顶点*p*−中值问题的有相同的最优解，区别就仅仅在于约束条件的不同，所以一般不区分这两种问题。

Revelle和Swain建立了*p*−中值问题的整数规划模型，则*p*−中值问题的模型可以描述为

$$\min\sum\_{i \in I}^{}{\sum\_{j \in J}^{}{f\_{i}d\_{ij}X\_{ij}}}$$  
$$s.t.\sum\_{j \in J}^{}X\_{ij} = 1,\ \forall i$$  
*Xij* − *Yij* ≤ 0, ∀*i*, ∀*j*  
$$\sum\_{j \in J }^{}Y\_{j} \leq P$$  
*Xi*, *Yj* ∈ {0, 1}, ∀*i*, ∀*j*  
其中，*i*为需求点的序号，*j*为备选设施的序号，*fi*为需求点*i*的需求量，*dij*为需求点*i*到备选设施*j*的距离，*Xij*为备选设施*j*是否为需求点*i*服务，若备选设施*j*为需求点i服务，则*Xij*为1，否则为0。约束(1)式表示备选设施至多只能为一个需求点服务，约束(2)式表示只有建立设施才能为需求点服务，约束(3)式表表示建立设施的最大数量不能超过P，约束(4)式是变量的取值范围约束。

Hakimi和Kariv证明了一般网络的*p*−中值问题是NP-完全的，如果是顶点*p*−中值问题，可行解的数目是 $\begin{pmatrix} n \\ p \\ \end{pmatrix} = \frac{n!}{p!\left( n - p \right)!} = O\left( n^{p} \right)$。 所以，当*p*为定值时，*p*−中值问题显然是在多项式时间内可解的；当*p*为变量时，*p*−中值问题仍然是NP-完全的，我们可以求从*p* = 1到*p* = *n*的所有*p*−中值问题，这仍然是在多项式时间内可解的[1]。

因为*p*−中值问题可以表示为整数规划模型，所以*p*−中值问题的求解可以用整数规划的求解方法来求解。常用的求解方法有基于对偶的多阶段法、对偶定界法、拉格朗日松弛法和分支定界法等等。还有一些启发式算法，例如近似算法、交换启发式算法、邻域搜索算法等等。以上的算法当中，拉格朗日松弛法式最好的，拉格朗日松弛法得到的解的质量高，并且可以给出解的上界和下界，可以用于评价解的质量。但是拉格朗日松弛法的求解耗时较长，而且得不到1-中值问题到(*p* − 1)−中值问题的解。

另外，一些现代启发式算法也被用于求解*p*−中值问题，例如禁忌搜索算法、模拟退火算法、遗传算法和神经网络算法等等，也取得了不错的效果。

*p*−中值选址问题常常应用在企业选址中，例如工厂、仓库的选址，因为企业选址问题常常以成本最低为目标，这一类追求经济效益最大化的目标通常被称为“经济效益性”目标。同时，*p*−中值选址问题也可以应用于公共设施的选址中，例如学校、图书馆的选址。这一类追求公共服务效果最好的目标通常被称为“集体福利性”目标。*p*−中值问题常常解决的就是这两类目标的问题。

### 1.2.2 覆盖选址问题

首先，我们引入覆盖的概念: 当某个需求点可以在规定的时间内被某个设施服务，则称该需求点被该设施覆盖。

覆盖选址问题研究的就是用最少成本的设施来最大程度的满足需求点需求的问题。

覆盖问题可以分为集合覆盖问题和最大覆盖问题两种。

集合覆盖问题最早由Daskin于1964年提出，其目标是建设若干设施在满足所有需求点需求的条件下使建设成本最小，模型可以描述为

$$\min\sum\_{j \in J}^{}{c\_{j}Y\_{j}}$$  
$$s.t.\sum\_{d\_{j} \leq D}^{}Y\_{j} \geq 1,\ \forall i$$  
*Yj* ∈ {0,1}, ∀*i*, ∀*j*  
其中，*cj*为备选设施*j*建设的成本，*Xij*为备选设施*j*是否为需求点*i*服务，若备选设施*j*为需求点*i*服务，则*Xij*为1，否则为0，*Yj*为备选设施*j*是否建设，若备选设施*j*建设，则*Yj*为1，否则为0。约束(1)式表示每个需求点至少有一个备选设施为其服务，约束(2)式是变量的取值范围约束。

该模型假设了所有的需求点的需求都应被满足，但是实际上，如果某个需求点的需求量很小，那么为了满足这个需求点的需求而建设设施很可能导致入不敷出，所以就有了最大覆盖问题。

最大覆盖问题最早由Church RL和ReVelle C于1988年提出，其目标是建设*p*个设施使需求点被满足的需求量最大，其模型可以被描述为

$$\min\sum\_{i \in I}^{}{f\_{i}X\_{i}}$$  
$$s.t.\sum\_{d\_{ij} \leq D}^{}Y\_{j} \geq X\_{i},\ \forall i$$  
$$\sum\_{j \in J}^{}Y\_{j} \leq P$$  
*Xi*, *Yj* ∈ {0, 1}, ∀*i*, ∀*j*  
其中，*dij*为需求点*i*到备选设施*j*之间的最短距离，*D*为设施与需求点之间允许的最大距离，*fi*为需求点*i*的需求量，*Xi*为需求点是否被覆盖，若需求点被覆盖，则*Xi*为1，否则为0。*Yj*为备选设施*j*是否建设，若备选设施*j*建设，则*Yj*为1，否则为0。约束(1)式表示需求点被覆盖时应当满足的条件，约束(2)式表示建立设施的最大数量不能超过*P*，约束(3)式是变量的取值范围约束。

与*p*−中值问题类似，一般网络上的覆盖选址问题是NP-完全的。

Hakimi提出的顶点最优性质不适用于覆盖选址问题，实际上，通常没有顶点限制时的解要优于有顶点限制的解。但是，Church RL和Meadows B提出了覆盖选址问题的伪顶点最优性质: 对于任何网络，存在一个顶点有限扩充集合，它至少包含集合覆盖问题或最大覆盖问题的一个最优解。

集合覆盖问题的求解有精确求解方法，例如分支定界法、基于动态次梯度的分支定界法等等。另外，一些启发式算法也可以用来求解覆盖选址问题，例如替换启发式方法、拉格朗日松弛法、贪婪算法、遗传算法、神经网络算法、混合启发式算法以及其他启发式算法等等。

而最大覆盖问题的求解的方法主要有分支定界法，除此之外就是贪婪添加算法、贪婪添加-替换算法、拉格朗日松弛法等启发式算法。

覆盖选址问题常常应用在应急服务设施的选址上，例如消防站、急救中心等，因为应急服务设施通常满足在一定时间内或者一定距离内可以服务到需求点即可(即覆盖到需求点)，并且通常希望一定的应急服务设施服务到的需求点越多越好。

### 1.2.3 *p*−中心选址问题

*p*−中心选址问题研究选取*p*个设施的位置来服务若干个需求点，使得最差的情况达到最优的问题。*p*−中心问题的目标可以是最大反应时间最小，可以是最大距离最小，也可以是最大损失最小。*p*−中心问题最早也是由Hakimi提出的。

*p*−中心问题可以分为绝对*p*−中心问题和顶点*p*−中心问题，其区别与*p*−中值问题的分类中的两种问题的区别类似。

*p*−中心选址模型可以描述为

min *D*  
$$s.t.\sum\_{j \in J}^{}X\_{ij} = 1,\forall i$$  
*Xij* − *Yj* ≤ 0, ∀*i*, ∀*j*  
$$\sum\_{j \in J}^{}Y\_{j} = p$$  
$$D - \sum\_{j \in J}^{}{d\_{ij}X\_{ij}} \geq 0,\ \forall i$$  
*Xij*, *Yj* ∈ {0, 1}, ∀*i*, ∀*j*  
其中，*dij*为需求点*i*到备选设施*j*之间的最短距离，*D*为设施与需求点之间允许的最大距离，*Xij*为备选设施*j*是否为需求点*i*服务，若备选设施*j*为需求点*i*服务，则*Xij*为1，否则为0，*Yj*为备选设施*j*是否建设，若备选设施*j*建设，则*Yj*为1，否则为0。约束(1)式表示备选设施至多只能为一个需求点服务，约束(2)式表示只有建立设施才能为需求点服务，约束(3)式表表示建立设施的最大数量不能超过P，约束(4)式表示设施与需求点之间的距离要小于允许的最大距离，约束(5)式是变量的取值范围约束。

与覆盖选址问题类似，Hakimi提出的顶点最优性质同样不适用于*p*−中心选址问题，实际上，通常绝对*p*−中心选址问题的解要优于顶点*p*−中心问题的解。但是，Kariv和Hakimi提出了*p*−中心选址问题的伪顶点最优性质: 对于任何网络，存在一个顶点有限扩充集合，它包含绝对*p*−中心选址问题的一个最优解。

*p*−中心选址问题本质上是minmax问题，其最优目标值通常被称为*p*−半径，这是一种降低风险的方法。通常，*p*−中心选址问题常常应用在军队、医院、应急服务设施的选址上，因为这些设施通常有服务承诺，或者说这些设施的目标通常是降低最坏情况发生的风险，这一类目标通常被称为“经济平衡性”目标。

### 1.2.4 其他选址问题

后来，研究学者们以这三个选址问题作为基础，研究了更贴近实际也更复杂的一些选址问题模型，这些模型主要有带固定费用和容量限制的选址问题模型、多目标选址问题模型、可靠性选址问题模型、竞争选址问题模型、截流问题模型、hub选址问题模型、随机选址问题模型和动态选址问题模型等等。

## 1.3 主要研究问题和方法

本文主要研究容量限制的设施选址问题。

容量限制的设施选址问题考虑现实中设施的建设费用以及设施的容量限制。而这一类模型又通常分为两类: 一类是无容量限制的固定费用选址问题(Uncapacitated Facility Location Problem, UFLP)，这一类问题考虑设施建设费用，但是不考虑设施的容量限制；另一类是带容量限制的选址问题(Capacitated Facility Location Problem, CFLP)，这一类问题又可以分为弱容量限制的选址问题(Soft-Capacitated Facility Location Problem, SCFLP)和强容量限制的选址问题(Hard-Capacitated Facility Location Problem, HCFLP)。其中，弱容量限制的选址问题对设施的连接数加以限制，每个设施最多连接*ui*个顾客，但是每个地点允许建设多个设施，而强容量限制的是选址问题，除了限制每个设施最多连接*ui*个顾客，还限制每个地点仅能建设一个设施。

# 2 强容量限制的设施选址问题

## 2.1 模型描述

强容量限制的设施选址问题在生产和生活中十分常见。它研究的就是在考虑设施服务能力限制的条件下，选取设施的位置来服务顾客，使得总成本(设施建设成本和服务成本)最小的问题。

强容量限制的选址问题模型的可以描述为下列的整数规划模型:

$$\min\sum\_{i \in I}^{}{f\_{i}y\_{i}} + \sum\_{i \in I}^{}{\sum\_{j \in J}^{}{c\_{ij}x\_{ij}}}$$  
*s*.*t*.*xij* ≤ *yi*, ∀*i*, ∀*j*  
$$\sum\_{j \in J}^{}w\_jx\_{ij} \leq u\_{i}y\_{i},\forall i$$  
$$\sum\_{i \in I}^{}x\_{ij} = 1,\ \forall j$$  
*xij*, *yj* ∈ {0, 1}, ∀*i*, ∀*j*  
其中，*i*为顾客的序号，*j*为备选设施的序号，*fi*为设施*i*的建设费用，*cij*为设施*j*为顾客*i*提供服务的服务费用，*xij*为备选设施*j*是否为需求点*i*服务，若备选设施*j*为需求点*i*服务，则*xij*为1，否则为0。约束(1)式表示只有建立的设施才能为顾客提供服务，约束(2)式表示备选设施服务的顾客数不能超过它的容量上限，约束(3)式表示，约束(4)式是变量的取值范围约束。 ## 2.2 数学性质 **性质1** 如果设施*Fi*1满足下列条件：

1. 存在设施*Fi*2 ∈ *F*，满足*Fi*2的建设费用小于*Fi*1的建设费用
2. 对于任意顾客*Cj* ∈ *C*，*Fi*2的服务费用小于*Fi*1的服务费用

则称*Fi*1为绝对劣势的设施，设施*Fi*1一定不建设。

**证明:** 在强容量限制的选址问题中，设施*Fi*2可以用设施*Fi*1来替代，且替代后可以使目标函数更优，故不建设设施*Fi*2的解要优于建设设施*Fi*2的解，故设施*Fi*2一定不建设。

**性质2** 如果设施*Fimin*满足下列条件：

1. 对任意设施*Fik* ∈ *F*，都满足*Fimin*的建设费用小于*Fik*的建设费用
2. 对任意设施*Fi* ∈ *F*，和任意顾客*Cj* ∈ *C*，都满足设施*Fimin*的服务费用小于设施*Fi*的服务费用

则设施*Fimin*一定建设。并且，建设[*n*/*uimin*](向上取整)个设施*Fimin*，使设施*Fimin*为所有顾客服务。

**证明:** 容量限制的选址问题中，此性质显然成立。

**性质3** 如果可以为顾客*Cj*提供服务的所有设施中，设施*Fi*1提供的服务费用最小，设施*Fi*2提供的服务费用次小，且设施*Fi*1提供的服务费用加上设施*Fi*1的建设费用小于设施*Fi*2提供的服务费用，则顾客*Cj*一定由设施*Fi*1提供服务。

**证明:** 在软容量限制的选址问题中，设施可以建设多次，故如果顾客*Cj*被其他设施服务，那么可以用设施*Fi*1来替代其他设施，则替代后服务费用减少，目标函数更优，故设施*Fi*1为顾客*Cj*提供服务的解要优于其他设施为顾客*Cj*提供服务的解，故顾客*Cj*一定由设施*Fi*1提供服务。

## 2.3 求解算法

本文采用一种新型的启发式算法——竞争决策算法来对模型进行求解。 ### 2.3.1 竞争决策算法简介 竞争决策算法(Competitive Decision Algorithm, CDA)是在某种竞争规则和决策规则下，根据优胜劣汰的性质来求解组合优化问题的算法。

竞争决策算法的思想来自于自然界中优胜劣汰，适者生存的性质。自然界中资源的分配往往是在一定的初始状态下，在竞争者的绝对实力、竞争者间的相对实力差距、竞争者对环境的适应能力等多因素的影响下，经过多轮竞争和决策，最终达到另一种资源的分配状态，如果这一状态的结果要优于初始竞争状态的结果，那么就达到了优化的目的。

竞争决策算法就是利用这一原理，通过构造多个竞争者对有限的资源进行竞争，根据优胜劣汰的原则对资源重新进行分配，最终使得一些更具优势的竞争者获得更多的资源，而处于劣势的竞争者失去资源的占有权，甚至死亡，最终使得总体目标最优。

### 2.3.2 竞争决策算法基本概念

竞争者: 参与竞争的对象，可以占有和竞争资源。如果只有一个竞争者，可以假设一个虚拟竞争者，代表自然，虚拟的竞争者没有竞争决策函数。

资源: 竞争者竞争的对象。

竞争力函数: 用于计算某个竞争者对某个资源的竞争力。

决策函数: 用于判定资源的分配。

竞争决策状态: 指的是某个时刻，每个竞争者占有资源的状态。初始状态就是竞争开始前，每个竞争者占有的资源状态。

竞争决策均衡: 指的是一种特殊的竞争决策状态，在这种状态下，不存在一个非虚拟的竞争者能够根据竞争力函数以及决策函数占有更多的资源。

资源交换规则: 是指当达到竞争决策均衡时，通过资源交换规则交换资源，使得达到一种非均衡的状态，从而可以让竞争者重新竞争资源，再次达到一个竞争决策均衡状态。

### 2.3.3 竞争力函数和决策函数

在选址问题中，将每个设施看作一个竞争者，将顾客看作被设施竞争的资源，设施为顾客服务即竞争者获得了资源，用集合表示竞争者获得的资源。

竞争决策状态为当前每个竞争者占有的资源情况，即设施为顾客服务的情况，也是选址问题整数规划模型的一个解。

竞争决策状态的初始状态为每个竞争者都不占有资源，即没有任何设施为顾客服务。

竞争决策状态的结束状态为每个竞争者最终占有的资源情况，即最终每个设施服务顾客的情况。

竞争力函数为设施对顾客的吸引力，与设施的服务费用成反比，同时，如果当前服务顾客需要建设设施，则与设施的服务费用和建设费用之和成反比，其表达式为:

$$compete\left( F\_{i},C\_{j} \right) = \left\{ \begin{matrix} \frac{1}{{(c}\_{ij} + f\_{i})}，设施未确定建设或设施容量已满 \\ \frac{1}{c\_{ij}},\ 设施已经确定建设且容量未满 \end{matrix} \right.\ $$  
决策函数为根据当前竞争决策状态，计算目标函数值，如果目标函数值大于目前最优竞争决策状态的目标函数值，则更新最优竞争决策状态。

## 2.3.4 竞争决策算法流程

**step 1:** 根据性质1至3，在竞争者集合中清除绝对劣势竞争者，保留绝对优势竞争者，并固定绝对优势竞争者获取到的资源。

若此时所有资源均被固定分配给了优势竞争者，则进入step 5；否则进入step 2

**step 2:** 设所有竞争者占有的资源均为空集｛｝

对每一个资源*Cj*，计算所有竞争者*Fi*对资源*Cj*的竞争力函数值，并将资源*Cj*分配给竞争力函数值最大的竞争者

**step 3:** 执行下列操作：

do

根据目前的竞争决策状态和竞争力函数，更新每一个竞争者对每一个资源的竞争力函数值 若任一竞争者*Fi*1对顾客*Cj*的竞争力函数大于当前占有该资源的竞争者*Fi*2，则根据决策函数，将资源*Cj*分配给竞争者*Fi*1，形成新的竞争决策状态

while

任一竞争者都不能根据决策函数获得更多的资源，即达到竞争决策均衡

**step 4:** 在当前竞争者集合中清除未占有资源的竞争者，在当前的竞争者集合中执行下列操作

do

若交换任一竞争者*Fi*1拥有的资源*Cj*1和另一竞争者*Fi*2拥有的资源*Cj*2，可使得目标函数更优，则交换两个资源

while

交换任意两个竞争者的任意两者资源都不能使目标函数更优

**step 5:** 算法结束，输出竞争结果

## 2.4 算例

**算例1:**

设候选设施集合*Fi* ∈ *F*, *i* = 1, 2, 3, 4, 5 ，顾客集合*Cj* ∈ *C*, *j* = 1, 2, 3, 4, 5，设每个候选设施的服务能力为1，每个顾客的需求为1，候选设施的建设费用:  
*H* = [10, 29, 22, 16, 17]  
候选设施与顾客的服务费用:  
$$D=\begin{bmatrix} 3 & 7 & 12 & 13 & 14 \\ 17 & 13 & 14 & 16 & 17 \\ 14 & 9 & 11 & 10 & 6 \\ 15 & 10 & 8 & 6 & 5 \\ 16 & 11 & 9 & 7 & 6 \end{bmatrix}$$  
**算法计算过程:**

第一步：根据性质1，因为设施*F*2的建设费用和为任何顾客服务的费用都大于设施*F*1，即设施*F*2处于绝对劣势，所以在候选设施集合中清除设施*F*2，并且在矩阵中删除行*D*2 = [17, 13, 14, 16, 17]。根据性质3，设施*F*2对顾客*C*1的服务费用最小，且小于服务费用次小的设施*F*2对顾客的服务费用减去*F*2的建设费用，所以*F*2一定建设，且一定为顾客*C*1服务。

第二步：

# 3 软容量限制的设施选址问题

## 3.1 模型描述

软容量限制的选址问题研究的是设施提供服务的能力有限，但是可以通过多次建设设施来使设施的服务能力得到提升，目标是追求建设费用和服务费用之和最小。

软容量限制的选址问题模型的可以描述为下列的整数规划模型:

$$\min\sum\_{i \in I}^{}{f\_{i}y\_{i}} + \sum\_{i \in I}^{}{\sum\_{j \in J}^{}{c\_{ij}x\_{ij}}}$$  
*s*.*t*.*xij* ≤ *yi*, ∀*i*, ∀*j*  
$$\sum\_{j \in J}^{}w\_jx\_{ij} \leq u\_{i}y\_{i},\forall i$$  
$$\sum\_{i \in I}^{}x\_{ij} = 1,\ \forall j$$  
*xij* ∈ {0, 1}, ∀*i*, ∀*j*  
*yj* ∈ *N*\*, ∀*j*  
其中，*i*为顾客的序号，*j*为备选设施的序号，*fi*为设施*i*的建设费用，*cij*为设施*j*为顾客*i*提供服务的服务费用，*xij*为备选设施*j*是否为需求点*i*服务，若备选设施*j*为需求点*i*服务，则*xij*为1，否则为0。约束(1)式表示只有建立的设施才能为顾客提供服务，约束(2)式表示备选设施服务的顾客数不能超过它的容量上限，约束(3)式表示，约束(4)(5)式是变量的取值范围约束。

## 3.2 数学性质

强容量限制的设施选址问题中的性质1-3在软容量限制的选址问题中均适用。

# 4 基于混合策略的竞争决策算法

基于混合策略的竞争决策算法是一种改进的竞争决策算法。在一般的竞争决策算法中，在一定的竞争决策状态下，每个竞争者的策略是一定的，即争夺竞争力函数值最大的资源，这是一种简单的纯策略。

但是，自然界中，竞争者往往不是以简单的纯策略来进行决策的，竞争者的决策受到很多随机因素的影响，包括内部随机因素和外部随机因素。例如在选址问题中，设施服务顾客的决策往往受到设施内部管理、经营因素的影响，以及外部经济、政治环境的影响，而存在一定的随机性，所以假设竞争者的策略是简单的纯策略是不够合理的。

所以这里我们引入混合策略的概念，即在一定的竞争决策状态下，每个竞争者的策略是以某种概率分布随机的争夺不同的资源。而纯策略可以看作是混合策略的一种特例，即选择一种策略的概率为1，选择其余策略的概率为0，所以采取混合策略是对竞争决策算法中策略的推广。

基于混合策略的竞争决策算法通过引入混合策略的概念更好的模拟了自然界中竞争者的决策，也更好的模拟了竞争者对资源的竞争。

## 4.1 混合竞争策略

竞争策略: 竞争者在某局中的策略。

在选址问题中，竞争策略即竞争某个资源，而竞争者的竞争策略取决于该竞争者的竞争力，我们假设竞争者均采取下面的混合策略，

竞争者*Fi*争夺资源*Cj*的概率:

$$h\left( F\_{i},C\_{j} \right) = \frac{compete(F\_{i},\ C\_{j})}{\sum\_{j \in j}^{}{compete(F\_{i},\ C\_{j})}}$$  
由上式可以看出，竞争者对于资源的竞争力越大，竞争者争夺该资源的概率就越大，这样保证了竞争者争夺到有利的资源的几率较大，而争夺到不利资源的几率较小。

## 4.2 算法流程

**step 1:** 根据性质1至3，在竞争者集合中清除绝对劣势竞争者，保留绝对优势竞争者，并固定绝对优势竞争者获取到的资源。

若此时所有资源均被固定分配给了优势竞争者，则进入**step 5**；否则进入**step 2**

**step 2:** 设所有竞争者占有的资源均为空集｛｝

**step 3:** 如果任一竞争者都不能根据决策函数获得更多的资源，即达到竞争决策均衡，则进入**step 6**；否则进入**step 4**

**step 4:** 根据目前的竞争决策状态，更新每一个竞争者对每一个资源的竞争力函数值

**step 5:** 竞争者按照次序根据竞争策略争夺资源，如果根据竞争策略，*Fi*争夺资源*Cj*，且根据决策函数，将资源*Cj*分配给竞争者*Fi*可以使得目标函数更优，则将资源*Cj*分配给竞争者*Fi*，形成新的竞争决策状态，进入**step 4**；

**step 6:** 算法结束，输出竞争结果

## 参考文献

[1]:杨丰梅,华国伟,邓猛,et al.选址问题研究的若干进展[j].运筹与管理,2005,14(6):1-7.