



参赛密码 _____

(由组委会填写)



“华为杯”第十四届中国研究生 数学建模竞赛

学 校 中国石油大学（华东）

参赛队号 10425002

队员姓名 1.吕飞

2.魏佳

3.魏志毅

参赛密码 _____

(由组委会填写)



“华为杯”第十四届中国研究生 数学建模竞赛

题目 无人机在抢险救灾中的优化运用

摘 要：

针对灾后救援过程中如何利用无人机高效完成灾害巡查、生命探测、中继通讯、数据传输等问题，建立了多种规划模型，优化了包括无人机飞行时间、无人机数量以及巡查面积覆盖率等多种指标，实现了无人机资源的合理分配和有效利用。

问题一针对灾情巡查的优化问题，首先对目标区域进行网格划分，建立了提高巡查面积覆盖率和降低无人机数量的多目标规划模型，通过分析将问题等效为求解能够巡查更多点的最短路径问题，通过遗传算法、贪心算法等一系列方法进行求解，并对模型参数进行优化，最终得到同时满足多目标优化的无人机的飞行时间、路线，在初次巡查中覆盖率为 52.85%，共使用 6 架无人机；在次生灾害巡查中在覆盖率达到 66.18%。的情况下共使用 30 架飞机。该模型综合考虑了观察仰角、视野阻碍、山体阻碍路线等约束条件的影响并进行了数学描述。

问题二要求生命迹象探测中降低从第一架无人机飞出到最后一架完成任务的无人机回到基地的时间间隔。将问题规划为最大值最小化问题，并给出合理的约束条件。无人机分配任务是在聚类分析的基础上进行搜索优化，最终得到最短

的任务执行时间为 5.6 小时。

问题三要求设计太阳能无人机的飞机能够始终维持地面移动终端的网络通讯，并使用尽可能少的无人机。为此设计了静态布局模型，通过聚类分析减少地面通讯无人机的数量，并将中继无人机数量与网络长度联系起来，将问题简化为求解最短网络路径的最小生成树问题，最终得到最少无人机数是 101 架。

问题四通过敏感性分析，将提高无人机同时传输用户数作为重要的优化目标，同时设计功率的策略为平均分配。而后问题模型依靠聚类分析、枚举法和遗传算法等降低了无人机的使用架次，从而优化了任务时间总和为 4.41h。

将多种有关无人机轨迹的问题转化为可以由算法处理的规划问题，并综合全面的考虑了约束条件和模型参数的影响，得到了优化的、可实现的、满足题目要求的解决方案和计算结果，降低了救灾过程中的人力物力消耗，实现了无人机的最大效能，合理的解决了飞机续航能力限制、飞机侦察和传输能力限制、地形条件影响对无人机执行任务造成的困难，具有重要的现实意义。

关键词：无人机任务规划；旅行商问题；遗传算法；最小生成树；最大值最小化问题

1 问题重述

1.1 问题背景

无人机作为一种新型运载工具，能够在地震救援行动中发挥重要作用，尤其是在交通道路设施毁坏严重、天气条件恶劣的情况下，外界救援部队无法立即到达救助被困人员时，给灾区被困人员投送水、食物、药品等必要物资；同时，使用无人机携带生命探测仪搜索生命迹象，能够给灾后救援提供准确的目标定位，也带回了大量的灾区现场数据资料，及时高效抢救灾区人民群众生命，减少地震损失的^[1]。

1.2 问题分析

根据震区情况及无人机资料，为充分发挥无人机使用效率，顺利完成抢险救灾任务，对无人机灾情巡查、生命迹象探测、灾区通信中继、无人机对地的数据传输问题进行优化与研究。

(1) 要求无人机在 4 小时之内使区域 S 内海拔 3000m 以下的地方尽可能多地被巡查到，制定巡查方案，确定无人机的架数、覆盖率，设计每架无人机的飞行路线；其次，无人机高度低于 4000 米的区域（不限于 S ）上空巡逻，问最少需要多少架、制定巡逻方案（包括无人机架数、每架无人机飞行时间、路线）保证其在 72 小时内，上述被巡查到的地方相邻两次被巡查的时间间隔不大于 3 小时。

(2) 规划无人机执行生命探测路线，使从第一架无人机飞出到最后一架完成任务的无人机回到基地的时间间隔尽量短，且全区域内海拔 3000 米以下部分能被探测到的面积尽可能大。

(3) 规划无人机飞行路线及方案保证在白天 12 小时内，任意两个地面终端之间都能实现不间断通信。

(4) 为无人机设计恰当的航线、速度以及所服务的用户，并为每一个用户分配恰当的功率，使得无人机完成对地传输数据所有任务的时间总和尽量短。

2 模型假设

①假设无人机最大续航时间为 8 小时，与其它障碍物(含地面)的安全飞行距离不小于 50 米，最大飞行高度为海拔 5000 米。

②所有无人机均按规划好的航路自主飞行，无须人工控制，完成任务后自动返回原基地。

③假设无人机主要以 60km/h 匀速直线运动，忽略转弯引起的速度变化和轨迹变化。

④多架飞机可以同时出发，飞行轨迹不会相互影响。

⑤飞机从基地出发可以瞬时加速到 60km/h，无视加速过程的影响。

⑥飞行时的转弯半径不小于 100 米，转弯的影响可以忽略不计。太阳能无人机能够以 100 米半径进行盘旋，在天明时就可以部署到指定位置。

⑦地面移动终端可以在起始位置半径 2km 内随意移动。

⑧无人机的飞行和信号传输不会受天气影响，地形和地面设施不会产生电磁干扰，无人机不会出现故障、迫降和损坏等问题。

3 符号说明

表 3-1 变量符号含义说明

符合	说明
i	无人机的数量
S_n	每架飞机巡查的面积, km^2
S_t	总的目标面积, km^2
S_n'	所有在无人机巡查范围外的目标面积, km^2
S_n''	所有被地形阻挡的目标面积, km^2
v	无人机速度, km/h
t_k	第 k 架飞机的任务完成时间, h
C	无人机扫描面积覆盖率
N_{mo}	山体的编号
$N(x_0)$	路径与山体的交点
\overrightarrow{XW}	地面到无人机的向量
i_1	地面通讯无人机的数量
i_2	中继无人机的数量
G	聚类分析所分类数
h_i	地面通讯无人机间的距离, km
$w(T)$	生成树的权值
B_i	无人机服务 i 的子信道带宽, Hz
$p_i(t)$	t 时刻无人机为第 i 个地面用户所在的子信道分配的功率, W
$d(u, i)$	表示 t 时刻无人机与 i 之间的欧氏距离, m
ρ_0	信道特性参数

4 问题一模型建立与求解

4.1 问题描述及分析

为及时了解灾区情况，合理制定救援方案，使用无人机携带视频采集装置巡查 7 个重点灾区区域中心方圆 10 公里(并集记为 S)以内的灾情。希望无人机从基地执行任务，在 4 小时之内使区域 S 内海拔 3000 米以下的地方尽可能多地被巡查到，给出救援方案及飞行路线图；为及时发现次生灾害，使用无人机在附件 1 给出的高度低于 4000 米的区域（不限于 S ）上空巡逻，最少需要多少架无人机、如何设定每架无人机的飞行时间、路线，才能保证在 72 小时内，上述被巡查到的地方相邻两次被巡查的时间间隔不大于 3 小时。

结合问题概述，为了解决上述问题，需要满足以下任务：

(1) 灾情巡查要求在规定的时间内完成，因此需要合理分配无人机架次和飞行路线，使得无人机的巡查能力得到有效发挥，以便在指定时间内使目标区域被以尽可能多地巡查到。

(2) 在保证一定巡查目标覆盖率的情况下，尽可能减少无人机的出勤架次，且规划的路线不能被山脉阻挡。

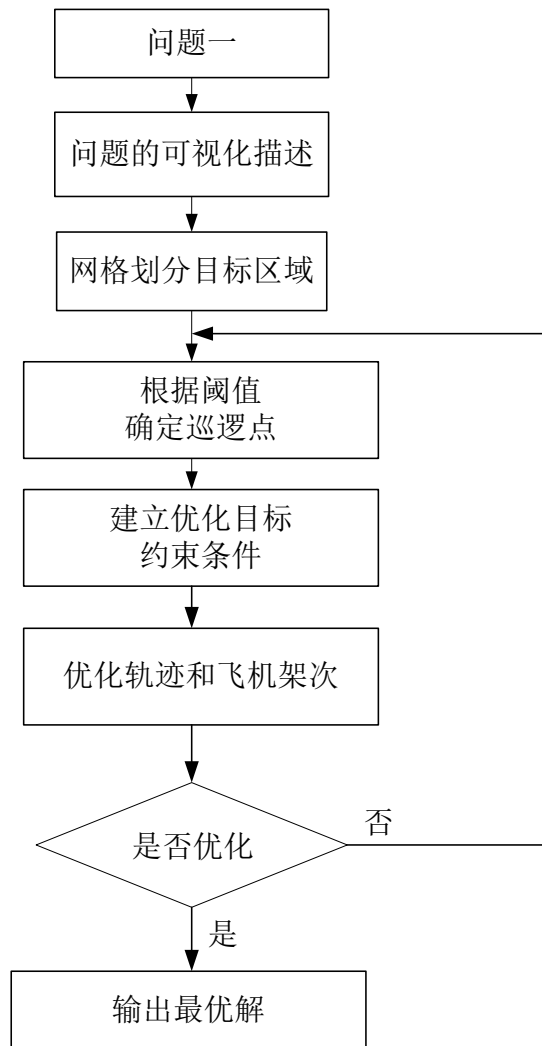


图 4-1 问题一求解思路及算法

4.1.1 3000m 以下灾情巡查分析

建模前，对原始数据进行预处理，绘制灾区情况示意图，如图 4-2 所示。

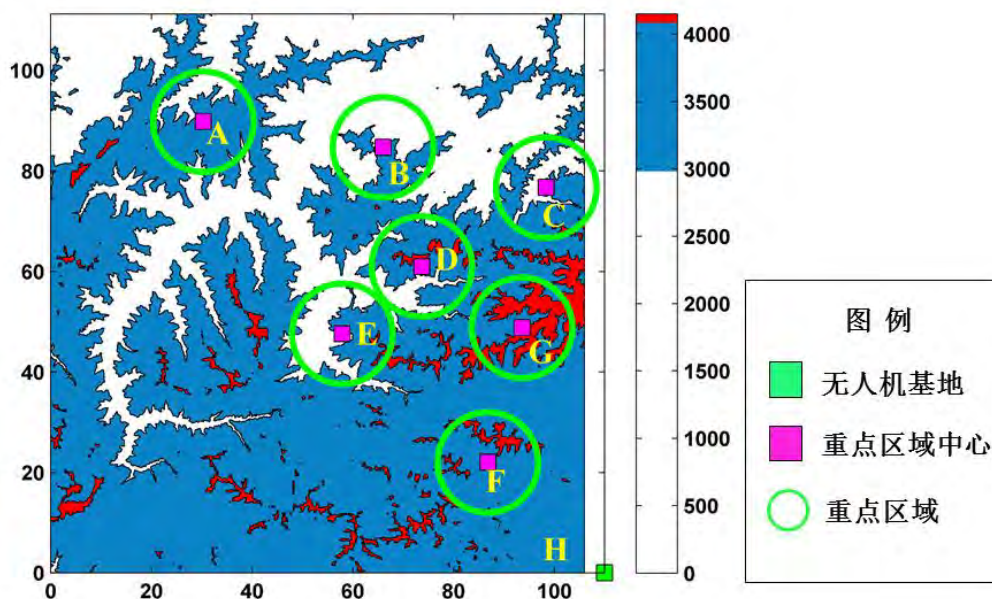


图 4-2 灾区情况示意图

第一步：网格划分

为了更好的掌握巡查重点 S 内的目标区域分布情况，进行网格划分，采用一定尺寸的正方形网格分割目标区域（如图 4-3 所示），以正方形中心作为参考点，计算正方形网格内可以被参考点上方无人机观测到的面积。如果目标观测覆盖率超过设定阈值，则该点即为高价值目标，可将正方形中心视为巡逻点，将会在后继的无人机轨迹设计阶段强制要求经过该点巡查。

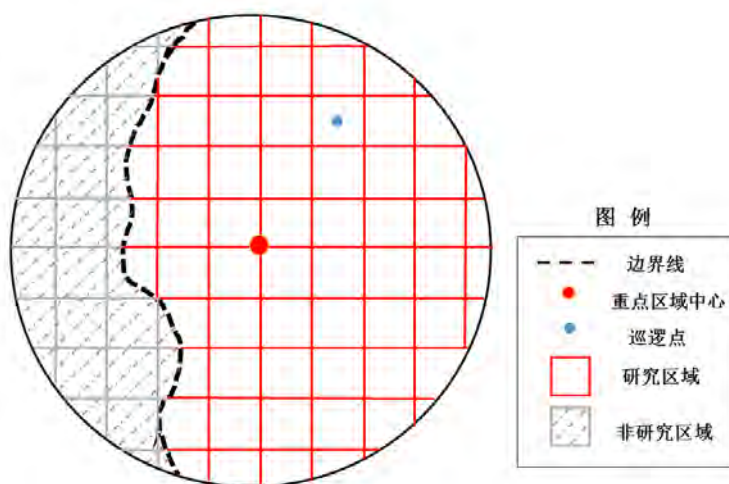


图 4-3 重点区域网格划分示意图

题中要求一个点可被视为侦查的条件是在地面某点看无人机的仰角大于 60° 且视线不被山体阻隔（如图 4-4 所示）。为了达到目标条件，需满足以下两个条件：

- (1) 要求该点与无人机的垂向高度差和两点间横向距离的比值大于 $\tan 60^\circ$ 。



图 4-4 观测点观测无人机示意图

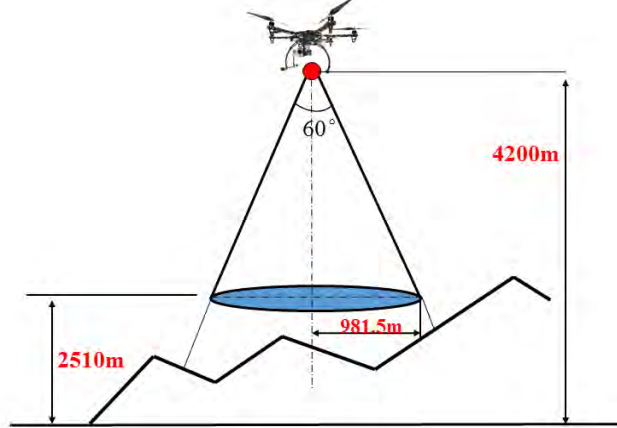


图 4-5 无人机巡查扫描区域示意图

由图 4-5 可知，扫描区域的尺寸与无人机相对于地面的高度呈几何关系，通过条件格式筛选出所有高度低于 3000m 的数据点并求平均值，可得目标区域的平均高度约为 2510m。由几何关系可得正方形边长为 1963m。

(2) 要求计算无人机是否满足在视野范围内。因此需要通过中心差分法计算关于高度的偏导数 $\frac{\partial z}{\partial x}$ 、 $\frac{\partial z}{\partial y}$ ，则近似球面在该点切线的法向量为 $\left(-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1\right)$ 。如果该法向量和该点与无人机连线的向量的向量积大于 0，则说明连线在地层的近似球面之外，不会被障碍物阻挡（如图 4-6 所示）。

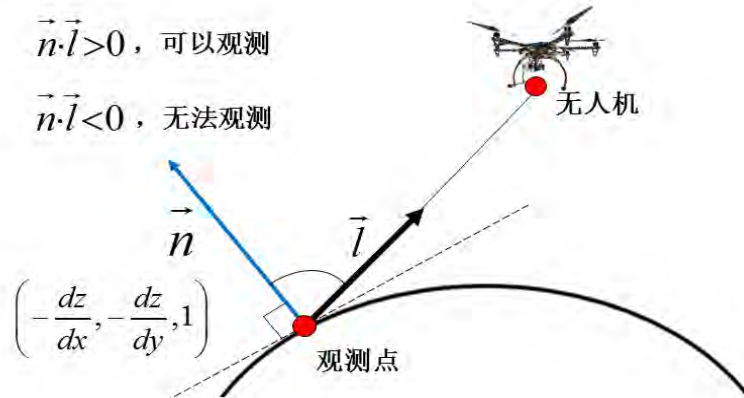


图 4-6 无人机视野观测情况判断示意图

该数学模型的设计目标是找到所有目标区域观测覆盖率大于阈值的巡逻点，约束条件包括：

- (1) 可观测数据点看无人机的仰角大于 60° ，中间没有视野阻碍；
- (2) 可观测数据点海拔高度小于 3000 米。

经过计算，得到图 4-7 所示的巡逻点集合分布。

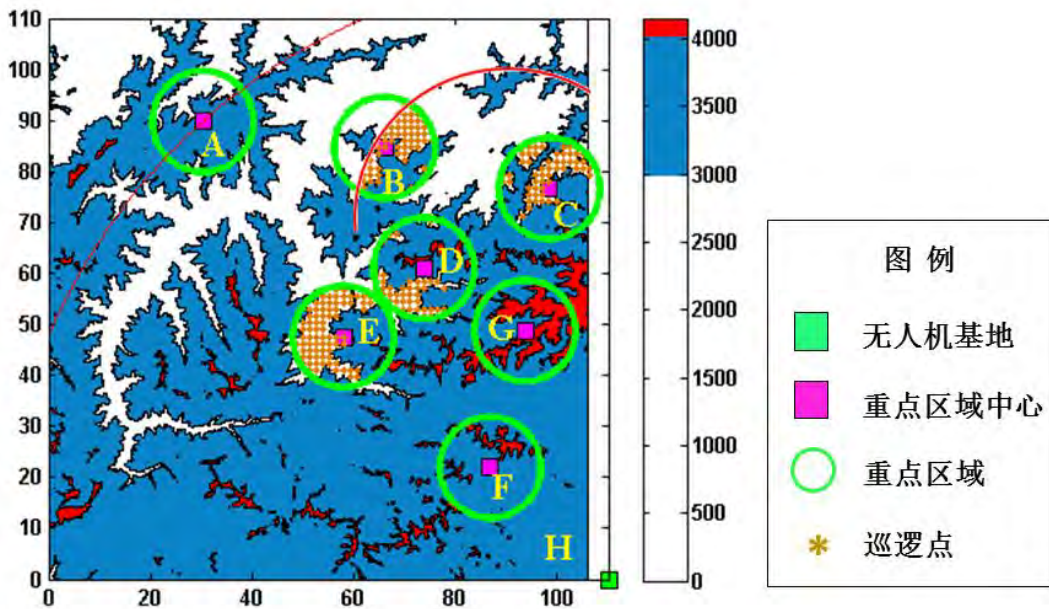


图 4-7 目标巡查区域巡逻点图

由图 4-7 可以看出，所控制的巡查面积尽可能覆盖了 3000m 以下的目标巡查区域，由于中心点 A 在 4 小时飞机航程以外，中心点 F、G 没有有价值目标，所以这三个区域放弃巡查。

第二步：轨迹优化

在给出任务所需要的巡逻点之后，需要安排无人机的数量和飞行路线。由于任务区域被分割成 7 个圈闭区域内，所以巡逻点以所在重点中心划分类群，每个类群根据任务量和时间约束条件（4 小时）安排一架或多架飞机。

山体阻隔同样是问题的一个重要约束条件。无人机的飞行高度为 4200m，无人机与地形的安全距离为 50m，因此所有高于 4150m 的地形都会阻碍无人机的飞行路线。为了在算法中实现对阻碍山体的数学描述，将主要山体简化为直线段，并在山体一侧设立迂回点，如图 4-8 所示。在轨迹优化建立距离矩阵的过程中，要求任何两点间线段如果和山脉交叉，则将两点距离设为无穷大。如果路径想要越过山体，便会从迂回点周转动。通过枚举试验测试迂回点位置，使得到达地图中每一个点的路径尽可能短，最终迂回点选择(65,35)，(110,60)。由于迂回点的周转，使得无人机的航程进一步缩短，区域 B 的一部分点超出航程，所以将其从巡查任务中舍去（图 4-7）。

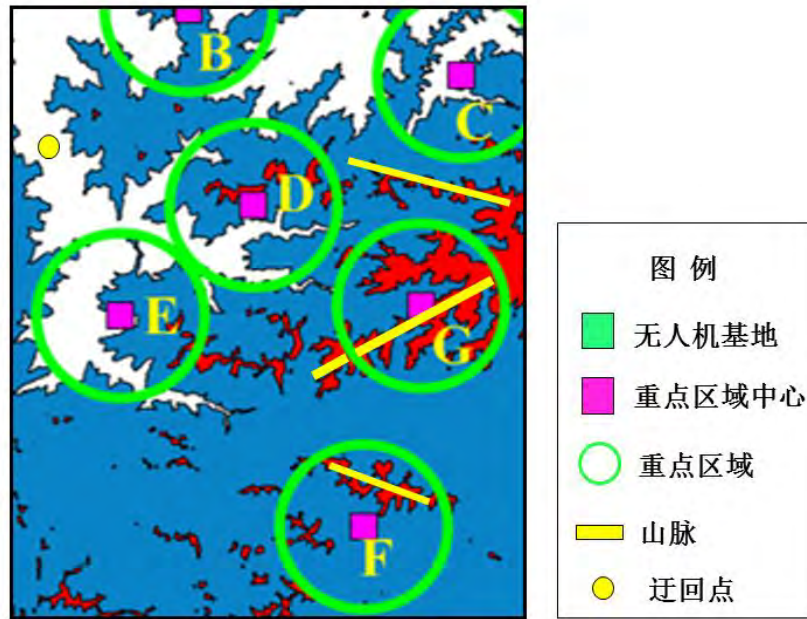


图 4-8 山脉数学描述示意图

4.1.2 次生灾害巡查分析

次生灾害巡查的分析方法和初步巡查大致相同，都需要网格划分和轨迹优化。网格划分的结果如图 4-9 所示。最重要的三个约束条件变化是：飞机对全局的 4000m 以下点进行巡查；飞机的续航时间变为 8 小时；要求在 72 小时之内每个巡逻点每三小时会被巡查一次。采取策略让无人机尽可能 8 小时满载飞行。一方面，通过满载每架无人机的巡逻点数量使得任务被分配到更少的无人机上，有助于减少无人机数量；另一方面，无人机需要 8 小时巡航加 1 小时基地维护，另一方面，所有巡逻点要求 3 小时巡逻一次。每架飞机相隔 3 小时出发，以 8 小时路程划分的任务区正好可够 3 架无人机无缝对接保持巡逻。

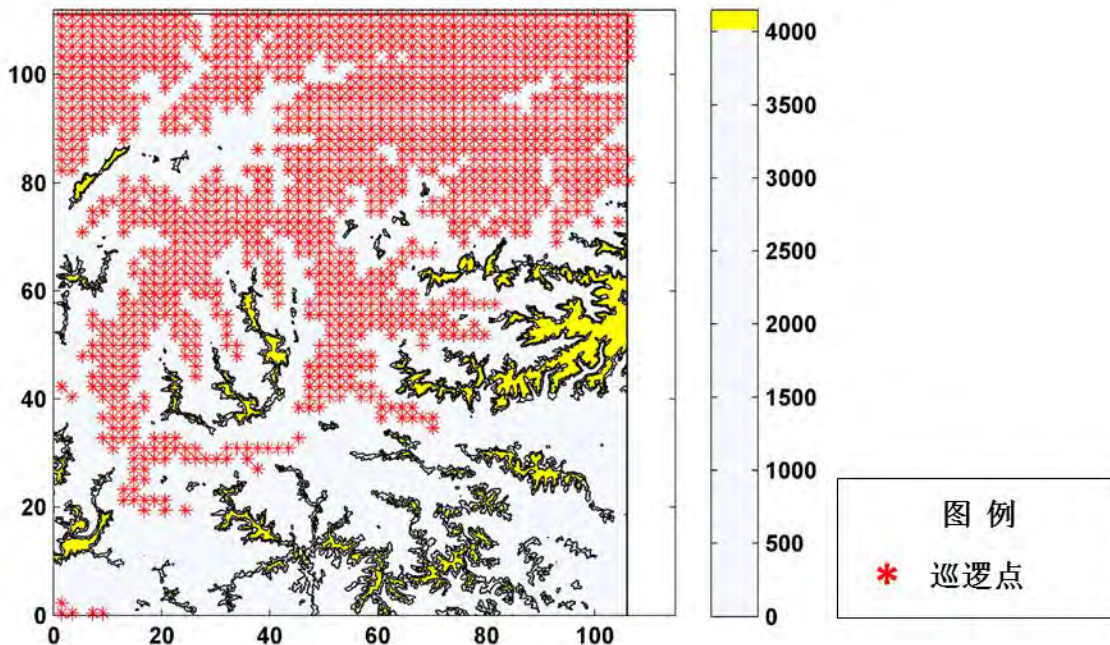


图 4-9 次生灾害巡查 4000m 以下区域网格划分巡逻点覆盖分布图

4.2 模型建立

4.2.1 3000m 以下灾情巡查分析

首先规定模型假设：首先假设无人机始终以 60km/h 匀速直线运动，忽略转弯引起的速度变化和轨迹变化；多架飞机可以同时出发，飞机从基地出发可以瞬时加速到 60km/h。规定安排无人机的数量为 i 。多目标规划的优化目标是获得目标区域 S 最大面积覆盖率 C 的同时使用最少的无人机，目标函数为：

$$\min \sum_S i \quad (4-1)$$

$$\max C = \frac{\sum_{n=1}^i S_n}{S_t} = \frac{S_t - \sum_{n=1}^i S_n' - \sum_{n=1}^i S_n''}{S_t} \quad (4-2)$$

式中， S_n 是每架飞机巡查的面积， S_t 是总的目标面积。 S_n' 是所有在无人机巡查范围外的目标面积， S_n'' 是所有被地形阻挡的目标面积。

约束条件包括：

①无人机速度为 60km/h： $v = 60$ ；

②飞机的任务完成时间在 4 小时之内， $0 < t_k < 4$ ；

③被无人机巡查覆盖的点与无人机的垂向高度差和两点间横向距离的比值大于 $\tan 60^\circ$ ， $X_0(S_i) \in S_{\alpha > 60^\circ}$ ，其中， $X_0(S_i)$ 代表 被无人机巡查覆盖的点。

④若建立近似球面在该点方向向上的法向量 $\left(-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1\right)$ 。如果该法向量和该点与无人机连线的向量积大于 0， $\left(-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1\right) \cdot \overrightarrow{XW} > 0$ 。其中， X 代表地面上的点坐标， W 代表无人机坐标。

⑤路径与所有山脉线段的交点总个数为 0。 $\sum_{k=1}^{N_{mo}} N(x_0) = 0$ ，式中 N_{mo} 是山体的编号， $N(x_0)$ 是路径与山体的交点。

因此，规划模型总的表达为：

$$\begin{aligned} & \min \sum_S i \\ & \max C = \frac{\sum_{n=1}^i S_n}{S_t} = \frac{S_t - \sum_{n=1}^i S_n' - \sum_{n=1}^i S_n''}{S_t} \\ & \left. \begin{aligned} & \sum_{k=1}^{N_{mo}} N(x_0) = 0 \\ & v = 60 \\ & 0 < t_k < 4 \\ & \left(-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1\right) \cdot \overrightarrow{XW} > 0 \\ & X_0(S_i) \in S_{\alpha > 60^\circ} \end{aligned} \right\} \quad (4-3) \end{aligned}$$

在时间约束的条件下，每个飞机的路径越短，巡查的点越多，所需要的飞机

就越多，所以多目标规划问题等效为求解能够巡查更多点的最短路径：

$$\min t(\pi_1, \pi_2, \dots, \pi_k) = \sum_{n=1}^k \frac{h_n}{v} \quad (4-4)$$

$$\max k \quad (4-5)$$

为了节约飞机用量，需要在指定的任务上优化轨迹获得最短的飞行时间，避免可以由一架飞机完成的任务分配给多架。由于飞机做匀速直线运动，所以飞机的飞行时间与飞行总长呈正比。该问题由此转化为求解最短路径的旅行商（TSP）问题，按照图论理论即在一个赋权完全图中，找出一个有最小权的 Hamilton 圈。

对于一定规模的旅行商问题，使用现代优化算法有着良好的效果，本模型采用遗传算法进行计算^[2,3]。

遗传算法(GA)是一种全局寻优搜索算法，它依据的是大自然生物进化过程中“适者生存”的规律。它首先对问题的可行解进行编码，组成染色体，然后通过模拟自然界的进化过程，对初始种群中的染色体进行选择、交叉和变异，通过一代代进化来找出最优适应值的染色体来解决问题，其原理图如图 4-10 所示^[4]。

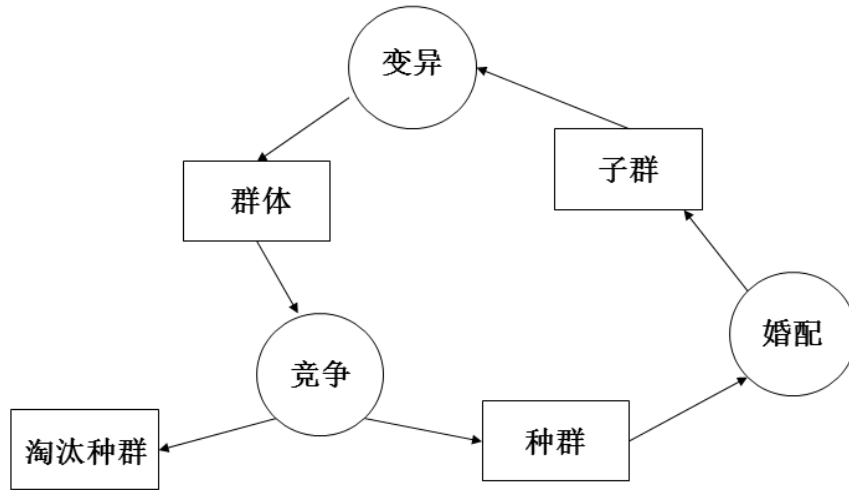


图 4-10 遗传算法 GA 原理示意图

(1) 编码策略

设巡查点个数为 N ，采用十进制编码，用随机数列 $w_1 w_2 \dots w_m$ 作为染色体，其中 $w_1 = 0$ ， $w_m = 1$ ， $0 \leq w_i \leq 1 (i = 2, 3, \dots, m-1)$ ， m 为 $N+2$ ，表示无人机从基地 H 出发，经过 N 个巡查点后回到基地的个数。每一个随机序列都和种群中的一个个体相对应。编码位置 i 代表目标 i ，位置 i 的随机数表示目标 i 在巡回中的顺序。

(2) 初始种群

先利用经典的近似算法：改良圈算法求得一个较好的初始种群。

对于随机产生的初始圈：

$C = \pi_1 \dots \pi_{u-1} \pi_u \pi_{u+1} \dots \pi_{v-1} \pi_v \pi_{v+1} \dots \pi_m$ ， $2 \leq \pi_u < \pi_v \leq m-1$ ，交换 u 和 v 之间的顺序，此时的新路径为 $\pi_1 \dots \pi_{u-1} \pi_v \pi_{u+1} \dots \pi_{v-1} \pi_u \pi_{v+1} \dots \pi_m$ 。

记 $\Delta f = (d_{\pi_{u-1} \pi_u} - d_{\pi_u \pi_{v+1}}) - (d_{\pi_{u-1} \pi_u} + d_{\pi_v \pi_{v+1}})$ ，若 $\Delta f < 0$ ，则以新路径修改旧路径，直到不能修改为止，就得到一个比较好的可行解。

直到产生 M 个可行解，并把 M 个可行解转换成染色体编码。

(3) 目标函数

目标函数为侦查所有目标的路径长度，适应度函数就取为目标函数。要求：

$$\min f(\pi_1, \pi_2, \dots, \pi_m) = \sum_{i=1}^{m-1} d_{\pi_i \pi_{i+1}} \quad (4-6)$$

(4) 交叉操作

交叉操作采用单点交叉，对于选定的两个父代个体 $f_1 = w_1 w_2 \dots w_m$ ， $f_2 = w_1' w_2' \dots w_m'$ ，随机地选取第 t 个基因处为交叉点，则经过交叉运算后得到的子代个体为 s_1 和 s_2 ， s_1 的基因由 f_1 的前 t 个基因和 f_2 的后 $m-t$ 个基因构成， s_2 的基因由 f_2 的前 t 个基因和 f_1 的后 $m-t$ 个基因构成。

(5) 变异操作

按照给定的变异率 P_m ，对选定变异的个体，随机的取三个整数，满足 $1 < u < v < w < m$ ，把 u 和 v 之间的基因段插到 w 后面。

(6) 选择

采用确定性的选择策略，在父代种群和子代种群中选择目标函数值最小的 M 个个体进化到下一代，保证父代的优良特性被保留下来。

在问题的分析中，参数覆盖率阈值是人为设定的。该参数直接决定巡逻点的个数，会对无人机轨迹的复杂程度产生影响。为了进一步优化目标，在保证覆盖率的情况下简化轨迹，降低无人机的出勤次数，需要对这两个参数进行敏感性因素分析，获取最优的模型参数。

因此，对覆盖率阈值进行敏感性因素分析。下调覆盖率阈值会增加巡逻点的数量，提升巡查面积，提高目标区域的覆盖率，但过高的阈值会增加无人机飞行轨迹的复杂度，增加任务完成用时，导致无人机使用数量增加。为了满足在 4 小时之内使区域 S 内海拔 3000 米以下的地方尽可能多地被巡查到，同时满足无人机数量最少，可以为每一个区域设定一个阈值，使之能够满足上述多目标规划的要求。

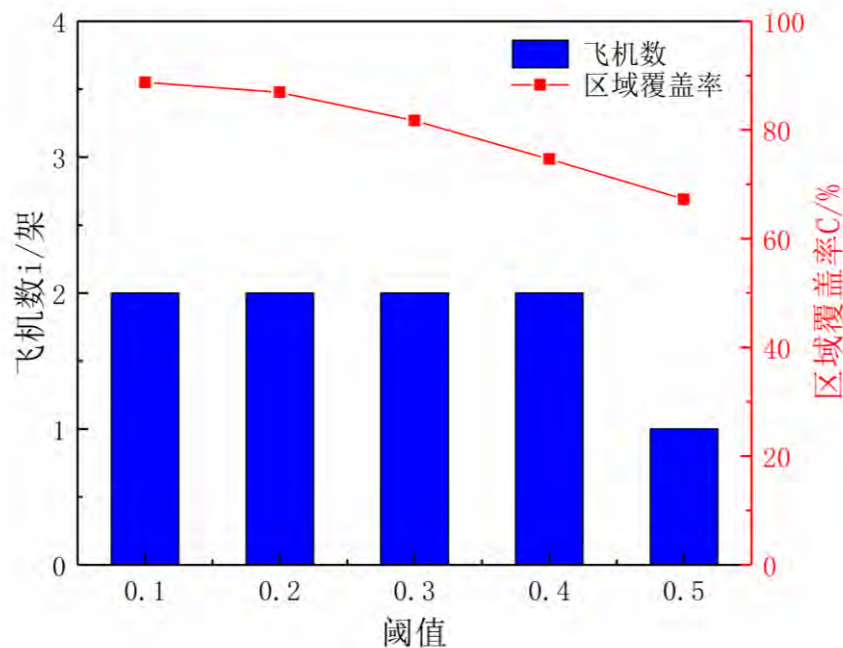


图 4-11 阈值敏感性分析

图 4-11 展示了区域 C 的阈值敏感性分析，由图可知，覆盖率阈值的减少会提高整体的巡查覆盖率，但也会增加无人机的数量。当阈值低于 0.5 时，覆盖率

增加不明显，但无人机的数量会急剧增加，飞机的效能会迅速降低。为了同时满足覆盖率和无人机数量的要求，该区域的阈值选择为 0.5。通过敏感性分析可以让每一个区域都得到一个阈值。

4.2.2 次生灾害巡查分析

次生灾害的优化模型也十分近似，如式（4-7）、式（4-8）和式（4-9）所示。

$$\min \sum_{S'} i \quad (4-7)$$

$$\max C = \frac{\sum_{n=1}^i S_n}{S'_t} = \frac{S_t - \sum_{n=1}^i S'_n - \sum_{n=1}^i S''_n}{S'_t} \quad (4-8)$$

$$s.t. \begin{cases} v = 60 \\ 0 < t_i < 8 \\ X(S_i) \in S_{\alpha > 60^\circ} \\ (-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1) \cdot \overline{XW} > 0 \end{cases} \quad (4-9)$$

相比于初始巡查，次生灾害巡查是涉及全局的，没有提前进行区域划分。为了更好地分配任务，降低无人机数量，通过相对于巡逻点的角度划分任务区间，如图 4-12 所示：

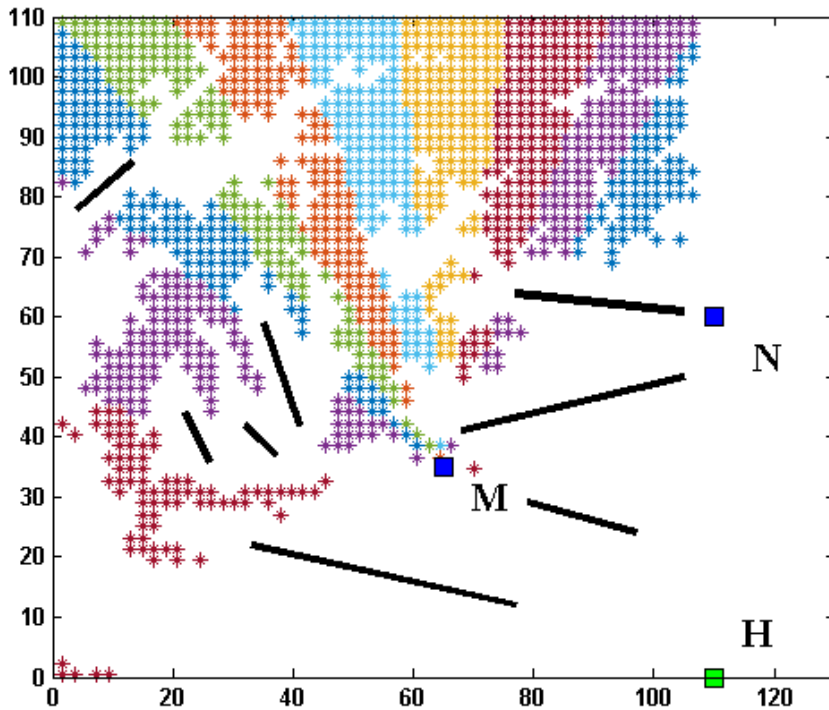


图 4-12 次生灾害巡逻点任务划分示意图

借助遗传算法，设计了针对次生灾害巡查模型的贪心算法：

Step1: 为最初的巡逻点分配无人机，用遗传算法计算最短路径和最短时间；

Step2: 添加逆时针方向一定角度的巡逻点给该无人机，如果该无人机的优化飞行时间小于 8 小时，继续为该无人机添加任务巡逻点，如果大于 8 小时，则执行 Step3；

Step3: 将 Step2 的巡逻点分配给下一台无人机，以此类推；

Step4: 重复 Step2 和 Step3, 直到为所有点分配完飞机。

该贪心算法通过局部最优解将尽可能多的点分配给每架无人机, 通过提高无人机的巡查量来减小无人机的使用数量, 同时合理分配巡查任务, 使每架无人机都能顺利返航。

4.3 模型求解

首先根据 3000m 以下灾情巡查模型, 求解无人机巡查灾区情况, 并计算区域覆盖率, 如表 4-1 所示, 无人机巡查路线如图 4-13 所示。

表 4-1 重点区域覆盖情况及执行巡查的无人机数量

区域中心点	阈值	飞机	区域覆盖率
B	0.5	3	40.05%
C	0.5	1	67.26%
D	0.5	1	70.67%
E	0.7	1	87.44%
总覆盖率		52.85%	

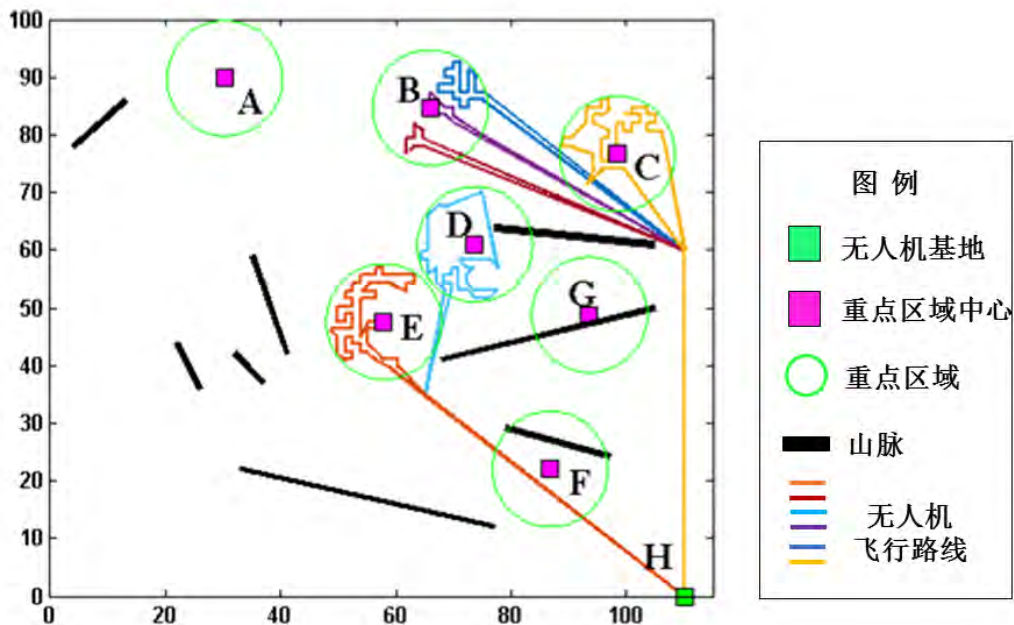


图 4-13 重点区域无人机巡查飞行路线

由图可以看出, 通过选择合理的覆盖率阈值能在保证一定的覆盖率的情况下尽可能减少无人机架次, 扫描总共使用 6 架飞机, 覆盖率为 52.85%。使得覆盖率较低的最主要原因是山体阻隔的影响, 这导致飞机消耗大量时间进行迂回, 限制了飞机的航迹自由。如果忽略山体的影响, 覆盖率可以进一步提升至 72%左右。

为及时发现次生灾害, 使用无人机在附件 1 给出的高度低于 4000 米的区域 (不限于 S) 上空巡逻, 通过次生灾害巡查优化模型的求解, 分 10 个任务区进行巡逻, 并确定每架无人机的飞行时间、路线, 保证在 72 小时内, 上述被巡查到的地方相邻两次被巡查的时间间隔不大于 3 小时 (如图 4-14 所示、表 4-2 所示)。

根据研究可知, 每任务区域派三架飞行, 共至少需要 30 架无人机, 覆盖率达到 66.18%。

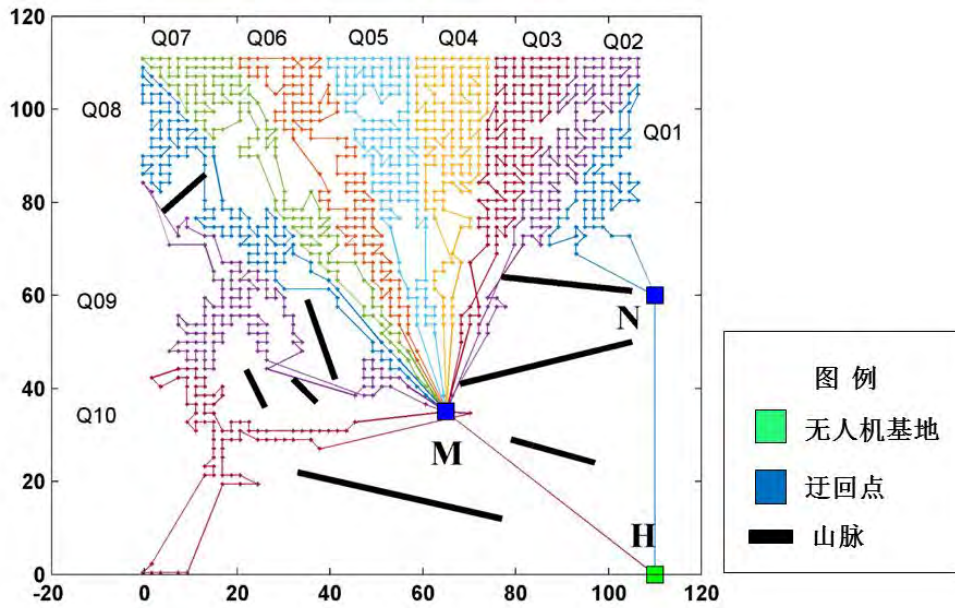


图 4-14 预防次生灾害无人机巡逻路径

表 4-2 无人机巡逻路径距离及时间

区域	起始角度	终止角度	飞行长度/km	中继点位置	距离/km	累计飞行长度/km	飞行时间/h
Q01	0	60	193.4	(110,60)	120	313.400	5.223
Q02	60	70	333.9			447.918	7.465
Q03	70	82	344.6			458.618	7.644
Q04	82	96	346			460.018	7.667
Q05	96	109	353.1			467.118	7.785
Q06	109	121	347.5	(65,35)	114.02	461.518	7.692
Q07	121	131	364.9			478.918	7.982
Q08	131	143	360.6			474.618	7.910
Q09	143	170	357.1			471.118	7.852
Q10	170	270	280			394.018	6.567

4.4 模型评价

(1) 该模型设计的巡查方案顺利完成了灾情巡查任务，并优化了无人机的数量和巡查覆盖率，合理解决了多目标规划问题。

(2) 依靠网格划分和贪心算法，将目标巡查区域重组为可由少量无人机完成的目标类群，合理的简化了目标布局，减少了无人机的各重点区域间的长途周转，无人机的巡查效能得到最大应用。

(3) 该模型综合考虑了观察仰角、视野阻碍、山体阻碍路线等约束条件的影响并进行了数学描述，使得本优化模型有较强的实用性，可应用于多种静态目标布局的无人机巡查路线设计中。

5 问题二模型建立与求解

5.1 问题描述与分析

无人机携带生命探测仪，可搜索生命迹象，能够给灾后救援提供准确的目标定位。题中从基地 H(110,0), J(110,55)(单位: km)处总共派出 30 架无人机(各 15 架)，任务完成后回到各自的出发地。其探测仪的有效探测距离不超过 1km, 且最大侧视角(探测仪到可探测处的连线与铅垂线之间的夹角)为 60 度。为使附件 1 所给出的全区域内海拔 3km 以下部分能被探测到的面积尽可能大, 规划无人机飞行路线, 且使从第一架无人机飞出到最后一架完成任务的无人机回到基地的时间间隔尽量短。



图 5-1 生命搜索示意图

问题二同样利用问题一中网格剖分的思想来分解问题, 问题要求一个点可被视为侦查的条件是在地面某点看无人机的仰角大于 30° 且视线不被山体阻隔, 如图 5-1。为了达到目标条件, 在简化路径的同时尽可能扫过更多的点, 由几何关系计算可得正方形边长为 1732m。同时, 通过敏感性因素分析, 选择用于巡逻点判定的覆盖率阈值是 0.3。

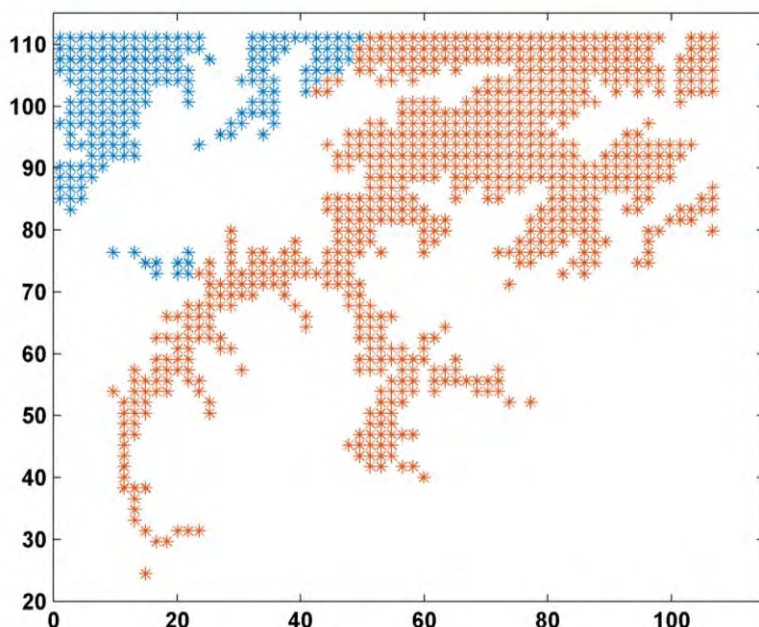


图 5-2 生命探测无人机出发基地分配图

首先通过两次 Q 类聚类分析方法^[5]将生命迹象检测任务分配给基地 H、J 的 30 架无人机。第一次聚类分析的目的是分配基地任务, 将所有的巡逻点依照相对于基地 H、J 的位置合理、均衡的分配给两个基地, 如图 5-2 所示。分类依据

是各点相对于两个基地的距离差，使用“车比雪夫距离”进行聚类。第二次聚类分析则根据巡逻点的聚落情况分配 30 架无人机各自的任务，每架飞机在续航时间内在保证一定覆盖率情况下，避免无意义的周转，尽可能控制总的行动时间，如图 5-3 所示。

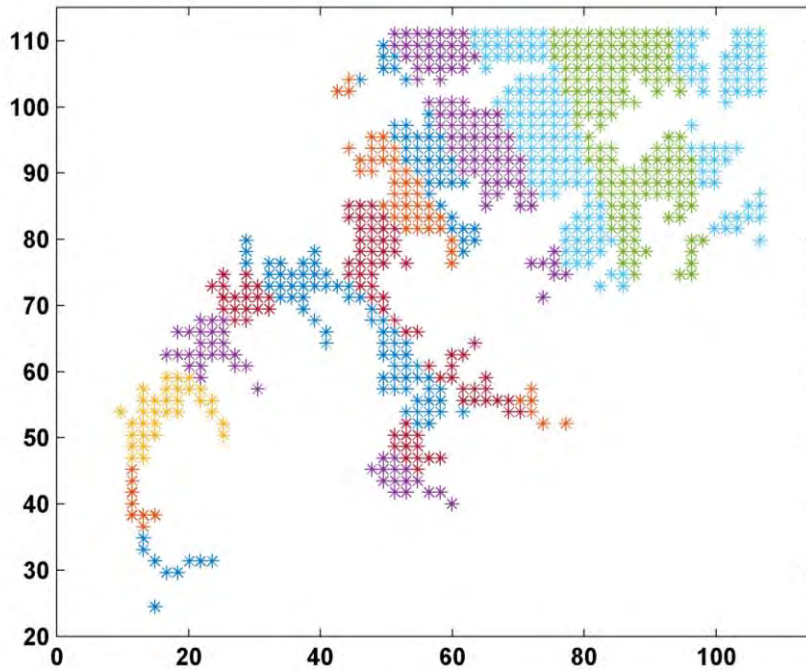


图 5-3 H 基地无人机聚类图

问题二相比于问题一约束条件和优化目标都有所变化：一是无人机可以进行高度变化，因此山脉不再会阻碍无人机的三维运动轨迹；而是无人机的探测范围有所变化，虽然无人机的探测仪到可探测处的连线与铅垂线之间的夹角拓宽至 60° ，但无人机探测仪有 1000m 的探测距离限制，就此，建立起关于问题二的优化模型：

5.2 模型建立

首先假设无人机始终以 60km/h 匀速直线运动，忽略转弯引起的速度变化和轨迹变化；30 架飞机可以同时出发，飞机的俯仰角能够满足飞机高度变化的要求。飞机从基地出发可以瞬时加速到 60km/h。问题二的优化目标是从第一架无人机飞出到最后架完成任务的无人机回到基地的时间间隔，从本质上来说，这是一个最大值最小化问题，即如何使用时最长的无人机耗时最小。为了保证一定的覆盖率，设置约束条件为覆盖率 C 大于 60% 其优化目标和约束条件如式 (5-1)、(5-2) 所示：

$$\min | \max | t_i t(\pi_1, \pi_2, \dots, \pi_{30}) || \quad (5-1)$$

$$\begin{cases}
 v = 60 \\
 0 < t_k < 8 \\
 s.t. \left\{ \begin{aligned}
 C = \frac{\sum_{n=1}^i S_n}{S_i'} \geq 0.6 \\
 X_0(S_i) \in S_{\alpha > 60^\circ} \\
 (-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1) \cdot \overline{XW} > 0
 \end{aligned} \right.
 \end{cases} \quad (5-2)$$

其搜索求解流程为：

Step1: 根据 Q 类聚类分析结果分配任务，通过遗传算法求解各自最短路径。

Step2: 通过比较找到耗时最多的飞机，分配部分巡逻点给周围的飞机。

Step3: 重新计算路径并寻找耗时最多的飞机，如果耗时明显减小，则重复 Step2，如果没有明显改变，则目前所得结果即为最优解。

5.3 模型求解

通过模型求解，规划无人机飞行路线，保证全区域内海拔 3km 以下部分能被探测到的面积尽可能大的情况下，无人机总巡航时间最小。基地各无人机的具体路线、飞行距离以及飞行时间，如图 5-4、表 5-1、表 5-2 所示。

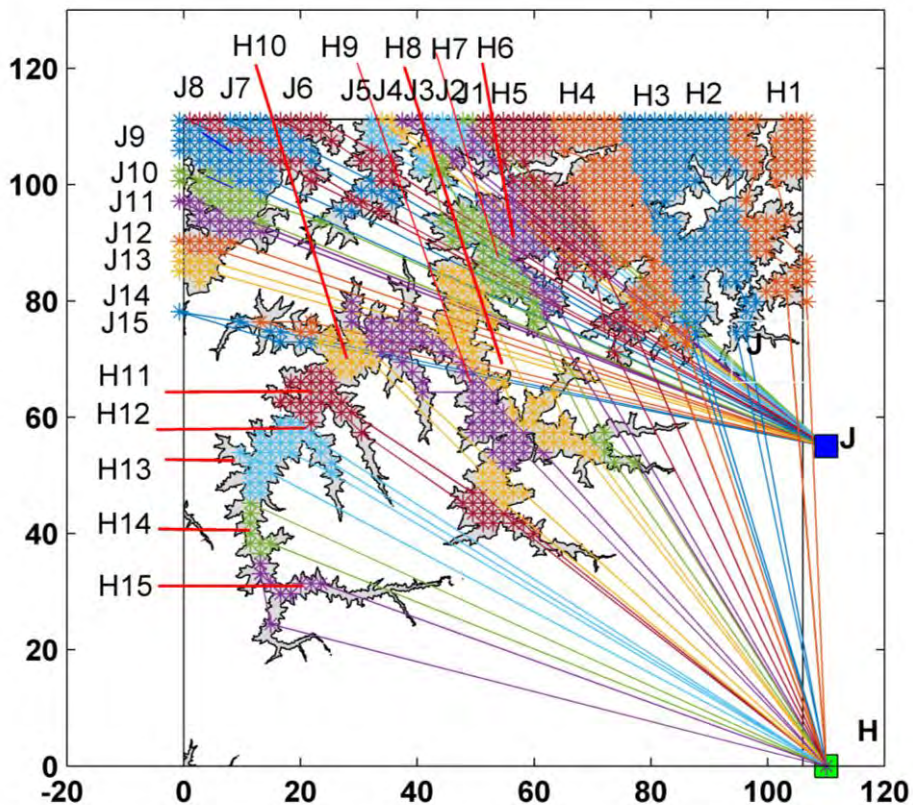


图 5-4 30 架无人机行探测生命任务时最优路线图

表 5-1 H 基地 15 架无人机执的不同路径的所需路程和时间

编号	路程/km	时间/h	编号	路程/km	时间/h
H1	293.596	4.893	H9	281.217	4.687
H2	302.763	5.046	H10	247.864	4.131
H3	288.494	4.808	H11	253.598	4.227
H4	296.996	4.950	H12	225.236	3.754
H5	336.991	5.617	H13	248.754	4.146
H6	339.898	5.665	H14	223.847	3.731
H7	276.740	4.612	H15	214.718	3.579
H8	256.965	4.283	/	/	/

表 5-2 J 基地 15 架无人机执行探测生命任务时的不同路径的所需路程和时间

编号	路程/km	时间/h	编号	路程/km	时间/h
J1	167.920	2.799	J9	272.933	4.549
J2	177.598	2.960	J10	255.252	4.254
J3	184.263	3.071	J11	240.572	4.010
J4	189.409	3.157	J12	233.335	3.889
J5	194.446	3.241	J13	236.934	3.949
J6	222.006	3.700	J14	198.646	3.311
J7	237.773	3.963	J15	226.459	3.774
J8	248.203	4.137	/	/	/

由表可以得出,经过优化的第一架无人机飞出到最后一架完成任务的无人机回到基地的时间间隔约为 5.6 小时,耗时最大值得到控制,基本达到题目要求。

5.4 模型评价

针对路径高度、探测范围、探测距离、探测时间等一系列约束条件的变化,对问题一的模型进行改进,通过两次 Q 类聚类分析合理地规划了基地的任务分配和无人机的任务分配,避免飞机周转浪费时间,并通过遗传算法求取简化路径,获得最短的任务耗时。结果表明该模型有较好的通用性。

6 问题三模型建立与求解

6.1 问题描述与分析

大地震发生后，地面电力设施被破坏，灾区通信中断。太阳能无人机(可以作地面移动终端之间的通信中继，为灾区提供持续的通信保障。假设无人机在空中飞行时，可与距离 3000 米以内的移动终端通信，无人机之间的最大通信距离为 6000 米，制定合理无人机的飞行路线和优化无人机数量，以保证在白天 12 小时内，任意两个地面终端之间都能实现不间断通信。

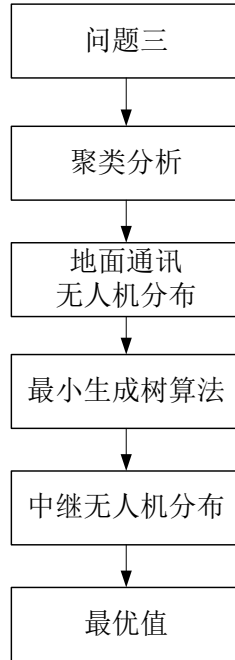
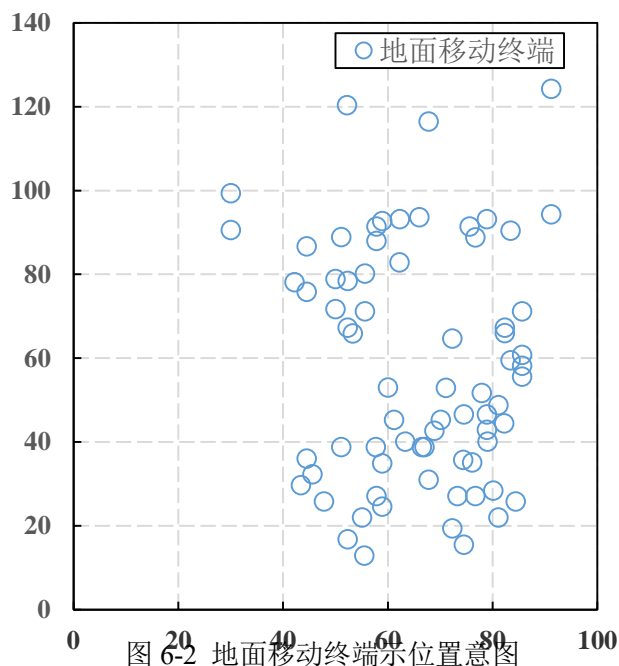


图 6-1 问题三求解思路及算法

作为中继的无人机之间的切换时间忽略不计，地面终端的移动距离不超过 2 千米，其地面移动终端位置如图 6-2 所示。



如果建立动态无人机模型，要求无人机做匀速直线运动，那么每一个移动通讯终端都需要大量飞机架次保障通讯，需要调动的无人机数量将会成倍增加。因此，本模型决定采用拟静态无人机分配方案，要求各个无人机对移动通讯终端逐个保障，无人机在移动通讯终端上空以最小转弯半径 100m 盘旋。

为了尽可能增加无人机的通讯距离，同时防止地形的阻碍，使得移动通讯终端实时得到通讯保障，本模型假设无人机在移动通讯终端以上 500m 处飞行。

出于控制无人机数量的目的，需要分析多个通讯终端使用一架无人机的情况。如图所示，地面终端的移动距离不超过 2km，要求无人机的通讯范围能够完全覆盖地面终端的移动范围。如果两个移动终端移动范围的中心距小于 2km，那么可以在两点连线中心处部署一架飞机，同时完成对两个终端的通讯。通过聚类分析的方法，设置聚类方式为“欧式距离”、“最大距离”，可以找出所有中心距小于 2km 的点类群，聚类结果如图 6-3 所示。

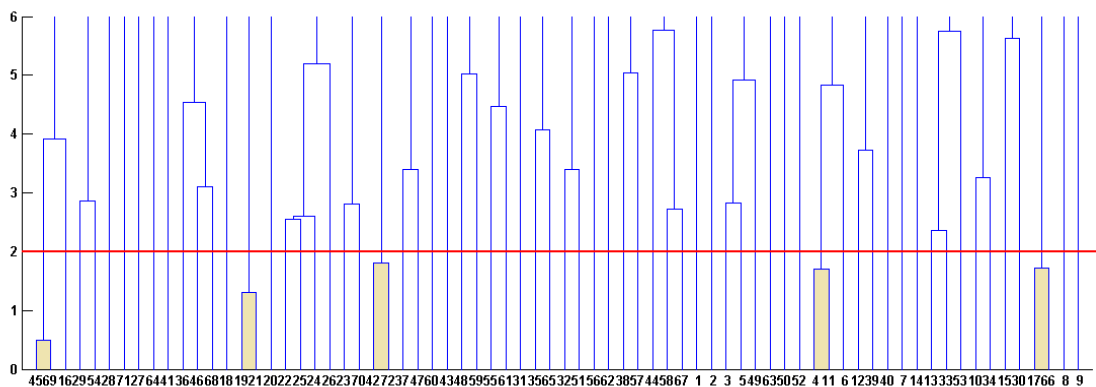


图 6-3 地面移动终端聚类分析结果

聚类图纵坐标为欧式距离，横坐标为终端序号。以 42 号终端(76.1,35.1)和 72 号终端(74.4,35.7)为例，两者在 2 以下便聚为一类，说明两者的坐标距离小于 2km，因此可以共享一架无人机。聚类分析的结果发现共有五组这样的共享无人机组，如表 6-1。

表 6-1 聚类分析五组无人机组终端序号

类群	终端序号	终端序号
第一类群	45	69
第二类群	19	21
第三类群	42	72
第四类群	4	11
第五类群	17	66

对于终端无人机间距离超过最大通讯距离 6km 的情况，以及山脉可能造成的通讯网络截断，可以根据通讯长度合理布置中继无人机。

问题要求尽可能使用较少的无人机，上文通过聚类分析的方式减少了用于地面终端通讯的无人机，但无人机之间通讯的中继无人机数量还没有优化。从总体布局分析，整个通讯网络的长度和中继无人机的数量呈正相关性，将终端无人机连接成一个网络所用的路径越短，中继无人机数量就越少。所以无人机数量优化问题可以等效为网络模型问题。

6.2 模型建立

假设太阳能无人机能够以 100 米半径进行盘旋，在天明时就可以部署到指定位置，地面移动终端可以在起始位置半径 2km 内随意移动；无人机与地面距离

恒定，忽略山体阻挡。题目给出的要求使用最少的无人机完成包含 72 个地面终端的网络通讯，目标函数与约束条件如式 (6-1)、式 (6-2)。

$$\min \sum (i_1 + i_2) \quad (6-1)$$

$$s.t. \begin{cases} i_1 = \sum G \\ i_2 = \sum INT(\frac{h_i}{6}) \\ v = 60 \\ 0 < t_k < 12 \\ (-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1) \cdot \overline{XW} > 0 \\ XW \leq 3 \\ W_i W_j \leq 6 \end{cases} \quad (6-2)$$

式中 i_1 为地面通讯无人机的数量，和聚类分析所得类数相同； i_2 为中继无人机数量，由大于 6km 的点间路径数决定。

与地面终端通讯的无人机的数量通过聚类分析优化。而针对中继无人机，最少的无人机数量可以等效为最短的网络路径，建立连接 n 个节点的路径网络，使得总里程最小，这本质上属于一类最小生成树问题^[6]，即：

$$w(T^*) = \sum_T \min \{w(T)\} \quad (6-3)$$

采用 Prim 算法构造最小生成树。构造连通赋权图 $G = (V, E, W)$ 的最小生成树，设置两个集合 P 和 Q，其中 P 用于存放 G 的最小生成树中的顶点，集合 Q 存放 G 的最小生成树中的边。令集合 P 的初值为 $P = \{v_1\}$ ，（假设构造最小生成树时，从顶点 v_1 出发），集合 Q 的初值为 $Q = \emptyset$ 。Prim 算法的思想是，从所有 $p \in P$ ， $v \in V - P$ 的边中，选取具有最小权值的边 pv ，将顶点 v 加入集合中，将边 pv 加入到集合 Q 中，如此不断重复，直到 $P=V$ 时，最小生成树构造完毕，这时集合 Q 中包含了最小生成树的所有边。

Prim 算法如下：

(1) $P = \{v_1\}, Q = \emptyset$ ；

(2) While $P \sim V$ 。

找寻最小边 pv ，其中， $p \in P$ ， $v \in V - P$ ；

$$\begin{cases} P = P + \{v\}; \\ Q = Q + \{pv\}; \end{cases} \quad (6-4)$$

6.3 模型求解

通过模型求解，得到无人机的网络路径以及各类无人机的部署情况，如图 6-4、表 6-2 所示。

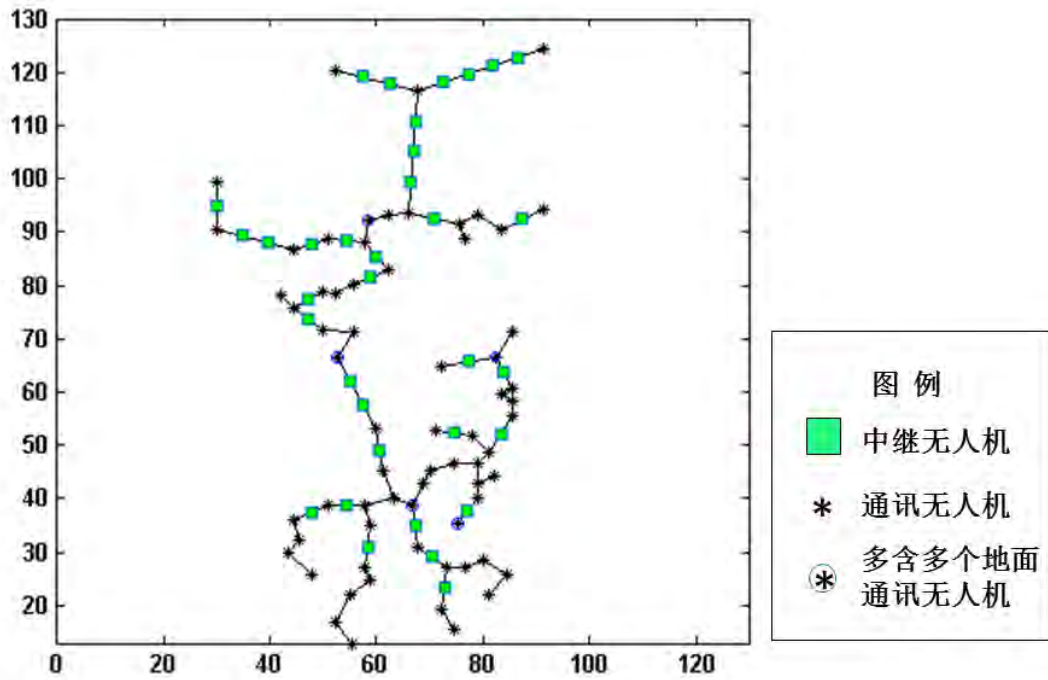


图 6-4 最小生成树的无人机部署及路线网络

表 6-2 中继无人机部署位置坐标

序号	X 坐标 /km	Y 坐标 /km	序号	X 坐标 /km	Y 坐标 /km	序号	X 坐标 /km	Y 坐标/km
1	87.3	92.35	13	55.2	62.07	25	77.3	65.68
2	70.8	92.5	14	60.6	49.15	26	67.35	110.78
3	60	85.45	15	77.13	37.75	27	66.9	105.05
4	54.45	88.45	16	54.4	38.8	28	66.45	99.33
5	47.8	87.8	17	74.5	52.3	29	57.4	119.1
6	58.9	81.55	18	47.8	37.4	30	62.6	117.8
7	47.25	77.35	19	67.28	34.9	31	86.52	122.74
8	47.25	73.75	20	70.55	29.05	32	81.84	121.18
9	34.83	89.3	21	72.8	23.25	33	77.16	119.62
10	39.67	88	22	58.35	31	34	72.48	118.06
11	30	95	23	83.35	52.2			
12	57.6	57.53	24	83.95	63.73			

表 6-2 移动终端无人机部署位置坐标

序号	X 坐标 /km	Y 坐标 /km	序号	X 坐标 /km	Y 坐标 /km	序号	X 坐标 /km	Y 坐标 /km
1	91.2	94.3	24	70.1	45.3	47	68.9	42.7
2	83.4	90.4	25	50	71.7	48	72.3	19.4
3	76.7	88.8	26	51.1	38.8	49	44.5	36
4	75.6	91.4	27	43.4	29.7	50	55.5	12.9
5	57.8	88	28	52.3	78.4	51	58.9	24.6
6	51.1	88.9	29	44.5	75.8	52	80.1	28.4
7	30	90.6	30	58.9	34.9	53	67.8	31
8	30	99.4	31	82.2	44.4	54	74.5	15.5
9	42.2	78.1	32	73.3	27.1	55	47.8	25.8
10	62.3	93.2	33	52.3	16.8	56	91.2	124.3
11	50	78.9	34	66	93.6	57	74.5	46.6
12	44.5	86.7	35	62.2	82.9	58	57.7	38.8
13	55.6	71.2	36	71.1	52.9	59	57.8	27.1
14	63.3	40.1	37	81.1	22	60	81.1	48.8
15	72.3	64.7	38	55.1	22	61	78.9	42.9
16	85.6	71.2	39	78.9	46.6	62	60	53
17	85.6	58.2	40	76.7	27.1	63	66.75	38.8
18	79	40.1	41	84.4	25.8	64	82.3	66.65
19	85.6	60.8	42	78.9	93.2	65	75.25	35.4
20	83.4	59.5	43	67.8	116.5	66	58.35	92.05
21	85.6	55.6	44	45.6	32.3	67	52.8	66.6
22	77.9	51.7	45	52.2	120.4			
23	61.2	45.3	46	55.6	80.2			

以上模型求解出来的无人机网络拥有最少的网络路径和最少的通讯无人机，每架无人机具有较长的服役时间，每架无人机得到充分利用。

6.4 模型评价

该灾区通讯中继模型设计了一套静态无人机网络，并依靠聚类分析和最小生成树问题解法优化了地面移动终端通讯机和中继通讯机的数量，使得完成 12 小时通讯任务所需要的无人机数量最少。相比于动态无人机网络，每架无人机有着更多的通讯时长，无人机所需数量也会相应减少。

7 问题四模型建立与求解

7.1 问题描述与分析

为实现对地数据传输，指挥中心拟从 H 派出 3 架无人机携带通信装备向灾区内的 72 个地面终端发送内容不同，总量均为 500M 的数据。

设每台通信装备的总功率是 $5w$ ，可同时向不超过 10 个地面终端发送数据。数据传输过程可以简化为：当地面终端 i 看无人机的仰角大于 30° 、距离不超过 3km 且没有山体阻隔时，如果无人机当前服务用户少于 10 个，则开始向 i 发送数据，并瞬间完成所有用户的功率再分配，否则，搁置 i 的需求，直到有地面用户退出，若此时 i 仍在可服务区域，以先到先服务的原则，先为 i 服务。若在一个服务时间区间内不能传完全部数据，则以后区间可以续传。

假设无人机飞行速度在 60~100 km/h 之间可调(水平面内最大加速度 $\pm 5m/s^2$ ，铅垂面内最大加速 $\pm 2m/s^2$ ，可同时在两个方向上加速)，建立数学模型，规划无人机设计恰当的航线、速度以及所服务的用户，并为每一个用户分配恰当的功率，使得无人机完成所有任务的时间总和尽量短。

假设 i 用户在时刻 t 接收到无人机发送的信息速率为：

$$r_i(t) = B_i \log_2 \left(1 + \frac{p_i(t)}{\rho_0 d^2(u,i)} \right) \quad (7-1)$$

式中， B_i 表示无人机服务 i 的子信道带宽，Hz； $p_i(t)$ 表示 t 时刻无人机为第 i 个地面用户所在的子信道分配的功率， w ； $d(u,i)$ 表示 t 时刻无人机与 i 之间的欧氏距离， m 。 ρ_0 为信道特性参数，取为 4.314×10^{-10} 。

以用户与无人机距离 3km，研究不同的 180、150、100 子信道带宽下的信息速率与用户数的敏感性分析，如图 7-1 所示。

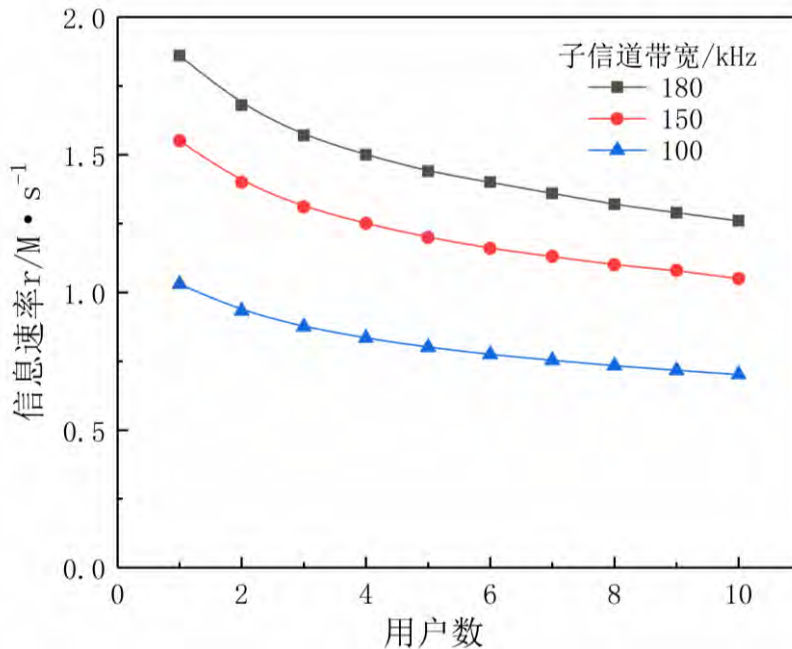


图 7-1 不同子信道带宽下的信息速率与用户数曲线

由图 7-1 可知，相同用户数情况下，子信道带宽越大，其信息速率越大；同与子信道带宽下，用户数越大，其用户单体的信息速率越小，其信息速率变化率

(斜率) 越小, 用户数对单体的信息速率的影响不大, 但整体用户的信息速率成倍增加, 因此, 在轨迹优化的过程中应增加无人机同时传输的用户数。

以用户数为 1, 研究不同的 180、150、100 子信道带宽下的信息速率与用户无人机间的距离的敏感性分析, 如图 7-2 所示。

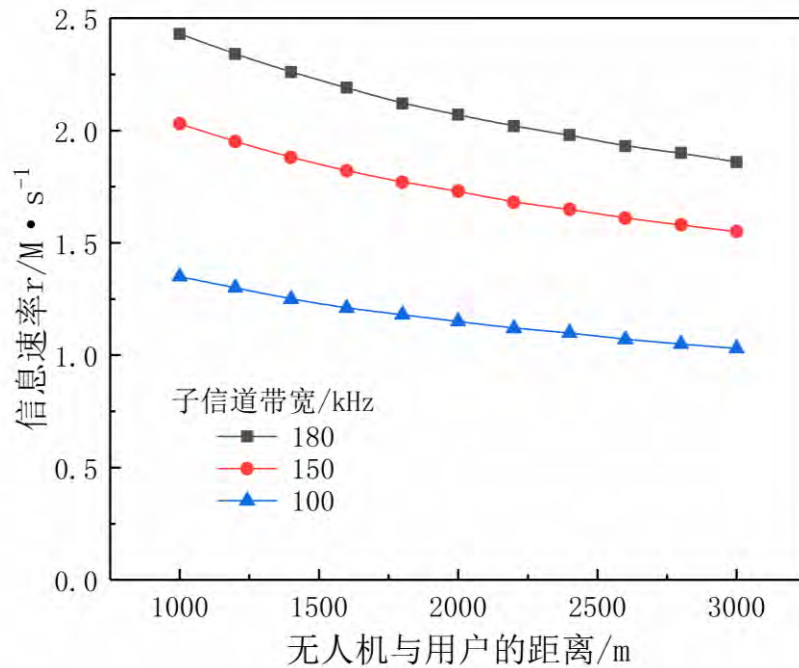


图 7-2 不同子信道带宽条件下信息速率与距离的变化曲线

由图 7-2 可知, 同一用户数情况下, 子信道带宽越大, 其信息速率越大; 同一子信道带宽条件下, 无人机与用户间距离越大, 其信息速率越低, 其信息速率变化率(斜率)越小, 表明无人机与用户间距离对信息速率的影响不大。

在多个地面终端接收无人机数据时, 如何合理地分配功率提高效率同样是本优化问题的重点。以 2 个地面终端分享无人机功率为例, 分析两地面终端所享的功率差值对平均传输速度的影响, 如图 7-3 所示。

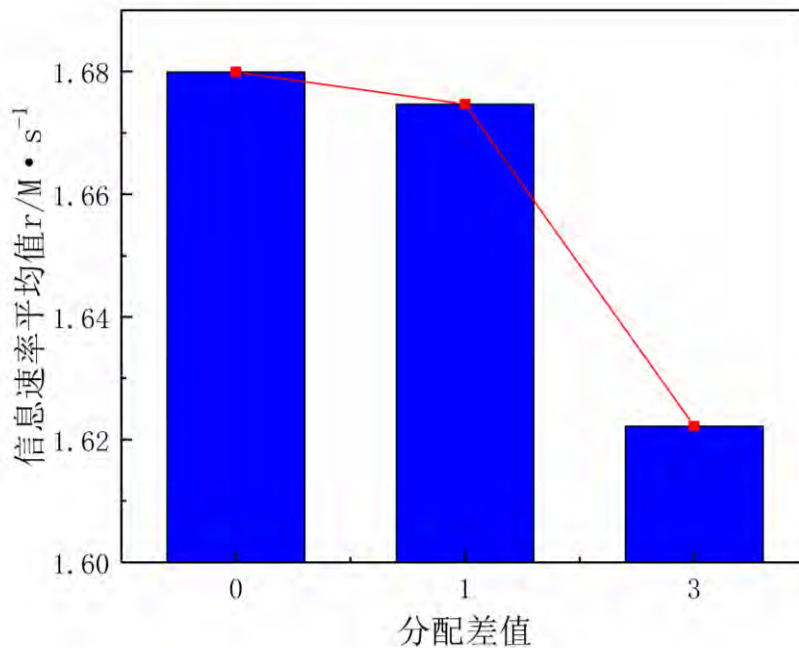


图 7-3 分配方式与平均传输速度的关系

由图 7-3 可知，两地面终端所享的功率差值越大，平均传输速度越慢。同时，差异分配功率会引起传输速率的不均衡，整体的传输速度会被最慢的一方拖累，最终降低效率。由于各个共享无人机的地面终端子信道带宽差异不大，因此当遇到需要分享无人机功率时，选择平均分配功率，提高传输效率。

增加无人机同时传输的用户数可以通过聚类分析实现，根据已知条件，凡是在距离 3000m，地面仰角 30°，地面半径 2598m 的区域内都可以共享一台无人机，为了将任务共享的终端编组，这里同样按照最大欧式距离进行聚类分析，使得所有 5196m 范围内的地面终端聚为一类，如图 7-4 所示、表 7-1 所示。

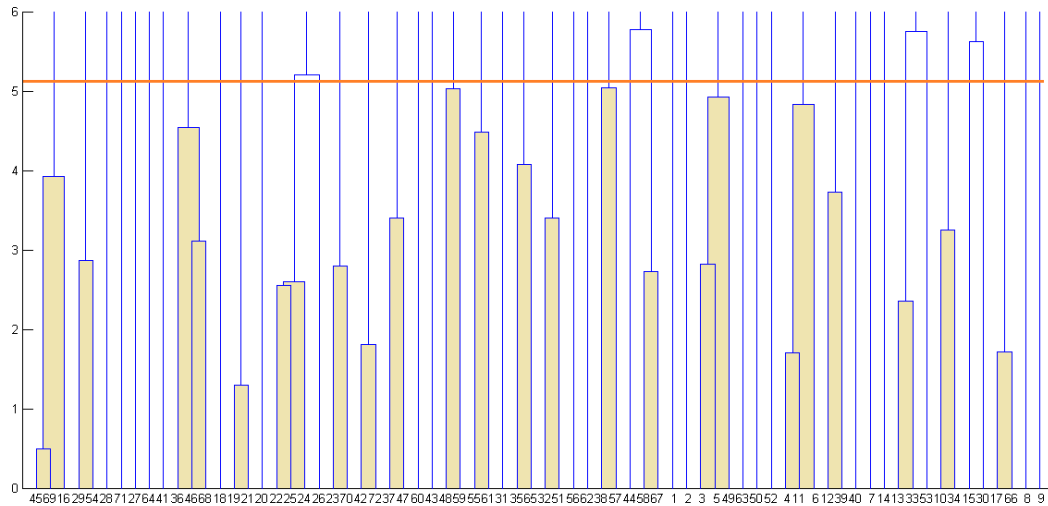


图 7-4 地面终端聚类分析结果

表 7-1 地面终端聚类分组表

类数	同一类序号			中心 X 坐标	中心 Y 坐标
1	45	69	16	65.6	39.2
2	29	54		69.5	44.0
3	36	46	68	80.7	46.6
4	19	21		82.3	66.7
5	22	25	24	84.9	59.5
6	23	70		79.0	41.5
7	42	72		75.3	35.4
8	37	47		75.0	27.1
9	48	59		82.3	27.1
10	55	61		73.4	17.5
11	35	65		58.3	36.9
12	32	51		44.5	31.0
13	38	57		53.9	14.9
14	58	67		58.4	25.9
15	3	5	49	77.1	91.1
16	4	11	6	58.2	90.7
17	12	39		64.2	93.4
18	13	33		51.2	78.7
19	10	34		43.4	77.0
20	17	66		52.8	66.6

经过测算，无论无人机给一个用户传输还是和多个目标传输，其时间都控制在 15 分钟之内，相比于无人机在行进过程中几小时的耗时时间，传输时间对于优化总耗时的影响较小，因此，无人机完成所有任务的时间总和优化问题可转化为无人机路径的最大值最小化问题。

为了更好地满足减小用时和尽快传输数据，结合以上分析，无人机的速度设计策略为：当遇到单用户传输任务时，尽可能调整飞行速度，使其能以传输数据所用时间恰好飞过传输半径区域，如图 7-5-a)所示；若以最低速度 60km/h 仍不能实现传输，需使无人机在移动终端上空进行半径 100m 的盘旋飞行，如图 7-5-b)所示；当遇到多用户传输时，无人机以 100km/h 的速度在中心上空盘旋，直到最后一个移动终端的传输任务完成再离开，如图 7-5-c)所示。

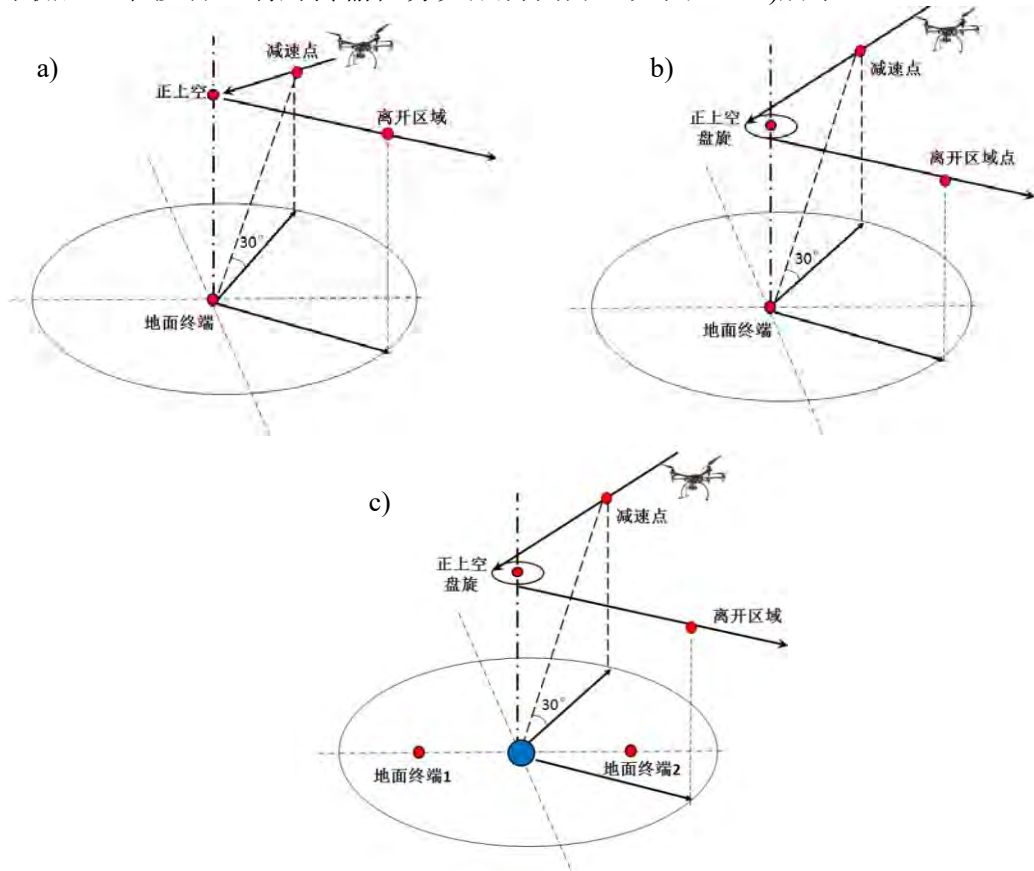


图 7-5 无人机与移动终端数据传输策略

7.2 模型建立

问题假设无人机与地面距离恒定，忽略山体阻挡；由于加速度较大，无人机可以瞬时变速；首先假设无人机始终以 60km/h 匀速直线运动，忽略转弯引起的速度变化和轨迹变化。问题四的优化目标属于最大值最小化问题，其目标函数如式 (7-2)。

$$\min | \max | t(\pi_1, \pi_2, \pi_3) | | \quad (7-2)$$

$$s.t. \begin{cases} 60 \leq v \leq 100 \\ 0 < t_k < 12 \\ (-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1) \cdot \overline{XW} > 0 \\ XW \leq 3 \\ X_0(S_i) \in S_{\alpha > 30^\circ} \end{cases} \quad (7-3)$$

针对该类最大值最小化问题，通过枚举法实现求解，步骤如下：

Step 1: 将地面终端用两条直线 C_1 、 C_2 分割成三台无人机的任务；

Step 2: 通过遗传算法计算三台无人机的最优路径，并选取时间最大值 T' 作为参考系。

Step 3: 通过枚举法不断获得新的直线 C_1' 、 C_2' ，重复 Step 2 的计算得到时间最大值 T'' ，并和 T' 作对比选较小者作为新的参考系。

Step 4 重复 Step 3，直至参考系不再明显下降。

7.3 模型求解

在模型运算前需要寻找足够多的分割直线 C ，使得算法能枚举到最优化的分配任务。试验了数量从 5 到 80 等分区域的直线对优化结果的影响，结果如图 7-6 所示。由图可以看出，当等分点数大于 40 时，得到的无人机路径时间最大值 $\max |t(\pi_1, \pi_2, \pi_3)|$ 最小，大致为 3.1h。因此选用等分点数为 40。

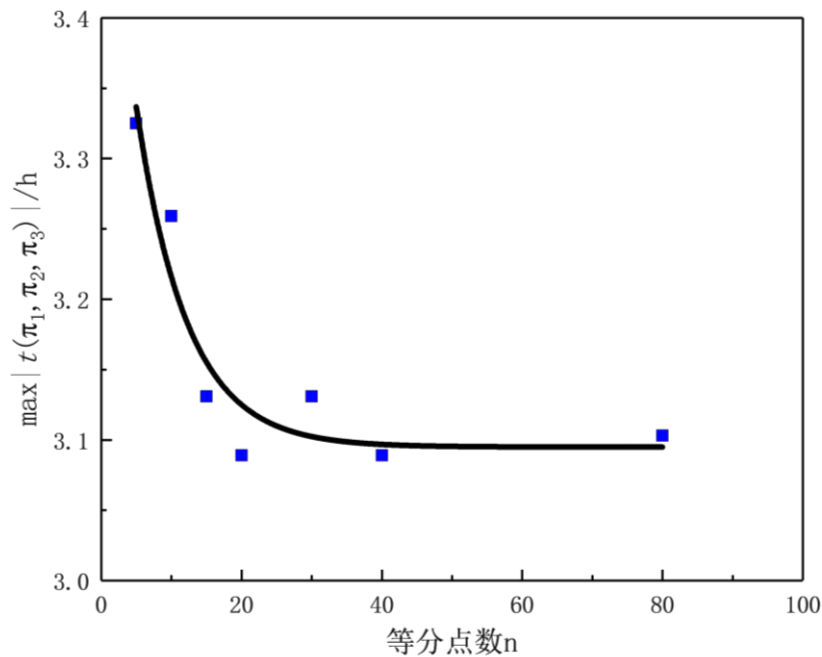


图 7-6 优化结果随等分点数的变化

根据算法运行结果，三架无人机的任务轨迹和任务分割曲线如图 7-7 所示。

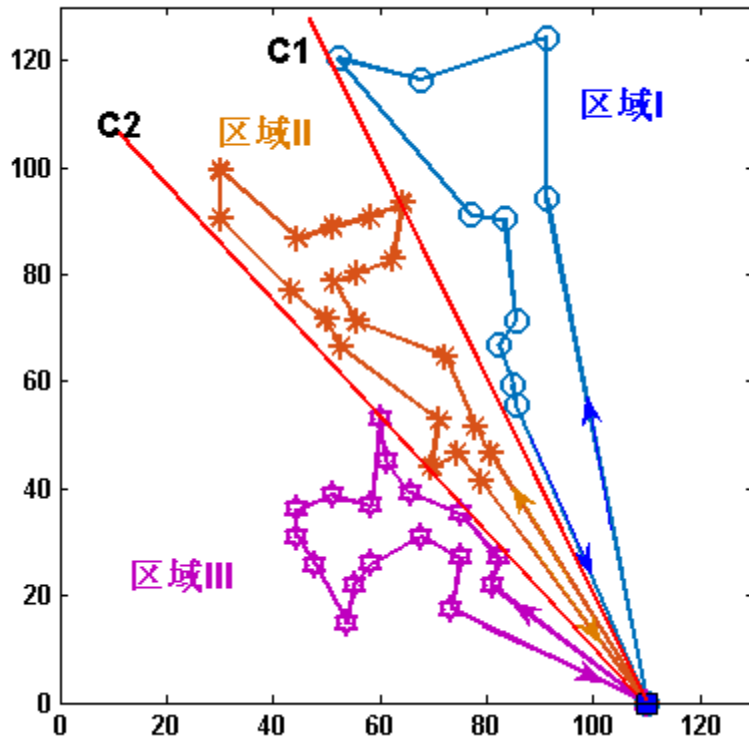


图 7-7 无人机飞行路线

三架无人机的服务用户、速度策略和任务用时如表 7-2、表 7-3 和表 7-4 所示。无人机完成所有任务的时间总和为 4.41h。

表 7-2 区域 I 无人机飞行方案

节点	X 坐标 /km	Y 坐标 /km	通信区域内速 度速度/ $\text{km} \cdot \text{h}^{-1}$	移动通信上空盘 旋飞行路程/km	通讯区域内所 飞行时间/h	用户数
1	110.0	0.0	100.00	0.00	0	/
2	85.6	55.6	60.00	1.88	0.12	1
3	84.9	59.5	100.00	16.34	0.22	3
4	82.3	66.7	100.00	15.08	0.20	2
5	85.6	71.2	60.00	1.88	0.12	1
6	83.4	90.4	61.84	0.00	0.08	1
7	77.1	91.1	100.00	13.82	0.19	3
8	52.2	120.4	60.00	0.63	0.10	1
9	67.8	116.5	60.00	1.26	0.11	1
10	91.2	124.3	60.00	1.88	0.12	1
11	91.2	94.3	69.57	0.00	0.07	1
12	110.0	0.0	100.00	0.00	0.05	/
非通信区域的无人机飞行时间/h					2.52	
优化最短的路径总长度/km					308.93	
总时间					3.90	

表 7-3 区域II无人机飞行方案

节点	X 坐标 /km	Y 坐标 /km	通信区域内速 度速度/km · h-1	移动通信上空盘 旋飞行路程/km	通讯区域内所 飞行时间/h	用户数
1	110.0	0.0	100.00	0.00	0.00	/
2	79.0	41.5	100.00	12.57	0.18	2
3	74.5	46.6	60.00	2.51	0.13	1
4	69.5	44.0	100.00	9.42	0.15	2
5	71.1	52.9	60.00	0.63	0.10	1
6	52.8	66.6	100.00	10.68	0.16	2
7	50.0	71.7	77.31	0.00	0.07	1
8	43.4	77.0	100.00	11.94	0.17	2
9	30.0	90.6	61.84	0.00	0.08	1
10	30.0	99.4	73.44	0.00	0.07	1
11	44.5	86.7	61.84	0.00	0.08	1
12	51.1	88.9	65.71	0.00	0.08	1
13	58.2	90.7	100.00	10.05	0.15	3
14	64.2	93.4	100.00	10.68	0.16	2
15	62.2	82.9	60.00	1.88	0.12	1
16	55.6	80.2	69.57	0.00	0.07	1
17	51.2	78.7	100.00	7.54	0.13	2
18	55.6	71.2	60.00	0.63	0.10	1
19	72.3	64.7	60.00	1.88	0.12	1
20	77.9	51.7	60.00	0.63	0.10	1
21	80.7	46.6	100.00	16.34	0.22	3
22	110.0	0.0	100.00	0.00	0.05	/
非通信区域的无人机飞行时间/h					1.93	
优化最短的路径总长度/km					302.55	
总时间					4.41	

表 7-5 区域III无人机飞行方案

节点	X 坐标 /km	Y 坐标/km	通信区域内速度 速度/km · h-1	移动通信上空盘 旋飞行路程/km	通讯区域内所飞 行时间/h	用户数
1	110	0	100	0	0	/
2	81.1	22	60	1.26	0.11	1
3	82.3	27.1	100	10.05	0.15	2
4	75.3	35.4	100	10.05	0.15	2
5	65.6	39.2	100	14.45	0.2	3
6	61.2	45.3	60	0.63	0.1	1
7	60	53	65.71	0	0.08	1
8	58.3	36.9	100	10.68	0.16	2
9	51.1	38.8	60	0.63	0.1	1
10	44.5	36	60	1.88	0.12	1
11	44.5	31	100	11.94	0.17	2
12	47.8	25.8	69.57	0	0.07	1

节点	X 坐标 /km	Y 坐标/km	通信区域内速度 速度/km · h-1	移动通信上空盘旋 飞行路程/km	通讯区域内所飞 行时间/h	用户数
13	53.9	14.9	100	10.68	0.16	2
14	55.1	22	60	3.14	0.14	1
15	58.4	25.9	100	15.08	0.2	2
16	67.8	31	73.44	0	0.07	1
17	75	27.1	100	15.08	0.2	2
18	73.4	17.5	100	12.57	0.18	2
19	110	0	100	0	0.05	/
非通信区域的无人机飞行时间/h					1.21	
优化最短的路径总长度/km					214.1	
总时间					3.61	

7.4 模型评价

通过敏感性分析，将提高无人机同时传输用户数作为重要的优化目标，同时设计功率的策略为平均分配。而后问题模型依靠聚类分析、枚举法和遗传算法等降低了无人机的使用架次，从而优化了任务时间总和为 4.41 小时。

8 总 结

根据救援任务特点和题设要求,利用无人机携带视频采集装置、生命探测仪等开展灾区巡查和救援任务,结合震区高程资料,设计合理的无人机救援和巡查任务方案,基本完成了四个问题的建模求解与讨论,并对各问题给出了相应的解答和注释。简要总结如下:

(1) 采用网格划分重点区域,并基于贪心算法,将目标巡查区域重组成可由少量无人机完成的目标类群,合理的简化目标布局,减少了无人机的各重点区域间的长途周转,无人机的巡查效能得到最大应用;综合考虑了观察仰角、视野阻碍、山体阻碍路线等约束条件的影响,并进行了数学描述,完善模型,最终制定了无人机巡查方案以及规划了合理的飞行路线:①针对灾情巡查重点中心区域 B、C、D、E 分别派 3、1、1、1 架无人机执行巡查任务,总覆盖率达到 52.85%;②对低于 4000m 的目标区域分任务区,每个任务区 3 架无人机无缝对接巡逻,共派 30 架无人机执行次生灾害巡逻任务,覆盖率为 66.18%。

(2) 改进问题一的模型,通过两次 Q 类聚类分析合理地规划了基地的任务分配和无人机的任务分配,充分考虑路径高度、探测范围、探测距离、探测时间等一系列约束条件的变化,基于遗传算法简化路径,获得最短的任务耗时 5.6 小时,制定出基地各 15 架无人机合理的具体飞行路线及实际行程。

(3) 采用一套静态无人机网络模型,基于依靠聚类分析和最小生成树问题解法,优化了地面移动终端通讯机和中继通讯机的数量,使得完成 12 小时通讯任务所需要的无人机数量最少。相比于动态无人机网络,每架无人机有着更多的工作时长,无人机所需数量也会相应减少。同时,给出无人机的网络路径以及各类无人机的位置部署情况,使用无人机总数为 101 架。

(4) 通过敏感性分析,将提高无人机同时传输用户数作为重要的优化目标,同时设计功率的策略为平均分配。而后问题模型依靠聚类分析、枚举法和遗传算法等降低了无人机的使用架次,从而优化了任务时间总和为 4.41 小时。

9 参考文献

- [1]陆博迪, 孟迪文, 陆鸣, 等, 无人机在重大自然灾害中的应用与探讨[J], 灾害学, 26(4), 122-126, 2011。
- [2]郑锐, 冯振明, 陆明泉, 基于遗传算法的无人机航路规划优化研究[J], 计算机仿真, 28(6), 88-91, 2011。
- [3]周青, 张锐, 索晓杰, 等, 具有时间约束的无人机遗传算法航迹规划[J], 航空计算技术, 46(2), 93-96, 2016。
- [4]司守奎, 孙玺菁, 数学建模算法与应用[M], 北京: 国防工业出版社, 305-312, 2013。
- [5]孙雷, 基于多阶段航迹预测的无人机任务规划方法研究[D], 哈尔滨工业大学, 80-81。
- [6]王化宇, 最小生成树算法及其应用[J], 内蒙古科技与经济, 6, 72-73, 2011。

附录

附录 1 数据处理

```
clc,clear all
[zz,txt,raw]=xlsread('区域高程数据.xlsx');
zz=zz';
x=(0:1:size(zz,2)-1)*38.2/1000; % 在 x 轴上取点
y=(0:1:size(zz,1)-1)*38.2/1000; % 在 y 轴上取 25 点
[xx,yy]=meshgrid(x, y);
surf(xx, yy, z);
contour(xx, yy, z,5)
contourf(xx, yy, zz);%功能填充二维等高线图
axis equal
hold on
P=[30.3 89.8;
   66.0  84.7;
   98.4  76.7;
   73.7  61.0;
   57.9  47.6;
   86.8  22.0;
   93.6  48.8];
hold on
plot(P(:,1),P(:,2),'s',...
     'MarkerEdgeColor','k',...
     'MarkerFaceColor','m',...
     'MarkerSize',10);
S=[110,0];
plot(S(:,1),S(:,2),'s',...
     'MarkerEdgeColor','k',...
     'MarkerFaceColor','g',...
     'MarkerSize',10);
r_tan=10;
for i=1:size(P,1)
    rectangle('Position',[P(i,1)-r_tan,P(i,2)-r_tan,2*r_tan,2*r_tan],
             'Curvature',[1,1],'EdgeColor','g','LineWidth',3);
end
axis equal;
end
```

附录 2 问题 1

2.1 确定 3000m 以下搜索区域

```
%%
%区域标记
clc ,clear all;
load '..\..\数据处理\zz.mat';
zz=zz';
P=[30.3 89.8;
   66.0  84.7;
   98.4  76.7;
   73.7  61.0;
   57.9  47.6;
   86.8  22.0;
   93.6  48.8];
%%
x=(0:1:size(zz,2)-1)*38.2/1000; % 在 x 轴上取点
```

```

y=(0:1:size(zz,1)-1)*38.2/1000; % 在 y 轴上取点
[xx,yy]=meshgrid(x, y);
[c,h]=contourf(xx, yy, zz);
set(h, 'ShowText', 'off', 'LevelList', [0,3000,4150])%设定等高线的值
hold on
plot(P(:,1),P(:,2), 's', ...
     'MarkerEdgeColor', 'k', ...
     'MarkerFaceColor', 'm', ...
     'MarkerSize', 10);
S=[110,0];
plot(S(:,1),S(:,2), 's', ...
     'MarkerEdgeColor', 'k', ...
     'MarkerFaceColor', 'g', ...
     'MarkerSize', 10);
r_tan=10;
for i=1:size(P,1)
    rectangle('Position', [P(i,1)-r_tan,P(i,2)-r_tan,2*r_tan,2*r_tan], 'Curvature', [1,1], 'EdgeColor', 'g', 'LineWidth', 3);
    axis equal;
end
rectangle('Position', [110-120,-120,2*120,2*120], 'Curvature', [1,1], 'EdgeColor', 'k', 'LineWidth', 1);
%%
%找出每个 P 点周围的方格
% zav=sum(zz(1==(zz<=3000)))/sum(sum((zz<=3000)));
zav=3000;
Lc=2*(4150-zav)*tan(30/180*pi)/1000;
W1=Wfind(P(1,1),P(1,2),10,Lc);
W2=Wfind(P(2,1),P(2,2),10,Lc);
W3=Wfind(P(3,1),P(3,2),10,Lc);
W4=Wfind(P(4,1),P(4,2),10,Lc);
W5=Wfind(P(5,1),P(5,2),10,Lc);
W6=Wfind(P(6,1),P(6,2),10,Lc);
W7=Wfind(P(7,1),P(7,2),10,Lc);
%%
%判断方格是否符合条件
W2f=W2(find((W2(:,4)>0.5)==1),:);
W2f=W2f(((W2f(:,1))-110).^2+(W2f(:,2)-60).^2).^0.5<52,:);
W3f=W3(find((W3(:,4)>0.5)==1),:);
W4f=W4(find((W4(:,4)>0.5)==1),:);
W5f=W5(find((W5(:,4)>0.7)==1),:);

Wxy=[W2f(:, [1,2])
     W3f(:, [1,2])
     W4f(:, [1,2])
     W5f(:, [1,2])];
plot(Wxy(:,1),Wxy(:,2), 'r*');
save('Wxy.mat', 'Wxy')
save('.\Wf.mat', 'W2f', 'W3f', 'W4f', 'W5f');
% [sum(W1(:,3)) sum(W2(:,3)) sum(W3(:,3)) sum(W4(:,3))
sum(W5(:,3))]/sum([W1(:,3);W2(:,3);W3(:,3);W4(:,3);W5(:,3);W7(:,3)])
sum([W2f(:,4);W3f(:,4);W4f(:,4);W5f(:,4)])/sum([W1(:,3);W2(:,3);W3(:,3);W4(:,3);W5(:,3);W7(:,3)])
sum([W2f(:,4);W3f(:,4);W4f(:,4);W5f(:,4)])/sum([W2f(:,3);W3f(:,3);W4f(:,3);W5f(:,3)])
fprintf('个区域\n')
%计算覆盖率
sum(W2f(:,4))/sum(W2(:,3))

```

```

sum(W3f(:,4))/sum(W3(:,3))
sum(W4f(:,4))/sum(W4(:,3))
sum(W5f(:,4))/sum(W5(:,3))

```

2.2 划分 3000m 以下巡逻小网格

```

function W= Wfind( x,y,R,Lc )%W=[Wx,Wy,Wf],小网格的 x, y 坐标, 调查范围覆盖
率及可视覆盖率
load '..\..\数据处理\zz.mat'
zz=zz';
W=[];
n=ceil(R/Lc);
for i=-n:1:n
    for j=-n:1:n%划分小网格的范围
        num=0;
        fnum=0;
        xn=x+i*Lc;%小网格的坐标
        yn=y+j*Lc;

        zj=max(1,ceil((xn-Lc/2)*1000/38.2)):min(size(zz,2)-1,floor((xn+Lc/2)*
1000/38.2));%小网格在地图中的可探测点
        zi=ceil((yn-Lc/2)*1000/38.2):floor((yn+Lc/2)*1000/38.2);
        for jj=zj
            for ii=zi
                zx=38.2*(jj-1)/1000;
                zy=38.2*(ii-1)/1000;
                if zz(ii,jj)<=3000&&((zx-x)^2+(zy-y)^2)^0.5<=10
                    num=num+1;

                dn=[-(zz(ii,jj+1)-zz(ii,jj-1))/2/38.2,-(zz(ii+1,jj)-zz(ii-1,jj))/2/38
.2,1];%法向量
                    dl=[xn-zx,yn-zy,(4150-zz(ii,jj))/1000];
                    if
                dn*dl'>=0&&atan(dl(3)/(dl(1)^2+dl(2)^2)^0.5)>=60/180*pi%判断不会被遮挡
和在可视角范围内
                    fnum=fnum+1;
                end
            end
        end
    end
    if num>0
        W=[W;xn,yn,num/((zj(end)-zj(1)+1)*(zi(end)-zi(1)+1)),fnum/((zj(end)-z
j(1)+1)*(zi(end)-zi(1)+1))];
    end
end
end

```

2.3 求取两点直线方程参数

```

function [A B C] = LinPara( D1,D2 )
%UNTITLED2 求出直线一般形式的参数 Ax+By=C
% 此处显示详细说明
A=D2(2)-D1(2);
B=D1(1)-D2(1);
C=D1(1)*D2(2)-D1(2)*D2(1);
end

```

2.4 判断两条线段是否交叉

```

function LinJX = LinJX( L1,L2 )

```

```

%UNTITLED4 判断线段是否交叉
% 此处显示详细说明
[A1 B1 C1]=LinPara(L1(1:2),L1(3:4));
[A2 B2 C2]=LinPara(L2(1:2),L2(3:4));
if A1*B2-A2*B1==0
    LinJX=0;
    return;
end
x0=(C1*B2-C2*B1)/(A1*B2-A2*B1);
if
    (min(L1(1),L1(3))<=x0&&x0<=max(L1(1),L1(3)))&&(min(L2(1),L2(3))<=x0&&
    x0<=max(L2(1),L2(3)))
    LinJX=1;
else
    LinJX=0;
end
end
end

```

2.5 遗传算法搜寻最短路径

```

clc,clear
load '..\Wf.mat'
load '..\W2f.mat'
load shanmai.mat
sj0=W5f(:,[1 2]); %
tic;
J1=[110 60]; %P07
J2=[65 35];
d1=J2;
[M,N]=size(sj0);%得出目标点个数
sj=[d1;sj0;d1]; %加入拐点
d=zeros(M+2); %距离矩阵 d 初始化
for i=1:M+1
    for j=i+1:M+2
        JX=0;
        for s=1:size(shanmai,1)
            JX=LinJX(shanmai(s,:),[sj(i,:) sj(j,)]);
            if JX==1
                d(i,j)=inf;
                break;
            end
        end
        if JX==0
            d(i,j)=((sj(i,1)-sj(j,1))^2+(sj(i,2)-sj(j,2))^2)^0.5;
        end
    end
end
end
d=d+d'; w=100; g=1000; %w 为种群的个数, g 为进化的代数
rand('state',sum(clock)); %初始化随机数发生器
for k=1:w %通过改良圈算法选取初始种群
    c=randperm(M); %产生 1, ..., 100 的一个全排列
    c1=[1,c+1,M+2]; %生成初始解
    for t=1:M+2 %该层循环是修改圈
        flag=0; %修改圈退出标志
        for m=1:M
            for n=m+2:M+1
                if
                    d(c1(m),c1(n))+d(c1(m+1),c1(n+1))<d(c1(m),c1(m+1))+d(c1(n),c1(n+1))
                        c1(m+1:n)=c1(n:-1:m+1); flag=1; %修改圈
                end
            end
        end
    end
end

```



```

        end
    end
    if flag==0
        J(k,c1)=1:M+2; break %记录下较好的解并退出当前层循环
    end
end
end
J(:,1)=0; J=J/M+2; %把整数序列转换成[0,1]区间上的实数,即转换成染色体编码
for k=1:g %该层循环进行遗传算法的操作
    A=J; %交配产生子代 B 的初始染色体
    c=randperm(w); %产生下面交叉操作的染色体对
    for i=1:2:w
        F=2+floor(M*rand(1)); %产生交叉操作的地址
        temp=A(c(i),[F:M+2]); %中间变量的保存值
        A(c(i),[F:M+2])=A(c(i+1),[F:M+2]); %交叉操作
        A(c(i+1),F:M+2)=temp;
    end
    by=[]; %为了防止下面产生空地址,这里先初始化
    while ~length(by)
        by=find(rand(1,w)<0.1); %产生变异操作的地址
    end
    B=A(by,:); %产生变异操作的初始染色体
    for j=1:length(by)
        bw=sort(2+floor(M*rand(1,3))); %产生变异操作的 3 个地址
        B(j,:)=B(j,[1:bw(1)-1,bw(2)+1:bw(3),bw(1):bw(2),bw(3)+1:M+2]); %交换位置
    end
    G=[J;A;B]; %父代和子代种群合在一起
    [SG,ind1]=sort(G,2); %把染色体翻译成 1, ...,102 的序列 ind1
    num=size(G,1); long=zeros(1,num); %路径长度的初始值
    for j=1:num
        for i=1:M+1
            long(j)=long(j)+d(ind1(j,i),ind1(j,i+1)); %计算每条路径长度
        end
    end
    [slong,ind2]=sort(long); %对路径长度按照从小到大排序
    J=G(ind2(1:w),:); %精选前 w 个较短的路径对应的染色体
end
path=ind1(ind2(1),:); flong=slong(1) %解的路径及路径长度
xx=sj(path,1);yy=sj(path,2);
plot([110;xx],[0;yy],'*-','MarkerSize',1) %画出路径
toc
hold on
P=[30.3 89.8;
    66.0 84.7;
    98.4 76.7;
    73.7 61.0;
    57.9 47.6;
    86.8 22.0;
    93.6 48.8];
plot(P(:,1),P(:,2),'s',...
    'MarkerEdgeColor','k',...
    'MarkerFaceColor','m',...
    'MarkerSize',10);
S=[110,0];
plot(S(:,1),S(:,2),'s',...
    'MarkerEdgeColor','k',...

```

```

    'MarkerFaceColor','g',...
    'MarkerSize',10);
r_tan=10;
for i=1:size(P,1)

rectangle('Position',[P(i,1)-r_tan,P(i,2)-r_tan,2*r_tan,2*r_tan],'Cur
vature',[1,1],'EdgeColor','g','LineWidth',1);
    axis equal;
end
for i=1:size(shanmai,1)
    plot(shanmai(i,[1 3]),shanmai(i,[2 4]),'k-','LineWidth',3);
end

```

2.6 确定 4000m 以下搜索区域

```

%%
%区域标记
clc ,clear all;
load '..\..\数据处理\zz.mat';
zz=zz';
x=(0:1:size(zz,2)-1)*38.2/1000; % 在 x 轴上取点
y=(0:1:size(zz,1)-1)*38.2/1000; % 在 y 轴上取点
[xx,yy]=meshgrid(x, y);
[c,h]=contourf(xx, yy, zz);
set(h,'ShowText','off','LevelList',[0,4000,4150])%设定等高线的值
hold on
%%
%找出每个 P 点周围的方格
% zav=sum(zz(1==(zz<=4000)))/sum(sum((zz<=4000)));
zav=2500;
Lc=2*(4150-zav)*tan(30/180*pi)/1000;
R=max(x(end)/2,y(end)/2);
W=Wfind(x(end)/2,y(end)/2,R,Lc);
save('W.mat','W');
%%
%判断方格是否符合条件
load W.mat
Wf=W(find((W(:,4)>0.3)==1),:);
% Wf=Wf(~((Wf(:,1)<40 & Wf(:,2)<70)),:);

save('Wf.mat','Wf');
hold on
plot(Wf(:,1),Wf(:,2),'r*');
sum(Wf(:,3))/sum(W(find((W(:,3))>=1)==1),3)

```

2.7 划分 4000m 以下巡逻小网格

```

function W= Wfind( x,y,R,Lc )%W=[Wx,Wy,Wf],小网格的 x, y 坐标, 调查范围覆盖
率及可视覆盖率率
load '..\..\数据处理\zz.mat'
zz=zz';
W=[];
nx=ceil((size(zz,2)-1)*38.2/1000/Lc/2);
ny=ceil((size(zz,1)-1)*38.2/1000/Lc/2);
k=0;
for i=-nx:1:nx
    k=k+1
    for j=-ny:1:ny%划分小网格的范围
        num=0;
        fnum=0;
        xn=x+i*Lc;%小网格的坐标
        yn=y+j*Lc;

```

```

zj=max(2,ceil((xn-Lc/2)*1000/38.2)):min(size(zz,2)-2,floor((xn+Lc/2)*
1000/38.2));%小网格在地图中的可探测点

zi=max(2,ceil((yn-Lc/2)*1000/38.2)):min(size(zz,1)-2,floor((yn+Lc/2)*
1000/38.2));
    for jj=zj
        for ii=zi
            zx=38.2*(jj-1)/1000;
            zy=38.2*(ii-1)/1000;
            if zz(ii,jj)<=4000
                num=num+1;

dn=[-(zz(ii,jj+1)-zz(ii,jj-1))/2/38.2,-(zz(ii+1,jj)-zz(ii-1,jj))/2/38
.2,1];%法向量
            dl=[xn-zx,yn-zy,(4150-zz(ii,jj))/1000];
            if
dn*dl'>=0&&atan(dl(3)/(dl(1)^2+dl(2)^2)^0.5)>=60/180*pi%判断不会被遮挡
和在可视角范围内
                fnum=fnum+1;
            end
        end
    end
end
if num>0

W=[W;xn,yn,num/((zj(end)-zj(1)+1)*(zi(end)-zi(1)+1)),fnum/((zj(end)-z
j(1)+1)*(zi(end)-zi(1)+1))];
    end
end
end
end

```

2.8 角度结合遗传算法划分搜索区域

```

clc,clear all
load '..\Wf.mat'
load shanmai.mat

% Wf=Wf((Wf(:,2)>35),:);
J1=[110 60];
J2=[65 35];
cc1=170/180*pi;
cc2=180/180*pi;
b(:,1)=Wf(:,1)-J2(1);
b(:,2)=Wf(:,2)-J2(2);
fan=acos(b(:,1)./(b(:,1).^2+b(:,2).^2).^0.5);

sj0=Wf((fan<cc2 & fan>=cc1)|(Wf(:,2)<35),:);
% sj0=Wf((fan<cc2 & fan>=cc1),:);
sj0=sj0(:,[1 2]);

tic;

d1=J2;
[M,N]=size(sj0);%得出目标点个数
sj=[d1;sj0;d1]; %加入拐点
d=zeros(M+2); %距离矩阵d初始化
for i=1:M+1
    for j=i+1:M+2

```

```

JX=0;
for s=1:size(shanmai,1)
    JX=LinJX(shanmai(s,:),[sj(i,:) sj(j,:)]);
    if JX==1
        d(i,j)=inf;
        break;
    end
end
if JX==0
    d(i,j)=((sj(i,1)-sj(j,1))^2+(sj(i,2)-sj(j,2))^2)^0.5;
end
end
end
d=d+d'; w=100; g=1000; %w 为种群的个数, g 为进化的代数
rand('state',sum(clock)); %初始化随机数发生器
J=[];
for k=1:w %通过改良圈算法选取初始种群
    c=randperm(M); %产生 1, ..., 100 的一个全排列
    c1=[1,c+1,M+2]; %生成初始解
    for t=1:M+2 %该层循环是修改圈
        flag=0; %修改圈退出标志
        for m=1:M
            for n=m+2:M+1
                if
d(c1(m),c1(n))+d(c1(m+1),c1(n+1))<d(c1(m),c1(m+1))+d(c1(n),c1(n+1))
                    c1(m+1:n)=c1(n:-1:m+1); flag=1; %修改圈
                end
            end
        end
        if flag==0
            J(k,c1)=1:M+2; break %记录下较好的解并退出当前层循环
        end
    end
end
end
J(:,1)=0; J=J/M+2; %把整数序列转换成[0,1]区间上的实数,即转换成染色体编码
for k=1:g %该层循环进行遗传算法的操作
    A=J; %交配产生子代 B 的初始染色体
    c=randperm(w); %产生下面交叉操作的染色体对
    for i=1:2:w
        F=2+floor(M*rand(1)); %产生交叉操作的地址
        temp=A(c(i),[F:M+2]); %中间变量的保存值
        A(c(i),[F:M+2])=A(c(i+1),[F:M+2]); %交叉操作
        A(c(i+1),F:M+2)=temp;
    end
    by=[]; %为了防止下面产生空地址,这里先初始化
    while ~length(by)
        by=find(rand(1,w)<0.1); %产生变异操作的地址
    end
    B=A(by,:); %产生变异操作的初始染色体
    for j=1:length(by)
        bw=sort(2+floor(M*rand(1,3))); %产生变异操作的 3 个地址
        B(j,:)=B(j,[1:bw(1)-1,bw(2)+1:bw(3),bw(1):bw(2),bw(3)+1:M+2]); %交换位置
    end
    G=[J;A;B]; %父代和子代种群合在一起
    [SG,ind1]=sort(G,2); %把染色体翻译成 1, ..., 102 的序列 ind1

```

```

num=size(G,1); long=zeros(1,num); %路径长度的初始值
for j=1:num
    for i=1:M+1
        long(j)=long(j)+d(ind1(j,i),ind1(j,i+1)); %计算每条路径长度
    end
end
[slong,ind2]=sort(long); %对路径长度按照从小到大排序
J=G(ind2(1:w),:); %精选前 w 个较短的路径对应的染色体
end
path=ind1(ind2(1),:); flong=slong(1); %解的路径及路径长度

xx=sj(path,1);yy=sj(path,2);
plot([110;xx],[0;yy],'*-','MarkerSize',2) %画出路径
hold on
S=[110,0];
plot(S(:,1),S(:,2),'s',...
     'MarkerEdgeColor','k',...
     'MarkerFaceColor','g',...
     'MarkerSize',10);
JJ=[J1;J2];
plot(JJ(:,1),JJ(:,2),'s',...
     'MarkerEdgeColor','k',...
     'MarkerFaceColor','b',...
     'MarkerSize',10);
for i=1:size(shanmai,1)
    plot(shanmai(i,[1 3]),shanmai(i,[2 4]),'k-','LineWidth',3);
end
toc

```

附录 3 问题 2

3.1 确定生命探测区域

```

%%
%区域标记
clc ,clear all;
load '..\数据处理\zz.mat';
zz=zz';
x=(0:1:size(zz,2)-1)*38.2/1000; % 在 x 轴上取点
y=(0:1:size(zz,1)-1)*38.2/1000; % 在 y 轴上取点
[xx,yy]=meshgrid(x, y);
[c,h]=contourf(xx, yy, zz);
set(h,'ShowText','off','LevelList',[0,3000,7000])%设定等高线的值
axis equal
hold on
%%
%找出每个 P 点周围的方格
Lc=2*500*1.732/1000;
R=max(x(end)/2,y(end)/2);
W=Wfind(x(end)/2,y(end)/2,R,Lc);
save('W.mat','W');
%%
%判断方格是否符合条件
load W.mat
Wf=W(find((W(:,4)>0.9)==1),:);
Wxy=Wf(:,[1,2]);
Wxy=Wxy(2:end,:);
save('Wxy.mat','Wxy');
plot(Wxy(:,1),Wxy(:,2),'*');
sum(Wf(:,4))/sum(W(:,3))

```

3.2 探测区域二聚类

```
%%
clc,clear
load Wxy.mat
b=Wxy;
a1=[b(:,1)-110 b(:,2)];
a1=(a1(:,1).^2+a1(:,2).^2).^0.5;
a2=[b(:,1)-110 b(:,2)-55];
a2=(a2(:,1).^2+a2(:,2).^2).^0.5;
a=[a1 a2];
y=pdist(a,'chebychev'); %求 a 的两两行向量间 chebychev 距离
yc=squareform(y); %转换成距离方阵
z=linkage(y,'complete'); %产生等级聚类树
[h,t]=dendrogram(z); %画聚类图
%%
TM={};
T=cluster(z,'maxclust',2); %把对象划分成 3 类
for i=1:2
    tm=find(T==i); %求第 i 类的对象
    tm=reshape(tm,1,length(tm)); %变成行向量
    WW=Wxy(tm,:);
    figure(2)
    hold on
    plot(WW(:,1),WW(:,2),'*')
    TM{i}=WW;
    fprintf('第%d类的有%s\n',i,int2str(tm)); %显示分类结果
end
save('TM.mat','TM');
```

3.1 每区域 15 聚类

```
clc,clear
load TM.mat
load Wxy.mat
a=TM{1,2};
aa=[a(:,1)-110 a(:,2)];
y=pdist(aa,'cosine'); %求 a 的两两行向量间的欧氏值距离
yc=squareform(y); %转换成距离方阵
z=linkage(y,'complete'); %产生等级聚类树
[h,t]=dendrogram(z); %画聚类图
%%
T=cluster(z,'maxclust',15) %把对象划分成 3 类
TM1={};
for i=1:15
    tm=find(T==i); %求第 i 类的对象
    tm=reshape(tm,1,length(tm)); %变成行向量
    WW=a(tm,:);
    TM1{i}=WW;
    figure(2)
    hold on
    plot(WW(:,1),WW(:,2),'*')
    fprintf('第%d类的有%s\n',i,int2str(tm)); %显示分类结果
end
```

附录 4 问题 3

4.1 最小生成树

```
clc,clear all
load D.mat
aa=D(:,[1 2]);
```

```

y=pdist(aa,'euclidean'); %求 a 的两两行向量间的欧氏值距离
z=linkage(y,'complete'); %产生等级聚类树
[h,t]=dendrogram(z); %画聚类图
%%
%最小生成树
P=[1:3 5:10 12:16 18 20 22:41 43 44 46:65 67 68 70 71];
P=aa(P,:);
P=[P;
    (aa(45,:)+aa(69,:))/2
    (aa(19,:)+aa(21,:))/2
    (aa(42,:)+aa(72,:))/2
    (aa(4,:)+aa(11,:))/2
    (aa(17,:)+aa(66,:))/2];
a=squareform(pdist(P,'euclidean')); %求 a 的两两行向量间的欧氏值距离
a(a==0)=inf;
result=[];p=1;tb=2:length(a);
while size(result,2)~=length(a)-1
    temp=a(p,tb);temp=temp(:);
    d=min(temp);
    [jb,kb]=find(a(p,tb)==d);
    j=p(jb(1));k=tb(kb(1));
    result=[result,[j;k;d]];
    p=[p,k];
    tb(find(tb==k))=[];
end
result=result';

```

4.2 中继无人机位置

```

%%
%划分中继通信无人机
S=[];
for i=1:size(result,1)
    x=[P(result(i,1),1) P(result(i,2),1)];
    y=[P(result(i,1),2) P(result(i,2),2)];
    if 6<result(i,3)&&result(i,3)<=12
        S=[S;[sum(x)/2,sum(y)/2]];
    else
        if 12<result(i,3)&&result(i,3)<=18
            S=[S;
                x(1)/3+x(2)*2/3,y(1)/3+y(2)*2/3;
                x(1)*2/3+x(2)/3,y(1)*2/3+y(2)/3];
        else
            if 18<result(i,3)&&result(i,3)<=24
                S=[S;
                    x(1)/4+x(2)*3/4,y(1)/4+y(2)*3/4;
                    x(1)*2/4+x(2)*2/4,y(1)*2/4+y(2)*2/4;
                    x(1)*3/4+x(2)*1/4,y(1)*3/4+y(2)*1/4
                ];
            else
                if 24<result(i,3)
                    S=[S;
                        x(1)*1/5+x(2)*4/5,y(1)*1/5+y(2)*4/5;
                        x(1)*2/5+x(2)*3/5,y(1)*2/5+y(2)*3/5;
                        x(1)*3/5+x(2)*2/5,y(1)*3/5+y(2)*2/5;
                        x(1)*4/5+x(2)*1/5,y(1)*4/5+y(2)*1/5
                    ];
                end
            end
        end
    end
end
plot(x,y,'-*k','MarkerSize',5);

```

```

        hold on
        axis equal;
    end
    plot(S(:,1),S(:,2),'s','MarkerSize',5,'MarkerFaceColor','g');
    %%
    r_tan=1;
    for i=size(P,1)-4:size(P,1)

        rectangle('Position',[P(i,1)-r_tan,P(i,2)-r_tan,2*r_tan,2*r_tan],'Curvature',[1,1],'EdgeColor','b','LineWidth',1);
        axis equal;
    end
end

```

附录 5 问题 4

5.1 飞行及盘旋时间求取

```

clc,clear all
load D.mat
load JHD.mat
load jh.mat
load ce.mat
load path1.mat
load path2.mat
load path3.mat
D(:,4)=3*1.732./(500*10^6./(D(:,3)*1000*log2(1+5./(4.314*10^-10*3000^2))))/3600);
j=1;
for i=1:size(D,1)
    if sum(jh==i)==0
        if D(i,4)>60
            DD(j,:)=[D(i,[1 2 4]) 0];
        else
            n=ceil((60*(500*10^6./(D(i,3)*1000*log2(1+5./(4.314*10^-10*3000^2))))/3600)-3*1.732)/0.200/pi);
            DD(j,:)=[D(i,[1 2]) 60 n*0.200*pi];
        end
        j=j+1;
    end
end
JHD=[JHD zeros(size(JHD,1),2)];
for i=1:size(JHD,1)
    n=ceil(max((500*10^6./(D(ce{i},3)*1000*log2(1+5/size(ce{i},2))./(4.314*10^-10*3000^2))))/3600))*100/0.2/pi);
    JHD(i,:)=[JHD(i,[1 2]) 100 n*0.200*pi];
end
DD=[DD;JHD];
%飞机路径标号求取
load fani.mat
path=double(fani{3}).*(1:47)';
path(find(path==0))=[];
DD(path(path3(2:end-1)-1),:);
r_tan=1;
for i=1:size(D,1)

    rectangle('Position',[D(i,1)-r_tan,D(i,2)-r_tan,2*r_tan,2*r_tan],'Curvature',[1,1],'EdgeColor','b','LineWidth',1);
        axis equal;
    end
end

```


5.2 贪心法划分区域

```
clc,clear all
load D.mat
load JHD.mat
load jh
j=1;
tic;
for i=1:size(D,1)
    if sum(jh==i)==0
        DD(j,:)=D(i,[1 2]);
        j=j+1;
    end
end
DD=[DD;JHD];

J1=[110 0];

b(:,1)=DD(:,1)-J1(1);
b(:,2)=DD(:,2)-J1(2);
fan=acos(b(:,1)./((b(:,1).^2+b(:,2).^2)).^0.5);
ccmin=min(fan);
ccmax=max(fan);
ccall=linspace(ccmin,ccmax,5);
Key=[0,0,inf,inf,inf];
for cc1=2.0186%ccall
    for cc2=2.3163%ccall(ccall>cc1)
        % cc1=115/180*pi;
        % cc2=130/180*pi;
        fani{1}=90/180*pi<fan & fan<=cc1;
        fani{2}=cc1<fan & fan<=cc2;
        fani{3}=cc2<fan & fan<=180/180*pi;
        for cci=1:3
            sj0=DD(fani{cci},:);
            sj0=sj0(:,[1 2]);

            d1=J1;
            [M,N]=size(sj0);%得出目标点个数
            sj=[d1;sj0;d1]; %加入拐点
            d=zeros(M+2); %距离矩阵d初始化
            for i=1:M+1
                for j=i+1:M+2
                    % JX=0;
                    % for s=1:size(shanmai,1)
                    % JX=LinJX(shanmai(s,:),[sj(i,:)
sj(j,:)]);
                    % if JX==1
                    % d(i,j)=inf;
                    % break;
                    % end
                    % end
                    % if JX==0
d(i,j)=((sj(i,1)-sj(j,1))^2+(sj(i,2)-sj(j,2))^2)^0.5;
                    % end
                end
            end
            end
            d=d+d'; w=100; g=1000; %w 为种群的个数, g 为进化的代数
            rand('state',sum(clock)); %初始化随机数发生器
            J=[];
```

```

for k=1:w %通过改良圈算法选取初始种群
    c=randperm(M); %产生 1, ..., 100 的一个全排列
    c1=[1,c+1,M+2]; %生成初始解
    for t=1:M+2 %该层循环是修改圈
        flag=0; %修改圈退出标志
        for m=1:M
            for n=m+2:M+1
                if
d(c1(m),c1(n))+d(c1(m+1),c1(n+1))<d(c1(m),c1(m+1))+d(c1(n),c1(n+1))
                    c1(m+1:n)=c1(n:-1:m+1); flag=1; %修改圈
                end
            end
        end
        if flag==0
            J(k,c1)=1:M+2; break %记录下较好的解并退出当前层循环
        end
    end
end
J(:,1)=0; J=J/M+2; %把整数序列转换成[0,1]区间上的实数,即转换成染色
体编码

for k=1:g %该层循环进行遗传算法的操作
    A=J; %交配产生子代 B 的初始染色体
    c=randperm(w); %产生下面交叉操作的染色体对
    for i=1:2:w
        F=2+floor(M*rand(1)); %产生交叉操作的地址
        temp=A(c(i),[F:M+2]); %中间变量的保存值
        A(c(i),[F:M+2])=A(c(i+1),[F:M+2]); %交叉操作
        A(c(i+1),F:M+2)=temp;
    end
    by=[]; %为了防止下面产生空地址,这里先初始化
    while ~length(by)
        by=find(rand(1,w)<0.1); %产生变异操作的地址
    end
    B=A(by,:); %产生变异操作的初始染色体
    for j=1:length(by)
        bw=sort(2+floor(M*rand(1,3))); %产生变异操作的 3 个地址
        B(j,:)=B(j,[1:bw(1)-1,bw(2)+1:bw(3),bw(1):bw(2),bw(3)+1:M+2]); %交换位
置

        end
        G=[J;A;B]; %父代和子代种群合在一起
        [SG,ind1]=sort(G,2); %把染色体翻译成 1, ..., 102 的序列 ind1
        num=size(G,1); long=zeros(1,num); %路径长度的初始值
        for j=1:num
            for i=1:M+1
                long(j)=long(j)+d(ind1(j,i),ind1(j,i+1)); %计算每条
路径长度

            end
        end
        [slong,ind2]=sort(long); %对路径长度按照从小到大排序
        J=G(ind2(1:w),:); %精选前 w 个较短的路径对应的染色体
    end
    path=ind1(ind2(1),:); flong(cci)=[slong(1)]; %解的路径及路径
长度

    xx=sj(path,1);yy=sj(path,2);

```

```

        plot(xx,yy, '*-', 'MarkerSize',2) %画出路径
        hold on
    end
    if max(flong)<max(Key(3:5))
        Key=[cc1,cc2,flong]
    end
end
end
end
S=[110,0];
plot(S(:,1),S(:,2),'s',...
     'MarkerEdgeColor','k',...
     'MarkerFaceColor','g',...
     'MarkerSize',10);
JJ=[J1];
plot(JJ(:,1),JJ(:,2),'s',...
     'MarkerEdgeColor','k',...
     'MarkerFaceColor','b',...
     'MarkerSize',10);
toc

```

