

# 2018 年中国海洋大学数学建模竞赛

## 参赛队员信息

参赛题号（从 A/B/C/D 中选择一项填写）：                     **B**                    

队名	姓名	学号	年级专业	电子邮箱	联系电话
文明富强	黄佳城	16060012014	16 海渔	<a href="mailto:305179568@qq.com">305179568@qq.com</a>	17864295856
	李鑫军	16120012018	16 海渔	<a href="mailto:277329640@qq.com">277329640@qq.com</a>	17755655789
	但心怡	16050011009	16 生科	<a href="mailto:1060207280@qq.com">1060207280@qq.com</a>	17854223732

# 无线回传拓扑规划

## 摘要

Relay 无线回传方案利用 FDD LTE 或 TDD LTE 制式承载来为站点回传，相对微波有较强的 NLOS（非视距）传输能力，可以解决城区、农网等场景下的传统传输方式不可达的问题，同时在部分场景下也可以替代微波，有效降低站高，节省建站费用。

我们首先分析了 Relay 无线回传方案：整个架构由两大类型的站点构成：宿主站和子站。宿主站之间的交流由微波连接，传送的距离受到 LOS（视距）的影响：传送范围不能超过 50km。宿主站和子站与子站和子站之间的连接有无线回传实现连接，连接有以下限制：宿主站和总子站的连接（首跳）不能超过 20km，子站和子站之间的连接（第二跳、第三跳）距离不能超过 10km。无线中继的跳数不能超过 3 跳。从这样的架构中，我们将宿主站和子站的信号中继范围和信号传播范围分开：由于手机不能接受微波的信号，只能接受无线回传的信号。

题目要求我们根据提供的站点和站点信息给出满足成本最低和损耗最小的目标。我们将整个过程分成两个部分来实现：首先确定宿主站的位置，再确定子站的位置。

宿主站的位置主要有站点的地理位置和不同站点的信号中继范围和信号传播范围来决定：对于一个宿主站和其连接的子站，在空间地理分布上呈一个簇状分布，因此为了确定这个簇和宿主站的位置，我们考虑先对给出的站点进行 AP 聚类算法，以球面三角距离公式进行亲和度聚类，便得到了一个具有聚类中心的簇，先取聚类中心作为可能的宿主站候选集。接着使用最大覆盖模型进行对于宿主站进一步的筛选，尽量使得宿主站的信号可以覆盖尽可能多的站点。但是这个时候得到的宿主点有一些仍不能满足宿主站之间的距离要求：小于 50km。因此要遍历每个聚类中心之间的距离，如果不满足条件的宿主站，则对其所在簇运用 K-means 进行进一步聚类，使得宿主站满足所有的条件。

第二步确定子站的位置，必须由一些隐藏条件和题目给出的条件来实现：我们认定在覆盖尽可能多的需求点的条件下，我们需要同时保证该地区的信号覆盖率足够高：也就是说整个区域的面积减去宿主站和子站的信号覆盖范围的差需要足够的小，保证整个区域内信号覆盖率。事实上这个条件是确定子站最为重要的一个条件，我们将其认作为基础条件，利用这个条件可以做到先筛选子站分布的结果，这也应该是构建一个无线拓扑结构的初衷。在满足覆盖率最大的条件下，又同时满足成本最小和损耗最小的目标，那么这就是一个多目标规划问题。我们针对这个题目开发了以下多目标规划模型：同时满足基础条件和成本最小、基础条件和损耗最小、基础条件和成本最小和损耗最小三个多目标规划。为了满足题目对于算法效率的要求，我们利用智能粒子群算法、模拟退火算法、蚁群算法、这三种智能算法来降低整个多目标规划的计算复杂度，并根据站点数的不同划分出最佳运行速度的算法。

关键字：AP 聚类算法、最大覆盖模型、K-means 算法、多目标规划、智能算法

# 目录

摘要.....	2
一、 问题重述.....	4
二、 问题的分析.....	10
三、 模型假设.....	11
四、 符号与约定.....	11
五、 模型的建立和求解.....	12
5.2 确定宿主站.....	13
5.2 确定子站.....	15
5.2.1 题目分析.....	15
5.2.2.1 多目标规划的数学模型.....	15
5.2.2.2 多目标规划解的定义.....	16
5.2.2.3 多目标规划的基本解法.....	16
5.2.3 一些参数.....	16
5.2.4 成本最低.....	16
5.2.5 损耗最低.....	16
5.2.6 覆盖率最大.....	17
5.2.7 多目标规划的三个模型.....	17
5.3 算法效率优化.....	17
六、 模型检验.....	19
七、 模型评价.....	19
7.1 模型的优点.....	19
7.2 模型的缺点.....	20
八、 参考引文.....	20
九、 代码附录.....	20

## 一、 问题重述

### 1. 背景介绍

在城区建设基站，传输光纤部署最后一公里的成本高，光纤到站率低，全球综合来看低于 60%；如果使用微波传输，由于微波只能在 LOS（视距）场景下部署，而城区场景中 LOS 信道比例低于 50%。

在农村网建设基站，单站业务量低，收入低，ROI（投资回报率）差，运营商建站对成本较为敏感。卫星传输租金、光纤传输建设费用对于运营商是很大的负担，而如果使用微波传输，对于相当一部分站点需要提升铁塔高度来满足微波的 LOS 场景要求，铁塔费用的增加对于运营商来说同样是不小的负担。

Relay 无线回传方案利用 FDD LTE 或 TDD LTE 制式承载来为站点回传，相对微波有较强的 NLOS（非视距）传输能力，可以解决城区、农网等场景下的传统传输方式不可达的问题，同时在部分场景下也可以替代微波，有效降低站高，节省加站费用。

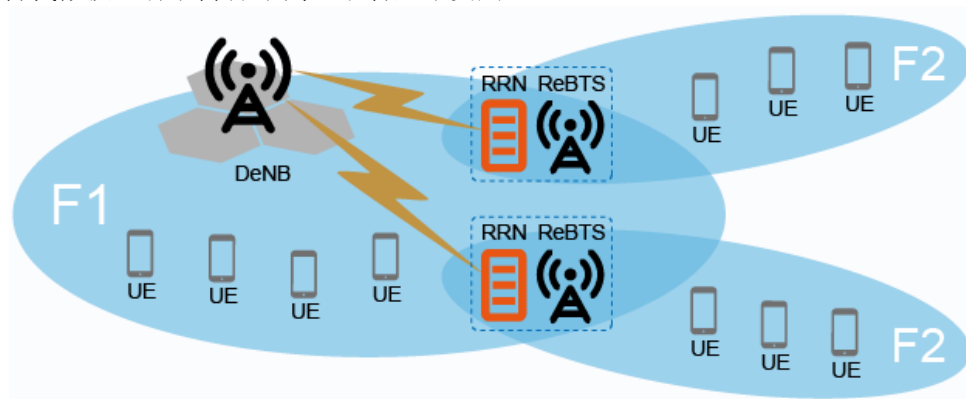


图 1 Relay 架构

RRN（eRelay Remote Node），是 Relay 方案中的无线回传设备，它用于为基站提供无线回传服务。如图 1 所示。Relay 组网包含宿主基站 DeNB 和中继站 RN 两个逻辑节点：

- DeNB 是在普通基站（DeNB）上增加了 Relay 功能，DeNB 支持普通手机（UE）接入，也支持 RRN 的接入；
- RN 包括 RRN 和 ReBTS 两部分。RRN 通过无线信号接入 DeNB 并建立空口承载；ReBTS 可供覆盖范围内的 UE 接入；ReBTS 的传输由 RRN 提供

为了方便理解，这里分别将 DeNB 和 RRN 称作宿主站和子站，一个宿主基站通常可以有 1~3 个宿主小区，分别覆盖不同的方向（可理解为扇区的定义），如图 2 所示。图 2 中方块代表子站，每个宿主小区可以接入一定数量的子站，子站与子站之间可以级联（即多跳），但跳数有限制。

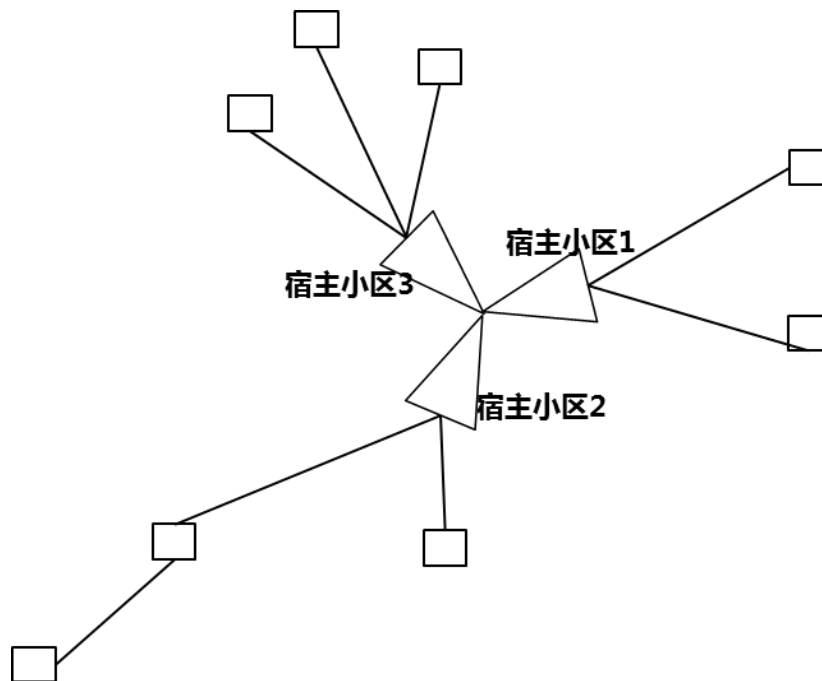


图 2 Relay 拓扑关系示意图

## 2. 任务表述

### 2.1 任务简述

本任务中，在给定一个地区中候选站点的位置分布的情况下，参赛队伍需要根据站点间的相互位置、站点间拓扑关系限制等条件，在满足一定回传质量（本次任务仅根据宿主站与子站的距离是否满足某门限来判断是否满足最低回传质量要求。而实际 Relay 部署时，影响回传质量的因素包括距离、地形阻挡、普通手机接入影响、ReBTS 干扰、相邻基站干扰等多种复杂因素）的前提下，设计成本最优的部站方案，包括：

- 候选站点是安装子站还是宿主站？
- 候选站点间的连接关系如何？

结合现网中对于无线回传拓扑规划问题的具体需求，算法还应该具有以下特点：算法收敛速度快、尽可能覆盖更多的站点。

### 2.2 输入输出

#### 1、输入：

每个地区内，所有站点列表，包括：

- 站点经纬度；

- 站型：RuralStar 或蝴蝶站；

各种站型的综合成本，包括：

- 宿主站的综合成本；
- 子站的综合成本；
- 卫星设备成本；

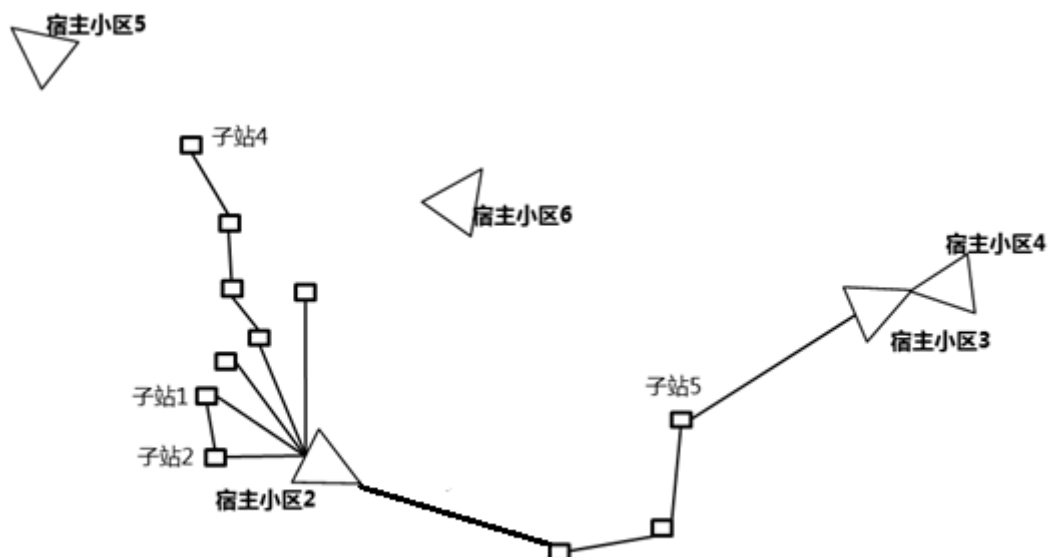
## 2、约束

输出的拓扑关系，应满足如下限制条件：

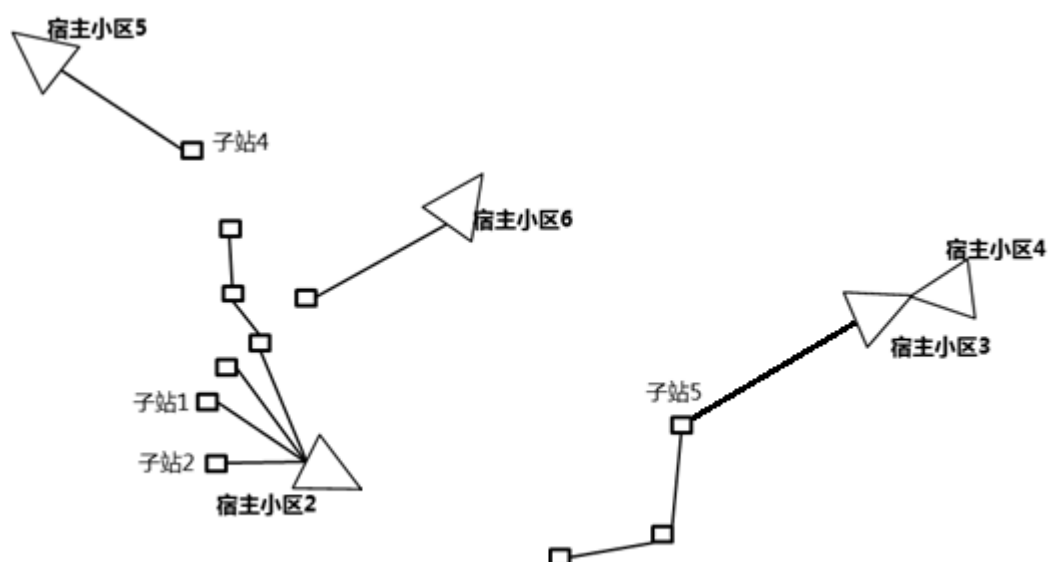
- 首跳距离 $\leq 20\text{km}$ ，之后每跳距离 $\leq 10\text{km}$
- 站点包含 RuralStar 和蝴蝶站两种不同站型；其中，RuralStar 共包含 1 个扇区，蝴蝶站共包含 2 个扇区；若该站点为宿主站，则每个扇区第一级最大接入子站数 4，最大总接入子站数 6；为了简化问题，暂不考虑蝴蝶站的扇区覆盖方向；
- 宿主站之间采用微波连接，最大通信距离为 50KM
- 宿主站和子站以及子站之间采用无线回传连接
- 每个子站最多只能有 2 条无线回传连接；
- 任意子站只能归属一个宿主站，到达所属宿主站有且只有一条通路，且该通路包含的跳数小于等于 3
- 任意宿主站都有且只有一颗卫星负责回传，成片连接的宿主站可共享同一颗卫星，但一颗卫星最多只能负担 8 个成片宿主站的回传数据
- 成片宿主站中，宿主站总数不设上限

例如，如下图所示的连接关系中

- 宿主小区 2 不满足“每个扇区第一级最大接入数 4，最大总接入数 6”
- 子站 1、子站 2 不满足“任意子站只能归属一个宿主站，到达所属宿主站有且只有一条通路”
- 子站 4 不满足“任意子站只能归属一个宿主站，到达所属宿主站有且只有一条通路，且该通路包含的跳数小于等于 3”中的“跳数小于等于 3”
- 子站 5 不满足“任意子站只能归属一个宿主站，到达所属宿主站有且只有一条通路，且该通路包含的跳数小于等于 3”中的“任意子站只能归属一个宿主站”



上图连接关系可修改如下（前提是其它约束条件也满足），即可满足约束条件：



### 3、输出：

按输入数据中站点顺序，输出以下数据：

输出文件包含以下两个

Graph.csv

包含：

- 1) 二维矩阵表示所有站点间的连接关系，0 表示没有连接关系，1 表示采用无线回传连接，2 表示采用微波连接；

Posi.csv

包含以下数组，按列存储：

1) 一维数组表示站点类型，0 表示子站，1 表示宿主站；

例如：

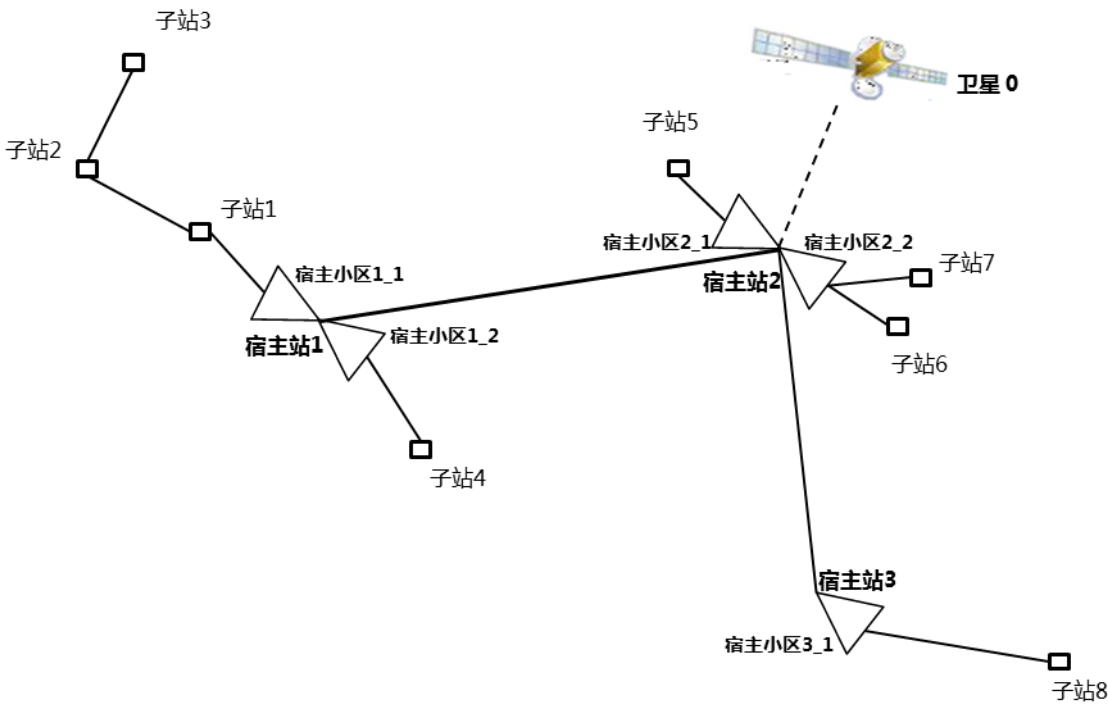


图 3 某个无线回传拓扑结构

如上图所示的连接关系，以上数组将表述为：

	宿主站 1	宿主站 2	宿主站 3	子站 1	子站 2	子站 3	子站 4	子站 5	子站 6	子站 7	子站 8
宿主站 1	0	2	0	1	0	0	1	0	0	0	0
宿主站 2	2	0	2	0	0	0	0	1	1	1	0
宿主站 3	0	2	0	0	0	0	0	0	0	0	1
子站 1	1	0	0	0	1	0	0	0	0	0	0
子站 2	0	0	0	1	0	1	0	0	0	0	0
子站 3	0	0	0	0	1	0	0	0	0	0	0
子站 4	1	0	0	0	0	0	0	0	0	0	0
子站 5	0	1	0	0	0	0	0	0	0	0	0
子站 6	0	1	0	0	0	0	0	0	0	0	0
子站 7	0	1	0	0	0	0	0	0	0	0	0
子站 8	0	0	1	0	0	0	0	0	0	0	0

站点名	站点类型
宿主站 1	1
宿主站 2	1
宿主站 3	1
子站 1	0



子站 2	0
子站 3	0
子站 4	0
子站 5	0
子站 6	0
子站 7	0
子站 8	0

算法效率：5 分钟内

站点规模：1000 站点左右

## 2.2 挑战目标

在拓扑架构满足约束条件的前提下，

**挑战目标 1（最高优先级）：更低的总体成本**

总体成本：宿主站数量\*宿主站成本+子站数量\*子站成本+卫星数量\*卫星成本

平均成本=总体成本/地区内站点总数

这里，卫星的数量等于  $\text{Ceil}(\text{宿主站数量}/8)$ ， $\text{Ceil}()$ 表示向上取整。

下表为各种传输方式的成本，单位：W USD

宿主站成本	10
子站成本	5
卫星成本	50

**挑战目标 2：更低的回传路径损耗**

虽然无线回传中存在 NLOS 影响，但为了简化问题，采用自由空间传播模型估计站点之间的路径损耗，公式如下：

$$PL=32.5+20*\lg(D)+20*\lg(F)$$

其中，PL 是路径损耗，是两个站点之间的距离，D 单位为 km，F 是发射频率，单位为 MHz，这里默认采用 900MHz。

系统平均损耗=所有无线回传连接的损耗之和/无线回传连接数

需要注意，该路径损耗只考虑子站回传部分，宿主站之间采用微波传输，只需满足距离限制，不计算该损耗。

附：球面距离公式

计算球面两点间距离的公式，设 A 点纬度  $\beta_1$ ，经度  $\alpha_1$ ；B 点纬度  $\beta_2$ ，经度  $\alpha_2$ ，则距离 S 为：

$$S=R \cdot \arccos[\cos \beta_1 \cos \beta_2 \cos (\alpha_1 - \alpha_2) + \sin \beta_1 \sin \beta_2]$$

其中  $R$  为地球半径，本题中取  $6378\text{km}$ ;

## 二、 问题的分析

首先，我们需要简单地分析一下 Relay 架构表达出来的无线回传结构的几点信息。整个架构由两大类型的站点构成：宿主站和子站。宿主站之间的交流由微波连接，传送的距离受到 LOS（视距）的影响：传送范围不能超过  $50\text{km}$ 。宿主站和子站与子站和子站之间的连接有无线回传实现连接，连接有以下限制：宿主站和总子站的连接（首跳）不能超过  $20\text{km}$ ，子站和子站之间的连接（第二跳、第三跳）距离不能超过  $10\text{km}$ 。无线中继的跳数不能超过 3 跳。显然，宿主站和子站之间由于设备的不同、连接方式的不同会有不同的信号传播和连接距离。由图 1 的图示已知：手机设备同时可以接受无线回传和微波两种信号传递方式，也就是说：对于宿主站，他的信号中继范围应该是  $50\text{km}$ ，对于子站，他的信号中继范围应该是  $10\text{km}$ 。而对于手机这种设备而言，他只能接受无线回传方式，并不能接受微波的通信方式。因此考虑到这一点，对于宿主站来说，他的信号通信范围因为第一条的限制距离，也就是  $20\text{km}$ ，而对于子站来说，他的信号通信范围应为  $10\text{km}$ 。对于一个成为宿主站的站点，他的类型还有蝴蝶站和 RuralStar 的区别：主要是站点的承载能力不同，蝴蝶站由于其两倍的扇形区域而拥有 RuralStar 两倍的承载能力。

题目中要求我们对输入的站点进行站点的规划：对输入的经纬度和站点的类型进行规划：决定哪个站是宿主站、哪个站是子站，并写出站点间的连接方式。加以成本最低和传播信号损耗最低的条件，给出不同的站点规划方式。我们将这个过程分成两个部分来完成：

首先完成对于宿主站的确定。对于一个宿主站和其连接的子站，在空间地理分布上呈一个簇状分布（如图 3 所示），因此为了确定这个簇和宿主站的位置，我们考虑先对给出的站点进行 AP 聚类算法，以球面三角距离公式进行亲和度聚类，便得到了一个具有聚类中心的簇，先取聚类中心作为可能的宿主站候选集。接着使用最大覆盖模型进行对于宿主站进一步的筛选，尽量使得宿主站的信号可以覆盖尽可能多的站点。但是这个时候得到的宿主点有一些仍不能满足宿主站之间的距离要求：小于  $50\text{km}$ 。因此要遍历每个聚类中心之间的距离，如果不满足条件的宿主站，则对其所在簇运用 K-means 进行进一步聚类，使得宿主站满足所有的条件。

第二步要做到对于子站的选定，这就需要辅助以题目要求我们达到的目标：成本最低或者损耗最小来达到这个目的。而在第一步完成之后，我们得到了确定的宿主点和其所有可能连接子站的簇。我们认定在覆盖尽可能多的需求点的条件下，我们需要同时保证该地区的信号覆盖率足够高：也就是说整个区域的面积减去宿主站和子站的信号覆盖范围的差需要足够的小，保证整个区域内信号覆盖率。事实上这个条件是确定子站最为重要的一个条件，我们将其认作为基础条件，利用这个条件可以做到先筛选子站分布的结果，这也应该是构建一个无线拓扑结构的初衷。在满足覆盖率最大的条件下，又同时满足成本最小和损耗最小的目标，那么这就是一个多目标规划问题。我们针对这个题目开发了以下多目标规划模型：同时满足基础条件和成本最小、基础条件和损耗最小、基础条件和成本最小和损耗最小三个多目标规划。为了满足题目对于算法效率的要求，我们利用智能粒子群算法、模拟退火算法、蚁群算法、这三种智能算法来降低整个多目标规划的计算复杂度，并根据站点数的不同划分出最佳运行速度的算法。以求达到 5 分钟 1000 个站点的规划计算。

### 三、 模型假设

1. 布置 Relay 架构时仅考虑由于站点距离而引起的信号衰减。
2. 不考虑蝴蝶站两个扇形区域的分布方式。
3. 考虑无线回传的损耗时采用的是空间传播模型。
4. Relay 架构建立时同时追求覆盖率要高。
5. 站点的信号辐射范围均为圆形。
- 6.

### 四、 符号与约定

序号	符号	含义
1	$x$	输入集中的站点
2	$S_i$	宿主站
3	$Z_i$	子站
4	$D(A, B)$	A、B 之间的距离
5	$Z-D(A, B)$	A、B 两个站点的信号中继范围
6	$C-D(A)$	A 类型站点的信号传播范围
7	$SB_i$	蝴蝶站
8	$SR_i$	RuralStar
9	$SC_i$	宿主站 $S_i$ 所属的 ‘宿主簇’

## 五、模型的建立和求解

### 5.1 Relay 架构的模型<sup>[1]</sup>

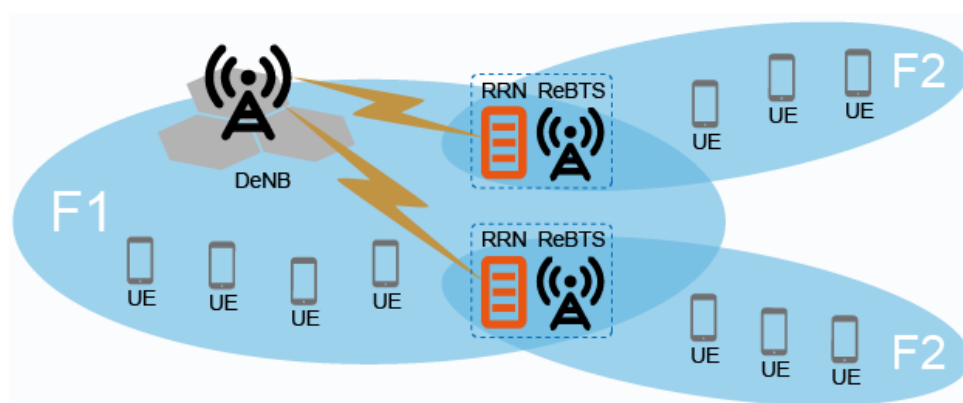


图 3 Relay 架构

Relay 架构如图 3 所示。分为两种站点：宿主站  $S_i$  与子站  $Z_i$ 。其中， $S_i$  与  $S_j$  的连接方式为微波连接，借助图 4 中的微波卫星进行中继。 $S_i$  与  $Z_i$  的连接方式为无线回传连接，借助图 3 中的 RRN 进行中继。 $Z_i$  与  $Z_j$  的连接方式也为无线回传连接。

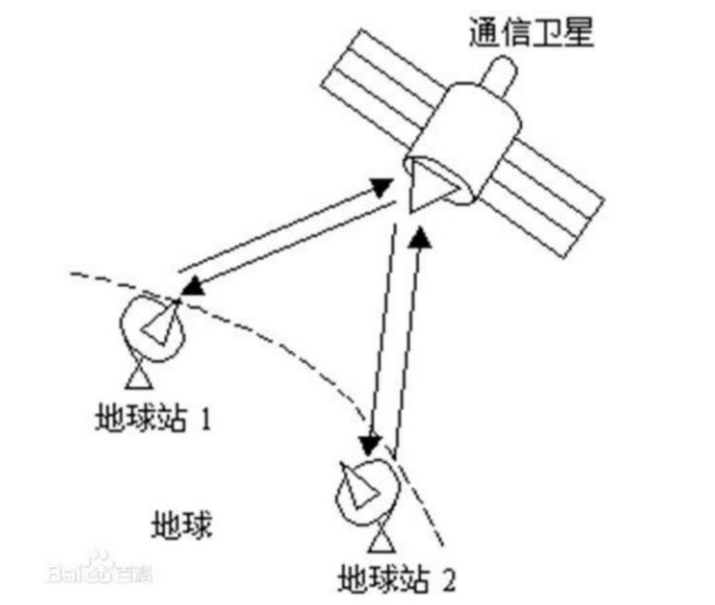


图 4 微波卫星运行方式

由于手机直接受无线回传的信号，以及其中继方式的不同。于是宿主站和子站便有着不同的信号中继范围  $Z-D(A, B)$  和信号传播范围  $C-D(A, B)$ 。其参数如下：

$$Z-D(S_i, S_j)=50\text{km}$$

$$Z-D(S_i, Z_j)=20\text{km}$$

$$Z-D(Z_i, Z_j)=10\text{km}$$

$$C-D(S_i)=20\text{km}$$

$$C-D(Z_i)=10\text{km}$$

宿主站  $S_i$  有两种站型：蝴蝶站  $SB_i$  与 RuralStar 站  $SR_i$ 。  $SB_i$  有  $SR_i$  两倍的承载能力，可以在第一级接入 8 个子站，最大总接入数为 12 个。•任意宿主站都有且只有一颗卫星负责回传，成片连接的宿主站可共享同一颗卫星，但一颗卫星最多只能负担 8 个成片宿主站的回传数据。子站  $Z_i$  则最多只能有 2 条无线回传连接，只能归属一个宿主站，到达所属宿主站有且只有一条通路，且该通路包含的跳数小于等于 3。

## 5.2 确定宿主站

### 5.1.1 AP 聚类算法确定 ‘宿主簇’<sup>[2]</sup>

首先我们定义宿主簇为一个以某个宿主站  $S_i$ 、子站  $Z_i$  和某些既不是子站也不是宿主站的  $x$  组成的集合，他们满足：对于子站  $Z_i$ ，他的唯一的宿主站一定是这个宿主簇中唯一的宿主站  $S_i$ 。由于对于一个宿主站和其连接的子站，在空间地理分布上呈一个簇状分布（如图 3 所示），因此为了确定这个簇和宿主站的位置，我们考虑先对给出的站点进行 AP 聚类算法。

AP 聚类算法是基于数据点之间“消息传递”概念的聚类算法。与诸如 k-means 或 k-medoids 的聚类算法不同，亲和传播不需要在运行算法之前确定或估计聚类的数量。

开始时，将所有站点都作为潜在的最佳基站点，任取一候选基站点  $x_i$ ，其所处位置为  $(\alpha_i, \beta_i)$ ，根据经纬度计算其与另一任意点  $x_j(\alpha_j, \beta_j)$  的距离，因此作为相似度的测量指标表示  $x_i$  与  $x_j$  的相似程度，  $S(i, j)$ ，且距离越远相似度越小即：

$$S(i, j) = -R \cdot \arccos[\cos\beta_i \cos\beta_j \cos(\alpha_i - \alpha_j) + \sin\beta_i \sin\beta_j]$$

同时  $S(i, j)$  也可说明  $x_j$  作为任意  $x_i$  聚类中心的合适程度。

AP 算法的最终目的是能求得能代表点簇的最佳的一点，记为  $x_k$ ，并用  $x_{k'}$  表示非  $x_k$  的任意点。引入指标吸引度  $r(i, k)$ ，表示  $x_k$  对  $x_i$  的代表程度。

则在  $x_k$  处，可用  $S(i, k)$  与除去  $x_k$  外与  $x_i$  最接近的点的合适程度之差来表示，即：

$$r(i, k) = s(i, k) - \max[a(i, k') + s(i, k')]$$

而  $r(k, k)$  说明  $k$  成为聚类中心的能力，为初始设定值。

同时，欲说明  $x_k$  为某点簇的聚类中心，需证明  $x_k$  同时对点簇处的其他点也有相当大的代表程度。引入归属度指标  $a(i, k)$  表示  $x_k$  做聚类中心的适合程度。有：

$a(k, k)$  的含义与  $r(k, k)$  相同即：

$$a(k, k) = r(k, k)$$

而对  $a(i, k)$  可表示为  $x_k$  对任意以其为聚类中心的点的吸引度和与  $r(k, k)$  之和，即：

$$a(i, k) = \begin{cases} \min\left\{0, r(k, k) + \sum_{i' \neq i} \max[0, r(i', k)]\right\} & i \neq k \\ \sum_{i' \neq i} \max[0, r(i', k)] & i = k \end{cases}$$

由  $r(i, k)$  式两端加上  $a(i, k)$ ，得：

$$r(i, k) + a(i, k) = s(i, k) + a(i, k) - \max[a(i, k') + s(i, k')]$$

可知，当  $a(i, k) + r(i, k)$  不断变化时，即  $x_k$  与  $x_i$  在不断发生变化，形成一条不断变动的信息链，而当信息链达到某一稳态时，则表明已经找到了最能代表  $x_i$  点  $x_k$ ，宏观上说，找到了最能代表某点簇的  $x_k$  点。若输入整个点阵，该算法能求出迭代一定次数信息链不

发生改变的所有最佳代表点。对应到此处，可用 AP 聚类算法求解基站分布的所有最佳代表点。

### 5.2.2 利用最大覆盖模型确定筛选宿主站

在进行过 AP 聚类算法之后，我们相当是给出了一些候选的宿主站集合，接下来我们将通过最大覆盖模型来给出相对较为合理的宿主站  $S_i$ 。最大覆盖模型是在给定一定数量的服务设施下，如何确定位置使其尽可能覆盖最多的需求点：这也符合题目中要求的尽量覆盖更多的站点。

集合  $I$  是宿主簇中子站  $Z_i$  和某些既不是子站也不是宿主站的  $x$  组成的集合，集合  $J$  表示能够宿主站  $S_i$  的集合。 $N(i)$  表示能够覆盖子站  $Z_i$  某些既不是子站也不是宿主站的  $x$  站点的宿主站  $S_j$  的集合， $N(i)=\{j|D(Z_i \text{ or } x, S_j) \leq Z-D(S_i, Z_j)\}$ ，其中  $Z-D(S_i, Z_j)$  是宿主站和子站的信号中继距离，他的值为 20km。 $W_i$  便是覆盖子站  $Z_i$  某些既不是子站也不是宿主站的  $x$  站点的需求度的系数。

模型如下：

$$\begin{aligned} \max \quad & \text{imize } z = \sum_{i \in I} W_i \cdot y_i \\ & \begin{cases} y_i \leq \sum_{j \in N(i)} x_j & i \in I \\ \sum x_j = s \\ x_j \in \{0,1\}, y_i \in \{0,1\} \end{cases} \end{aligned}$$

$s$  是 AP 聚类得到的总聚类数，也即宿主站数。

通过 AP 聚类 and 最大覆盖模型处理之后，站点的分布变化应该如图 5 所示。左边的图是 AP 聚类算法之后得到的结果，右边的图是经过 AP 聚类算法和最大覆盖模型处理之后的结果。通过最大覆盖模型处理 AP 聚类算法后的结果，我们得到了数量更为精简的宿主站  $S_i$  的集合。

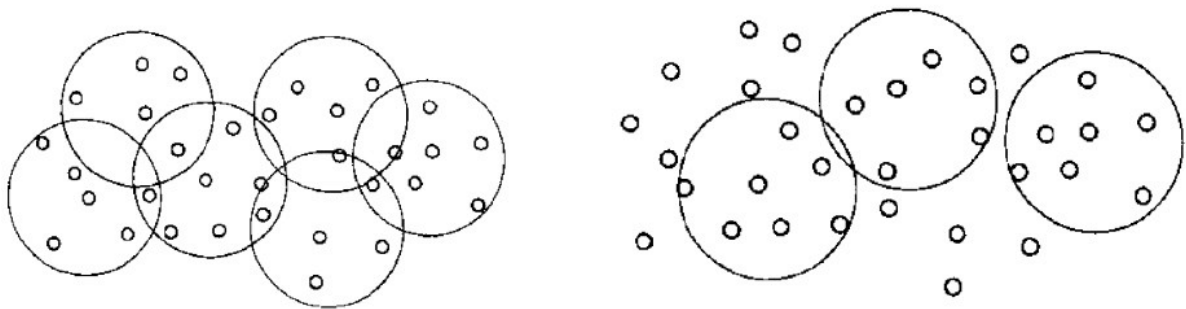


图 5 AP 聚类算法与最大覆盖模型的处理结果

### 5.2.3 运用 K-means 算法处理超出信号中继范围的宿主站

在之前考虑的几种情况中，我们并没有考虑关于宿主站之间的信号中继距离  $Z-D(S_i, S_j) \leq 50\text{km}$  的条件，在这里我们通过遍历的方法，把宿主站之间的距离超过 50km 的站点全部挑出来，并对其所在宿主簇用 K-means 聚类算法进行再次聚类，使得其满足宿主站之间的

信号中继距离  $Z-D(S_i, S_j) \leq 50\text{km}$  的条件。算法过程如下：

对于  $D(S_i, S_j) > 50\text{km}$  的宿主站  $S_i, S_j$  及其所在宿主簇  $SC_i, SC_j$  进行如下 K-means 算法进行再次拆分：

假定我们要进行聚类的  $SC_i$  的一共有  $n$  个对象，构成一个样本数据集  $X = \{x_1, x_2, \dots, x_n\}$ 。我们的目标是将这  $n$  个对象分为  $k$  个类。具体算法步骤如下：

① 先把原始数据输入进去，为  $k$  个聚类选取  $k$  个初始聚类中心，每个聚类中心记为  $Z_j(I)$ ， $j=1, 2, \dots, k$ 。

② 计算数据集  $X = \{x_1, x_2, \dots, x_n\}$  中每个数据与聚类中心的距离  $D(x_i, Z_j(I))$ ， $i=1, 2, \dots, n$ ， $j=1, 2, \dots, k$ 。现在做出如下判定：如果  $D(x_i, Z_j(I)) = \min_{j=1, 2, \dots, k} \{D(x_i, Z_j(I))\}$ ， $i=1, 2, \dots, n$ ，则  $x_i \in \omega_c$ 。

③ 将每个聚类中的样本均值作为  $k$  个新的聚类中心，即

$$Z_j(I+1) = \frac{1}{n_j} \sum_{k=1}^{n_j} x_k^j, \quad j=1, 2, \dots, k。$$

④ 现在再做一次判定，若  $Z_j(I+1) \neq Z_j(I)$ ， $j=1, 2, \dots, k$ ，则  $I=I+1$ ，返回第 ② 步；否则算法结束。

⑤ 最后得到的  $k$  个聚类中心就是我们想要的聚类结果。

在我们这个算法中， $K = \left\lfloor \frac{D(S_i, S_j)}{50} \right\rfloor + 1$ 。通过 K-means 算法，我们排除了出信号中继范围的宿主站。

在 5.1 这一节中，我们通过 AP 聚类算法与最大覆盖模型以及 K-means 聚类算法，确定了宿主站的位置。

## 5.2 确定子站

### 5.2.1 题目分析

对于子站的划分，我们要先考虑到题目里面要求的两个目标：成本最低和损耗最小。然后作为一个用来传播信号的架构，首先信号的覆盖率必须是先被首要考虑的，也就是说，对于一个给定的区域来说，整个区域的面积减去所有信号站的信号传播范围的值应该是要尽可能的小，这是一个信号架构首先需要实现的目标。因此我们现在手上就需要有三个限制条件：作为基础条件的覆盖率最高、还有题目要求的两个目标：成本最低和损耗最小。基础条件必须满足，然后配合以其他两个条件，这是一个多目标规划。我们针对这个题目开发了以下多目标规划模型：同时满足基础条件和成本最小、基础条件和损耗最小、基础条件和成本最小和损耗最小三个多目标规划。

### 5.2.2 多目标规划<sup>[3]</sup>

在实际问题中，可能会同时考虑几个方面都达到最优：成本最低，损耗最低，质量达标等等。多目标规划能很好地兼顾统筹处理多种目标的关系，得到更切合实际要求的解。多目标规划的概念是 1961 年由美国数学家查尔斯库柏首先提出的。

#### 5.2.2.1 多目标规划的数学模型

任何一个多目标规划问题，都是由两个基本的部分组成：

- (1) 两个以上的目标函数；
- (2) 若干个约束条件。

对于多目标规划问题，可以将其数学模型一般地描写为如下的形式

$$\begin{aligned} v = \min_{x \in r} & \{f_1(x), f_2(x), \dots, f_p(x)\} \\ \text{s.t.} & \begin{cases} g_j(x) \leq 0 & j = 1, 2, \dots, m \\ h_k(x) = 0 & k = 1, 2, \dots, l \end{cases} \end{aligned}$$

函数  $f_i, g_j, h_k$  满足

$$f_i: r^n \rightarrow r, g_j: r^n \rightarrow r, h_k: r^n \rightarrow r, p \geq 2$$

求目标函数的最大值或约束条件为大于等于零的情况，都可以通过取其相反数得到。

### 5.2.2.2 多目标规划解的定义

对于多目标规划，解的定义是一个非常重要的问题。文献中给出了多目标规划的基本定义和定理：

定义 6.1(绝对最优解) 设  $x^* \in r$ ，如果对任意的  $x \in r$ ，均有  $f(x^*) \leq f(x)$ ，即有  $f_i(x^*) \leq f_i(x) \quad i=1, 2, \dots, p$ ，则说  $x^*$  是 (V) 的绝对最优解，其全体记为  $r_{ab}$ ；

定义 6.2(有效解，也称 pareto 最优解) 设  $x^* \in r$ ，如果不存在  $x \in r$ ，使得  $f(x^*) \leq f(x)$ ，则说  $x^*$  是 (V) 的有效解，也叫 pareto 最优解，其全体记为  $r_{pa}$ 。

定义 6.3(弱有效解) 设  $x^* \in r$ ，如果不存在  $x \in r$ ，使得  $f(x^*) < f(x)$ ，则说  $x^*$  是 (V) 的弱有效解，其全体记为  $r_{ba}$

### 5.2.2.3 多目标规划的基本解法

对于多目标规划的解法，数学家们对此研究了很多也提出了很多有用的方法，此处就不全部列举，仅指出本题需使用的多目标规划方法：线性加权法

线性加权法是指事先根据目标函数  $f_1(X), f_2(X), \dots, f_p(X)$  的重要程度给出一组权系数  $\lambda$ ，满足， $\lambda > 0, \sum \lambda_i = 1$ ，然后定义评价函数：

$$h(f(x)) = \sum_{i=1}^p \lambda_i f_i(x)$$

求解非线性规划问题得到的最优解称为多目标规划的有效解

### 5.2.3 一些参数

$R_1$  : 宿主站的平均半径

$R_2$  : 子站的平均半径

$M$  : 宿主站的数量

$N$  : 子站的数量

$S$  : 整个需要覆盖范围的面积

$S_i$  : 第  $i$  个主站的覆盖面积

$X_i$  : 第  $i$  个子站覆盖的面积

$\theta$  : 子站与宿主站覆盖面积重合率

### 5.2.4 成本最低

在确定子站的分布之前，我们先考虑题目中要求的成本最低的挑战目标，这会成为我们规划子站顺序的一个非常重要的限制条件。

成本最低的模型如下：

宿主站数量为  $m$ ，子站数量为  $n$ ，我们可以得到第一个目标：

$$\min \quad c = 10m + 5n + \left\lceil \frac{m}{8} \right\rceil$$

### 5.2.5 损耗最低

我们由公式可知模型如下：



$$PL_i = 32.5 + 20 \lg D + 20 \lg H$$

$$D = mR_1 + nR_2$$

$$\min \sum PL_i$$

### 5.2.6 覆盖率最大

在此基础上，我们还得满足子站和宿主站在整个区域上未被覆盖的  $s$  趋于小，我们可以得到第二个目标：

$$\min s = s - \sum_n^m si - \sum_1^n xi + \sum_1^l \theta si xi$$

### 5.2.7 多目标规划三个模型

我们提出的无限回传拓扑要解决两个问题，第一个就是选定子站和主站的数量，第二个就是确定子站和主站的连接方式

则目标函数为：

$$H(f(x)) = (minc) + (mins)$$

对于其覆盖率有目标 2 中的目标函数

$$\min s = s - \sum_n^m si - \sum_1^n xi + \sum_1^l \theta si xi$$

由多目标规划的算法我们可以得出其最小值，由此可以得到下列覆盖率公式

$$\delta = \frac{\sum_n^m si + \sum_1^n xi - \sum_1^l \theta si xi}{s}$$

当且仅当上述的目标函数取得最小值时，覆盖率取得最大

## 5.3 算法效率优化

### 5.3.1 对于前两节建立的算法的思考

无可置疑的是，前面模型显然是 NP-Hard 问题，如果强行进行求解，那么算法的效率将会是极其低下的，于是我们考虑采用一些智能算法实现。我们主要考察的算法有：粒子群算法、模拟退火算法和蚁群算法。

### 5.3.2 粒子群算法

粒子群是模拟群鸟觅食的模型，它在寻找全局最优的应用上具有十分广泛的作用。在群鸟觅食的模型中。我们把每一只鸟儿都看作一个粒子，那么整个鸟群可以被看成一个粒子群。设在某一个  $D$  维的目标搜索空间中，有  $m$  个粒子所组成的粒子群，其中第  $i$  ( $i=1,2, \dots, m$ ) 个粒子正处于位置  $x_i = (x_i^1, x_i^2, \dots, x_i^D)$ ，这个位置可能正好是一个最优解，将  $x_i$  带入目标函数就可以得出其适应值，通过适应值的大小可以判断出这个位置的优劣。假设个体粒子经过的最好位置是  $P_i = (p_i^1, p_i^2, \dots, p_i^D)$ ，整体粒子群所经过最好的位置是  $P_g = (p_g^1, p_g^2, \dots, p_g^D)$ 。第  $i$  粒子的速度记为  $S_i = (s_i^1, s_i^2, \dots, s_i^D)$ 。

在粒子群算法运行中，用以下公式对粒子的位置进行不断的更新：

$$s_i^d = \omega s_i^d + c_1 r_1 (p_i^d - x_i^d) + c_2 r_2 (p_g^d - x_i^d)$$

$$x_i^d = x_i^d + \alpha s_i^d$$

其中， $i=1, 2, \dots, m$ ;  $d=1,2, \dots, D$ ; 惯性因子  $\omega \geq 0$ ；加速常数  $c_1, c_2 \geq 0$ ； $r_1, r_2$  是在  $[0,1]$  内变化的随机数；约束因子  $\alpha$  控制速度的权重； $s_i^d \in [-s_{\max}^d, s_{\max}^d]$ ，表明粒子的速度被最大速度限制，这个最大速度可以人为设定。最后，粒子群位置的改变，也即迭代次

数可以被人为设定。

### 5.3.3 模拟退火算法

模拟退火算法是一种通用的概率算法，用来在很大的搜索空间中寻找问题的最优解。算法的基本步骤一般如下：

① 随机挑选一个单元  $k$ ，并给他一个随机位移，并求出系统以为有而产生的能量变化  $\Delta E_k$ 。

② 若  $\Delta E_k \leq 0$ ，该位移可以被采纳，而变化后的系统转态作为下次变化的起点。若  $\Delta E_k > 0$ ，位移后的状态可采纳概率为：

$$P_k = \frac{1}{1 + e^{-\Delta E_k/T}}$$

$T$  为温度，然后从  $(0, 1)$  区间均匀分布的随机出中挑选一个数  $R$ ，若  $R < P_k$ ，则将变化后的状态作为下次的起点；否则，将变化前的状态作为下次的起点。

③ 转第 ① 步继续执行，知道达到平衡状态为止。

算法的流程图如下：

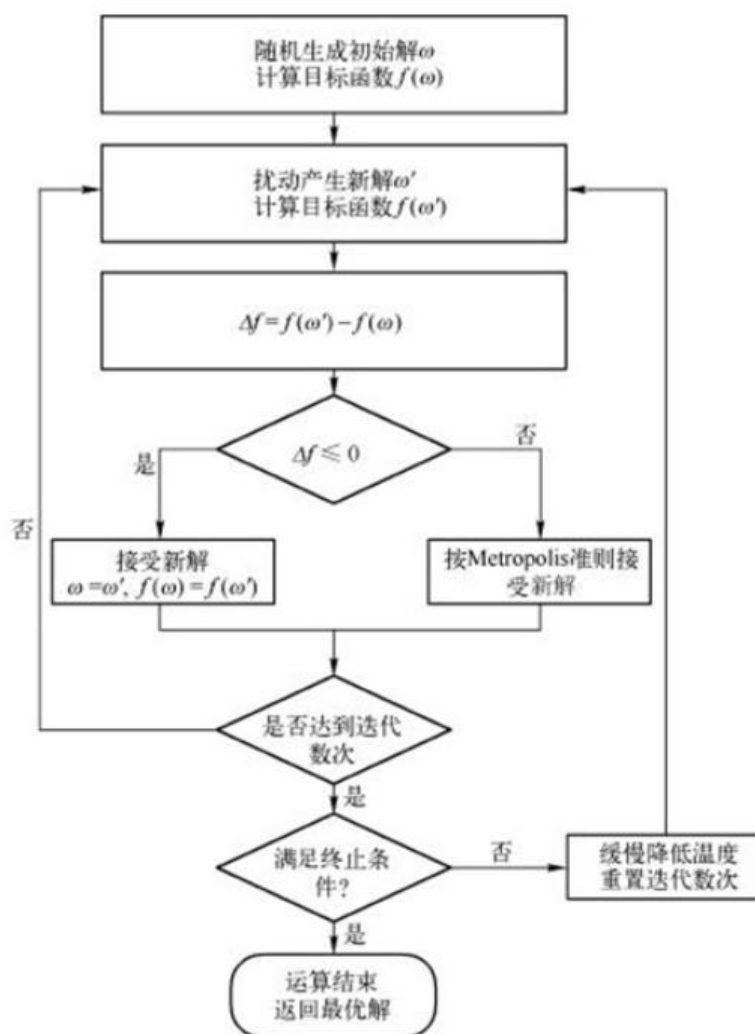


图 6 模拟退火算法流程

#### 5.3.4 蚁群算法

蚁群算法的基本原理来源于自然界蚂蚁觅食的最短路径原理，通过一种正反馈机制或称增强型学习系统；它通过“最优路径上蚂蚁数量的增加→信息素强度增加→后来蚂蚁选择概率增大→最优路径上蚂蚁数量更大增加”的过程来达到最终收敛于最优解上。该算法的流程图如下：

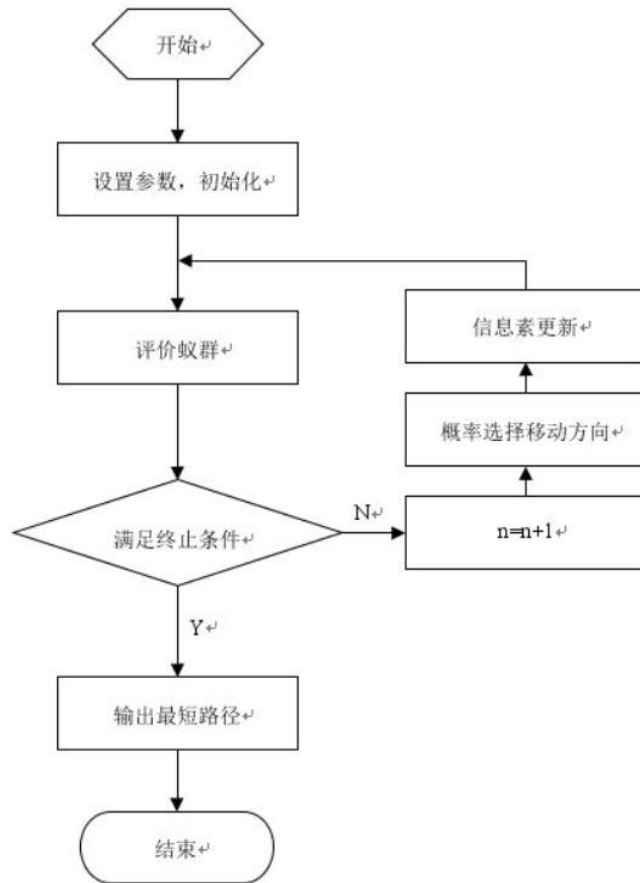


图 7 蚁群算法流程图

#### 5.3.2 几大智能算法的考察结果

非常遗憾的是由于时间的限制，我们只完成了粒子群的算法的编程。并不能按照预先的设想，根据的样本的数量给出

## 六、 模型检验

对于我们前面建立的多目标优化是一个 NP-Hard 问题，我们在确认子站的时候重新考虑了穷举算法，对比两个算法的结果。

## 七、 模型评价

### 7.1 模型的优点

1、我们采用了 ap 聚类、粒子群算法等等算法可以精确保证宿主站点的准确性以及

其覆盖率

2、多目标规划结合运筹学中经典理论，为决策者提供了最符合条件的子站选择，最大程度的为农村和城市之间的无线拓扑回传提供便利

3、在球面上进行精准的坐标计算

## 7.2 模型的缺点

1. 对子站的跳数有所忽略，这是模型最大的疏忽

2. 多目标规划的约束条件过于复杂，变量过多

3. 对于第二个目标可能存在区域最优解

## 八、 参考引文

【1】李文杰，基于 ap 聚类 and 集合覆盖模型的农电营业区域电费缴纳点选址研究，数学的实践和认识 2018 年 4 月

【2】王美玲，基于 ap 密度聚类方法的雷法辐射源信号识别，舰船电子对抗，2012 年 6 月

【3】万凤娇，基于多目标规划的危险废弃物物流选址-选线模型研究，武汉理工大学博士学位论文，2010 年 5 月

【4】甘月松，一种 ap 算法的改进:M-AP 聚类算法，计算机科学，2015 年 1 月

【5】郭秀娟，ap 聚类算法的分析和应用，吉林建筑大学学报，2013 年 8 月第 30 卷第四期

## 九、 代码附录

### 1. AP 聚类算法

```
from sklearn.cluster import AffinityPropagation
from sklearn import metrics
from sklearn.datasets.samples_generator import make_blobs
# 生成测试数据
centers = [[1, 1], [-1, -1], [1, -1]]
# 生成实际中心为 centers 的测试样本 300 个，X 是包含 300 个
(x, y) 点的二维数组，labels_true 为其对应的真是类别标签
X, labels_true = make_blobs(n_samples=300, centers=centers,
cluster_std=0.5,
                                random_state=0)

# 计算 AP
ap = AffinityPropagation(preference=-50).fit(X)
cluster_centers_indices = ap.cluster_centers_indices_
# 预测出的中心点的索引，如[123, 23, 34]
labels = ap.labels_ # 预测出的每个数据的类别标
签, labels 是一个 NumPy 数组

n_clusters_ = len(cluster_centers_indices) # 预测聚类
```

## 中心的个数

```
print('预测的聚类中心个数: %d' % n_clusters_)
print('    同    质    性    :    %0.3f' %
      metrics.homogeneity_score(labels_true, labels))
print('    完    整    性    :    %0.3f' %
      metrics.completeness_score(labels_true, labels))
print('V-    值    :    %    0.3f' %
      metrics.v_measure_score(labels_true, labels))
print('调  整  后  的  兰  德  指  数  :    %0.3f' %
      metrics.adjusted_rand_score(labels_true, labels))
print('调  整  后  的  互  信  息  :    %0.3f' %
      metrics.adjusted_mutual_info_score(labels_true, labels))
print('轮廓系数: %0.3f' % metrics.silhouette_score(X,
      labels, metric='sqeuclidean'))
```

## # 绘制图表展示

```
import matplotlib.pyplot as plt
from itertools import cycle
```

```
plt.close('all')    # 关闭所有的图形
plt.figure(1)    # 产生一个新的图形
plt.clf()    # 清空当前的图形
```

```
colors = cycle('bgrcmykbgrcmykbgrcmykbgrcmyk')
```

```
# 循环为每个类标记不同的颜色
```

```
for k, col in zip(range(n_clusters_), colors):
```

```
    # labels == k 使用 k 与 labels 数组中的每个值进行比较
```

```
    # 如 labels = [1, 0], k=0, 则 'labels==k' 的结果为[False,
    True]
```

```
    class_members = labels == k
```

```
    cluster_center = X[cluster_centers_indices[k]]    #
```

```
    聚类中心的坐标
```

```
    plt.plot(X[class_members, 0], X[class_members, 1], col
    + '.')
```

```
    plt.plot(cluster_center[0],    cluster_center[1],
    markerfacecolor=col,
```

```
            markeredgecolor='k', markersize=14)
```

```

        for x in X[class_members]:
            plt.plot([cluster_center[0], x[0]],
                    [cluster_center[1], x[1]], col)

```

```

plt.title('预测聚类中心个数: %d' % n_clusters_)
plt.show()``

```

## 2. K-means 聚类算法

%% kMeans 的核心程序，不断迭代求解聚类中心

```
function [ centroids ] = kMeans( dataSet, k )
```

```
    [m,n] = size(dataSet);
```

```
    %初始化聚类中心
```

```
    centroids = randCent(dataSet, k);
```

```
    subCenter = zeros(m, 2); %做一个 m*2 的矩阵，第一列存储类别，第二列存储距离
```

```
    change = 1; %判断是否改变
```

```
    while change == 1
```

```
        change = 0;
```

```
        %对每一组数据计算距离
```

```
        for i = 1:m
```

```
            minDist = inf;
```

```
            minIndex = 0;
```

```
            for j = 1:k
```

```
                dist= distance(dataSet(i, :),
```

```
centroids(j, :));
```

```
                if dist < minDist
```

```
                    minDist = dist;
```

```
                    minIndex = j;
```

```
                end
```

```
            end
```

```
            if subCenter(i, 1) ~= minIndex
```

```
                change = 1;
```

```
                subCenter(i, :)= [minIndex, minDist];
```

```
            end
```

```
        end
```

```
        %对 k 类重新计算聚类中心
```

```
        for j = 1:k
```

```
            sum = zeros(1, n);
```

```

        r = 0;%数量
        for i = 1:m
            if subCenter(i,1) == j
                sum = sum + dataSet(i,:);
                r = r+1;
            end
        end
        centroids(j,:) = sum./r;
    end
end

%% 完成作图
hold on
for i = 1:m
    switch subCenter(i,1)
        case 1
            plot(dataSet(i,1), dataSet(i,2), '.b');
        case 2
            plot(dataSet(i,1), dataSet(i,2), '.g');
        case 3
            plot(dataSet(i,1), dataSet(i,2), '.r');
        otherwise
            plot(dataSet(i,1), dataSet(i,2), '.c');
        end
    end
end
plot(centroids(:,1), centroids(:,2), '+k');
end
3. 粒子群算法
close all;
clear all;

```

PopSize=500;%种群大小

CityNum = 14;%城市数

OldBestFitness=0;%旧的最优适应度值

Iteration=0;%迭代次数

MaxIteration =2000;%最大迭代次数

```

IsStop=0;%程序停止标志
Num=0;%取得相同适应度值的迭代次数

c1=0.5;%认知系数
c2=0.7;%社会学习系数
w=0.96-Iteration/MaxIteration;%惯性系数,随迭代次数增加而递减

%节点坐标
node=[16.47 96.10; 16.47 94.44; 20.09 92.54; 22.39 93.37; 25.23
97.24;...
      22.00 96.05; 20.47 97.02; 17.20 96.29; 16.30 97.38; 14.05
98.12;...
      16.53 97.38; 21.52 95.59; 19.41 97.13; 20.09 94.55];

%初始化各粒子,即产生路径种群
Group=ones(CityNum,PopSize);
for i=1:PopSize
    Group(:,i)=randperm(CityNum)';
end
Group=Arrange(Group);

%初始化粒子速度(即交换序)
Velocity=zeros(CityNum,PopSize);
for i=1:PopSize
    Velocity(:,i)=round(rand(1, CityNum) '*CityNum); %round 取整
end

%计算每个城市之间的距离
CityBetweenDistance=zeros(CityNum, CityNum);
for i=1:CityNum
    for j=1:CityNum

CityBetweenDistance(i, j)=sqrt((node(i, 1)-node(j, 1))^2+(node(i, 2)
-node(j, 2))^2);
    end
end

%计算每条路径的距离

```



```

for i=1:PopSize
    EachPathDis(i) =
PathDistance(Group(:, i)', CityBetweenDistance);
end

IndividualBest=Group;%记录各粒子的个体极值点位置, 即个体找到的最短
路径
IndividualBestFitness=EachPathDis;%记录最佳适应度值, 即个体找到的
最短路径的长度
[GlobalBestFitness, index]=min(EachPathDis);%找出全局最优值和相应
序号

%初始随机解
figure;
subplot(2, 2, 1);
PathPlot(node, CityNum, index, IndividualBest);
title('随机解');

%寻优
while(IsStop == 0) & (Iteration < MaxIteration)
    %迭代次数递增
    Iteration = Iteration +1;

    %更新全局极值点位置, 这里指路径
    for i=1:PopSize
        GlobalBest(:, i) = Group(:, index);
    end

    %求 pij-xij , pgj-xij 交换序, 并以概率 c1, c2 的保留交换序
    pij_xij=GenerateChangeNums(Group, IndividualBest);
    pij_xij=HoldByOdds(pij_xij, c1);
    pgj_xij=GenerateChangeNums(Group, GlobalBest);
    pgj_xij=HoldByOdds(pgj_xij, c2);

    %以概率 w 保留上一代交换序
    Velocity=HoldByOdds(Velocity, w);

```

```

        Group = PathExchange(Group, Velocity); %根据交换序进行路径交换
        Group = PathExchange(Group, pij_xij);
        Group = PathExchange(Group, pgj_xij);
        for i = 1:PopSize % 更新各路径总距离
            EachPathDis(i) =
PathDistance(Group(:, i)', CityBetweenDistance);

        end

        IsChange = EachPathDis < IndividualBestFitness; %更新后的距离优于
更新前的, 记录序号
        IndividualBest(:, find(IsChange)) = Group(:, find(IsChange)); %
更新个体最佳路径
        IndividualBestFitness = IndividualBestFitness.*( ~IsChange) +
EachPathDis.*IsChange; %更新个体最佳路径距离
        [GlobalBestFitness, index] = min(EachPathDis); %更新全局最佳路
径, 记录相应的序号

        if GlobalBestFitness==OldBestFitness %比较更新前和更新后的适
应度值;
            Num=Num+1; %相等时记录加一;
        else
            OldBestFitness=GlobalBestFitness; %不相等时更新适应度值,
并记录清零;
            Num=0;
        end
        if Num >= 20 %多次迭代的适应度值相近时程序停止
            IsStop=1;
        end

        BestFitness(Iteration) =GlobalBestFitness; %每一代的最优适应
度

    end

%最优解
subplot(2, 2, 2);

```

```
PathPlot(node, CityNum, index, IndividualBest);  
title(' 优化解');  
%进化曲线  
subplot(2, 2, 3);  
plot((1:Iteration), BestFitness(1:Iteration));  
grid on;  
title(' 进化曲线');  
%最小路径值  
GlobalBestFitness
```