

**“华为杯”第十四届中国研究生
数学建模竞赛**

西安理工大学

学 校

10700045

参赛队号

1.王 远

队员姓名

2.刘曙铭

3.孙元元

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：

随着无人机的出现，及时高效的灾后救援成为一种重要手段，本文就无人机在抢险救灾中的进行研究。

针对问题一：根据题中给出的数据，做出等高线图，并着重分析了七个重点区域内的巡查区的面积以及平均海拔高度，通过最优分配思想，建立了非线性规划模型，采用贪心算法，得到飞机的最少架数 5 架，覆盖率达到 94.44% 以及每架无人机的飞行路线（见图 3）。对于第二小问，首先，应用等分法将探测区域划分为 23 份，建立线性规划模型，采用禁忌搜索算法求得最优分区（见图 5），使得每架无人机在海拔 4000 以下的区域执行任务的时间均接近 8 小时。可以每隔三小时起飞一架无人机，使得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，从而得到最优结果为 69 架。

针对问题二：首先通过均分法对 3000 以下区域进行扇形分区，对目标区域分成角度相同的 30 个扇形区域，使无人机在每个扇形域完成任务的时间基本相同。建立线性规划模型，采用模拟退火算法，求得每个扇形域最短探测路径和时间（见表 4、表 5），以最短时间的均衡率为指标，调整各区域任务点的数量，直

到各区域探测时间达到均衡，得到最短时间为 227.29 分钟。

针对问题三：首先，假设任意地面终端均可自由移动 0~2000 米，应用无人机构建通信网来覆盖所有地面终端的移动区域，建立线性规划模型，采用蚁群算法求解得到无人机最小架数为 116 台以及飞行路线（见图 9）。其次，假设任意地面终端可接收信息，并向接收到的信号点移动，以 72 个地面终端移动方向和距离作为变量，建立线性规划模型，采用狼群算法得到最优路线（见图 10）从而得到对应的无人机架数 75 台。

针对问题四：针对 3 架无人机完成对 72 个地面终端数据传输任务所用时间总和最短，通过均分法将所有地面终端分为三个区域，建立线性规划模型，采用蚁群算法对三个区域的地面终端进行路线划分，考虑到蚁群算法的随机性，建立非线性规划模型对结果进行小幅度优化，并求解得到最优传输高度以及水平速度，考虑不同区域地面终端的稀疏程度，对每个用户分配恰当的子信道分配的功率以及速度（见表 6），三架无人机同时出发，最终得到三个区域传输任务最优路线（见图 14）以及任务完成时间分别为 3.7445h，3.1720h，3.667h。最终使得无人机完成任务的时间总和为 3.7445h。

关键字：线性规划；模拟退火算法；灰狼算法；蚁群算法

目录

1 问题重述	1
2 问题分析	3
3 问题假设	4
4 变量说明	4
5 模型的建立与求解	5
5.1 问题 1 的建立与求解	5
5.1.1 模型分析	5
5.1.2 问题 1 的建立	7
5.1.3 问题 1 的求解	9
5.2 问题 2 的建立与求解	13
5.2.1 问题 2 的建立	13
5.2.2 问题 2 的求解	15
5.2.3 结果分析	18
5.3 问题 3 的建立与求解	18
5.3.1 问题 3 的建立	18
5.3.2 问题 3 的求解	19
5.3.3 结果分析	22
5.4 问题 4 的建立与求解	22
5.4.1 问题 4 的建立	22
5.4.3 结果分析	29
6 模型的评价与改进	31
6.1 模型的优点	31
6.2 模型的缺点	32
6.3 模型的推广	32
7 参考文献	32
附录	33

1 问题重述

2017 年 8 月 8 日,四川阿坝州九寨沟县发生 7.0 级地震,造成了不可挽回的人员伤亡和重大的财产损失。由于预测地震比较困难,及时高效的灾后救援是减少地震损失的重要措施。无人机作为一种新型运载工具,能够在救援行动中发挥重要作用。为提高其使用效率,请你们解决无人机优化运用的几个问题。

附件 1 给出了震区的高程数据,共有 2913 列,2775 行。第一行第一列表示 $(0,0)$ 点处的海拔高度值(单位:米),相邻单元格之间的距离为 38.2 米,即第 m 行第 n 列单元格中的数据代表坐标 $(38.2(m-1), 38.2(n-1))$ 处的高度值。震区 7 个重点区域的中心位置如附件中的表所示(单位:千米)。

除另有说明外,本题中的无人机都假设平均飞行速度 60 千米/小时,最大续航时间为 8 小时,飞行转弯半径不小于 100 米,最大爬升(俯冲)角度为 $\pm 15^\circ$,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。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。

问题一: 灾情巡查

大地震发生后,及时了解灾区情况是制订救援方案的重要前提。为此,使用无人机携带视频采集装置巡查 7 个重点区域中心方圆 10 公里(并集记为 S)以内的灾情。假设无人机飞行高度恒为 4200 米,将在地面某点看无人机的仰角大于 60° 且视线不被山体阻隔视为该点被巡查。若所有无人机均从基地 $H(110,0)$ (单位:千米)处派出,且完成任务后再回到 H ,希望在 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,最少需要多少架无人机?覆盖率是多少?每架无人机的飞行路线应如何设计?在论文中画出相应的飞行路线图及巡查到的区域(不同的无人机的飞行路线图用不同的颜色表示)。

进一步,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。问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(无人机均需从 H 出发并在 8 小时内回到 H ,再出发的时间间隔不小于 1 小时)。

问题二: 生命迹象探测

使用无人机携带生命探测器搜索生命迹象,能够给灾后救援提供准确的目标定位。拟从基地 $H(110,0)$, $J(110,55)$ (单位:千米)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。探测仪的有效探测距离不超过 1000 米,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° 。请你们规划它们的飞行路线,使附件 1 所给出的全区域内海拔 3000 米以下部分能被探测到的面积尽可能大,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。

问题三: 灾区通信中继

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(白天不受续航能力限制,其余条件同前述)可以作为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(地面终端只能与无人机进行通信,无人机之间只要不超过最大通信距离就可以互相通信,地面与地面之间的通信由无人机转接)。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,问最少需要多少架无人机、每架无人机的飞行路线如何,才能保证在白天 12 小时内,附件 2 中的任意两个地面终端之间都能实现不间断通信(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)。

问题四: 无人机对地的数据传输

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(分布见附件 2)发送内容不同,总量均为 500M(1M 按 10^6 比特计算)的数据。设每台通信装备的总功率是 5 瓦,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 3000 米且没有山体阻隔时,如果无人机当前服务用户少于 10 个,则开始时向 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,则为 i 服务(先到先服务)。如果在一个服务时间区间(即无人机和终端之间满足可传输数据条件的时间范围)内不能传完全部数据,则以后区间可以续传。再设某某 i 用户在时刻 t 接收到无人机发送的信息速率为

$$r_i(t) = B_i \log_2 \left(1 + \frac{p_i(t)}{\rho_0 d^2(u,i)} \right) \text{ (比特/秒)}, \text{ 其中 } B_i \text{ 表示无人机服务 } i \text{ 的子信道带宽为}$$

(取值见附件 2, 单位 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,单位: w(瓦), $d(u,i)$ 表示 t 时刻无人机与 i 之间的欧氏距离,单位:

米。 ρ_0 为信道特性参数,为简单起见,取为 4.314×10^{-10} (单位略),假设无人机飞行速度在 60~100 千米/小时之间可调(水平面内最大加速度 ± 5 米/秒²,铅垂面内最大加速 ± 2 米/秒²,可同时在两个方向上加速),请你们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,使得无人机完成所有任务的时间总和尽量短。

2 问题分析

问题一:对于第一小问,对应的无人机需求问题转化为单架无人机多次执行任务,因此我们可以将最优路径和区域扫描时间分离,即先计算出每个区域的最优扫描时间,然后根据各点的扫描时间,采用贪心算法分配无人机多次执行任务,当完成任务时,单架无人机执行任务次数即为所需无人机数量。对于第二小问,为了满足题中条件,不妨先将巡查区域划分,在每个巡查子区域内,使得无人机每次巡查所需时间接近 8 小时。假设无人机再出发时间为 1 小时,那么,单架无人机执行一次任务的周期为 9 小时,为了保证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,我们只需在每个巡查子区间内每隔三小时分配一次无人机即可。

问题二:本题中,从基地 H,J 派出的无人机数量相同,所执行的任务是有效探测海拔 3000 米以下的生命迹象。为了使多无人机执行任务的时间尽可能短,即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可能短。不妨假设无人机同时从基地 H,J 飞出,那么最后一架完成任务的无人机回到基地的时间,即为多无人机协同任务执行的总时间。对此我们首先进行区域划分,对每个子区域分别运用模拟退火算法求得其最短探测路径和最短探测时间。调整各区域内的任务点的数量,使得任务达到均衡。

问题三:为了使 72 个移动终端实现不间断通信,需要考虑的是路线分配,使得分配的路线总长度较小,若无人机的任务是作为通信中继保证地面人员之间信息及时准确的传输。在无人机执行中继任务之间,必须找到地面终端,才能建立起通信网络进行通信中继。因此本文根据任务需要将无人机分成两种状态:搜索状态和中继状态。对于搜索地面终端区域,本文将无人机的搜索方式分为两种情况,(1)假设地面终端用户可以移动,这种情况下无人机之间的扫描区域存在相交区域,信号覆盖区域包括 72 个移动终端的随机移动范围 (2km), (2)假设

地面终端用户可以按接收信息进行移动，这种情况无人机之间的扫描区域相切。把每个移动终端的移动距离看做变量进行求解，使得扫描路线总长度最小。

问题四：为了设计出三架无人机对地面终端信息的最优传输路线以及对所服务的每一个用户分配的功率，使得无人机完成所有的任务所用的时间综合尽量短。首先要使得 72 个地面终端用户传输信息达到比较均衡，应使每架飞机的巡航时间基本相同，进行子图划分，所有用户分为 3 个区域，应用蚁群算法将每个区域中的点与起始点进行连接，找到最短路线。因为蚁群算法具有随机性，因此再对之前的蚁群算法进行小幅度的优化，得到了信息传输的最佳路径。然后，对每个区域中相邻用户之间的垂直加速距离与水平加速距离进行分析，调整无人机的垂直和水平加速度。再根据每个用户传输的最优高度，分配恰当的功率，最后得到无人机完成所有与任务的最短时间。

3 问题假设

- 假设 1：不考虑气候对无人机飞行和执行任务时性能的影响；
 假设 2：无人机视为一个质点；
 假设 3：不考虑由于转弯、转弯、爬升、俯冲等机动动作带来的燃料损耗对于航行时间的减小。
 假设 4：地图外的坐标点海拔均为 0；
 假设 5：题设中参数以及事件发生情况均真实可靠；

4 变量说明

符号	符号含义
aim_{ij}	72个地面终端的坐标
aim_a_{ij}	A 组的地面终端坐标
aim_b_{ij}	B 组的地面终端坐标
aim_c_{ij}	C 组的地面终端坐标
$D_{ij,a}$	A 组的地面终端 i 与地面终端 j 的距离

$D_{ij,b}$	B 组的地面终端 i 与地面终端 j 的距离
$D_{ij,b}$	C 组的地面终端 i 与地面终端 j 的距离
L_a	A 组内最小路径
L_b	B 组内最小路径
L_c	C 组内最小路径

注：其他符号见正文具体说明。

5 模型的建立与求解

5.1 问题 1 的建立与求解

5.1.1 模型分析

对于灾区的救援工作，无人机的出现，大大减少了伤亡率，无人机对灾区救援的目的是为了在最短的时间内最大化的捕捉有效的信息。根据救援前侦察任务的要求，需要巡查 7 个重点区域中心方圆 10 公里（并记为 S）以内的灾情。为了更直观的了解重点区域距基地 H 的方位以及其地势情况，首先做出地势高度图（见图 1），图中白色部分表示海拔低于 3000 米（巡查区），红色部分表示海拔高于 4200 米（禁飞区），蓝色部分海拔介于两者之间。

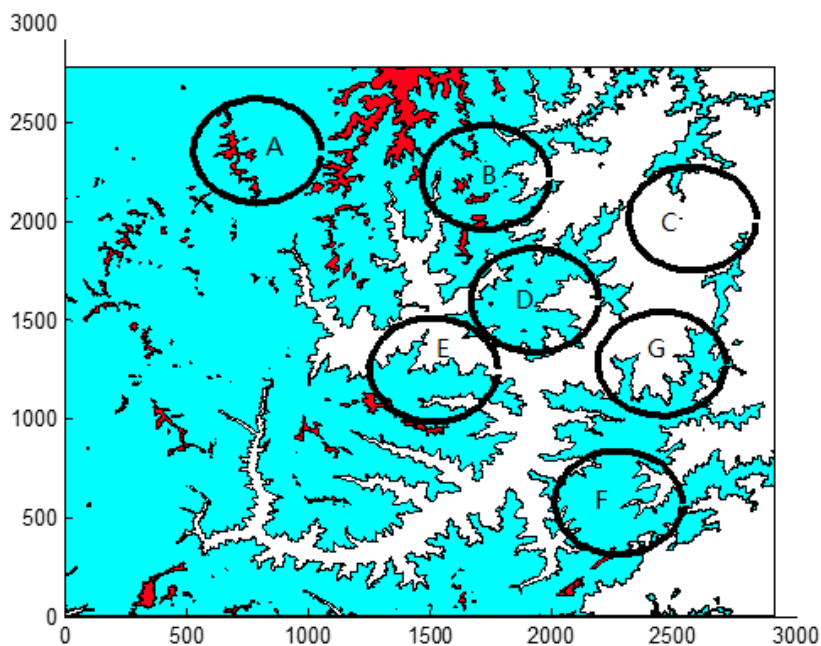


图 1 7 个重点区域的地势等高图

从图 1 可以看出，海拔低于 3000 米的区域主要集中在 C, E, G，由式

$$\gamma_i = \frac{S_{\text{区域内海拔低于3000米点数}}^i}{S_{\text{区域内所有坐标点数}}^i} \times 100\%, \quad i = 1, 2, \dots, 6$$

$$\Delta H = \sqrt{\frac{\sum_{i=1}^n \sum_{j=1}^n h_{ij}}{n^2}}$$

可以计算出各区域巡查区面积占比 γ_i 和各区域平均海拔（见表 1）。由于 A 区域海拔低于 3000 米的点仅占整块区域的 0.36%，其距离无人机的出发点 H 大约 120.0672 米，所需探测时间为 240.1344 分钟，已经超过了时间规定的 4 小时，同时派去无人机扫描 A 区域所携带回来的信息收益较小，对重点区域的总探测收益没有更大的影响。因此本问对 A 区域不进行扫描。将大量的时间用在剩余的 6 个重点区域的详细扫描上。

表 1 各区域巡查区域面积占比与平均海拔高度

区域	B	C	D	E	F	G
巡查区面积占比	18.87%	94.91%	34.25%	45.60%	25.33%	63.81%
平均海拔	3523.1	2144.9	3236.0	3019.9	3337.1	2700.6

假设无人机起飞点为 2200 米，扫描时飞行高度恒为 4200 米，对于 C, E, G 面积占比较大的区域的扫描方式都如图 2，对于 B, D, F 由于区域中心基本没有 3000 米以下区域，故采用围绕区域外围扫描的方式。

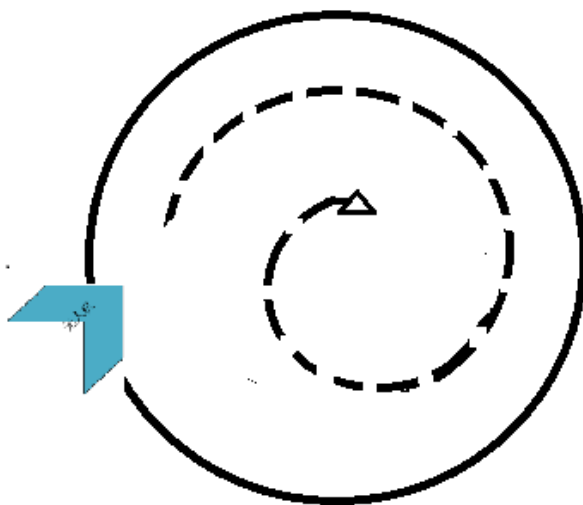


图 2 无人机对重点区域内的扫描方式

无人机最大侧视角为 60 度，则扫描半径随着海拔高度的变化而变化。由于无人机飞行高度大于扫描区域海拔高度，且禁飞区无人机不去扫描，可以假设无人机扫描过程中地势没有出现险峻的陡峭，即扫描过程中符合最大爬升(俯冲)角度为 $\pm 15^\circ$ ，那么，我们可以采用各区域平均海拔高度（见表 1）来计算无人机在各区域最大扫描面积

$$\Delta h = \sqrt{\frac{\sum_{i=1}^n \sum_{j=1}^n (4200 - h_{ij})}{n^2}} \quad (1)$$

式 1 中 Δh 为平均相对高度（无人机飞行海拔高度与地面海拔高度之间的高度差）， $h_{ij}, i=1, 2, \dots, N_i, j=1, 2, \dots, N_j$ 表示区域内各坐标点海拔高度，扫描过程中，最大侧视角为 60 度，可知最大扫描半径 R_{\max} 为

$$R_{\max} = \cos 60^\circ \Delta h \quad (2)$$

接下来模型的建立以及求解中，对区域扫描一直采用上述最大扫描半径。

5.1.2 问题 1 的建立

针对第一小问

本文将问题一中对应的无人机需求问题转化为单架无人机多次执行任务，因此我们可以将最优路径和区域扫描时间分离，即先计算出每个区域的最优扫描时间，然后根据各点的扫描时间分配无人机多次执行任务，当完成任务时，单架无人机执行任务次数即为所需无人机数量。

a. 目标函数

假设单架无人机扫描 n 个点，出发去 i 点飞行的时间为 $v_i, i=1, 2, \dots, n$ ，每个点需要扫描的时间为 $w_i, i=1, 2, \dots, n$ ，每次执行任务的时间为 240min。到达 i 点时，可以选择扫描 i 点的部分区域，而不一定要全部扫描。为了完成任务且采用最少的无人机，我们不妨将问题转化为让单架无人机执行任务的时间在规定范围内尽可能大。那么

$$\max \sum_{i=1}^n (t_i + t_i^1 x_i) \quad (3)$$

式中， x_i 为第 i 点扫描的比例。

b. 约束规则

1) 每次执行任务时间

由于无人机每次执行任务必须在 240min 内返回基地 H，故需要对其进行约束

$$\max \sum_{i=1}^n (t_i + t_i^1 x_i) \leq 240$$

2) 扫描时间

$$\sum_{i=1}^n t_i^1 x_i \leq c \quad c > 0 \quad (4)$$

式中， c 为各区域最优扫描时间总和

3) 扫描比例

$$0 \leq x_i < 1, \quad i = 1, 2, \dots, n$$

$$4) \quad t_i^1 > 0, \quad t_i > 0, \quad c > 0$$

针对第二小问

为了满足题中条件，不妨先将巡查区域分为 k 份，在每个巡查子区域内，使得无人机每次巡查所需时间接近 8 小时。假设无人机再出发时间为 1 小时，那么，单架无人机执行一次任务的周期为 9 小时，为了保证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，我们只需在每个巡查子区间内每隔三小时分配一次无人机即可。

a. 目标函数

假设每架无人机巡查子区域 $G_i, i = 1, 2, \dots, k$ 所需时间为 $t_i \text{ min}$ ，则要保证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，区域 G_i 需要 s_i 架无人机来覆盖，那么至少需要 f_s 架无人机才可以巡查整个区域 G

$$\min f_s = \sum_{i=1}^k s_i$$

b. 约束规则

1) 续航时间

$$\frac{t_i}{60} \leq 8, i = 1, 2, \dots, k,$$

2) 至少间隔三小时巡查

$$\frac{t_i}{60s_i} \leq 3, i = 1, 2, \dots, k$$

3) $s_i > 0$, $t_i > 0$

5.1.3 问题 1 的求解

针对第一小问：

为了保证无人机在规定时间内（4 小时之内）使区域 S 内海拔 3000 米以下的地方尽可能多的被巡查到，不妨先依次给 6 个重点区域编号为 1,2...6。在此基础上，提出了采用信息确定性指标来衡量特定时间内无人机对重点区域的探测收益，信息确定性指标主要和无人机在重点区域的探测时间、探测载荷的工作能力等有关，具体如下式所示：

$$G(t) = G_0 + G_1(1 - e^{-\beta t}) \quad (5)$$

式中， G_0 表示探测开始前无人机对重点区的已知的信息， $0 \leq G_0 \leq 1$ ， G_1 表示无人机对惹怒我去的信息不确定性部分，应满足 $G_0 + G_1 = 1$ ， β 表示无人机携带的探测载荷对重点区进行探测的能力指数，主要由探测载荷的股友能力、待探测重点区的特性以及无人机的性能决定。在此假设 $G_0 = 0$ 的情况下，

$$\beta = \frac{s'}{s} \quad (6)$$

式中 s' 表示无人机单位时间的有效搜索面积， s 表示任务区的面积大小。 又可将无人机单位时间有效搜索面积 s' 表示为：

$$s' = w \cdot v \quad (7)$$

式中， w 表示无人机对重点区的有效扫描宽度， v 表示无人机的飞行速度。综上所述，可把 （8）式表达成如下所示：

$$\beta = \frac{w \cdot v}{s} \quad (8)$$

已知无人机的飞行速度 $v=60\text{km/h}$ ，有效扫描宽度采用最大扫描半径 R_{\max} ，扫描方法采用第一节中给出的方式，可以计算出各区域扫描时间，和无法扫描区域占比

表 2 各区域扫描时间（单位：分钟）和无法扫描区域占比

区域	B	C	D	E	F	G
扫描时间	28.200	105.912	76.544	81.176	63.056	92.4080
无法扫描区域占比	0.68%	0%	4.11%	9.35%	1.55%	0%

从表 2 中可以看出, C, G 等区域扫描时间较长, 如果让单架飞机去扫描, 规定的 4 小时之内, 无法将整个区域扫描完毕, 可以假设一个区域可以多次去扫描。结合扫描时的最优路径, 我们可以采用贪心算法, 来最大限度的让单架无人机在规定时间内每次执行任务时收益最大, 从而使完成任务的总次数最少。

5.1.3.1 贪心算法

所谓贪心算法是指, 在对问题求解时, 总是做出在当前看来是最好的选择。也就是说, 不从总体最优上加以考虑, 他所做出的仅是在某种意义上的局部最优解。贪心算法的基本思路是指将问题的求解过程看作是一系列选择, 每次选择一个输入, 每次选择都是当前状态下的最好的选择 (局部最优解)。每做一次选择后, 所求问题会简化为一个规模更小的子问题。从而通过每一步的最优解逐步达到整体的最优解。

贪心算法求解步骤

- (1) 建立对问题精确描述的数学模型, 包括定义最优解的模型;
- (2) 将问题分解为一系列子问题, 同时定义子问题的最优解结构;
- (3) 应用贪心原则确定每个子问题的局部最优解, 并根据最优解的模型, 用子问题的局部最优解堆叠出全局最优解。

本问题与小数背包问题类似, 故在此不证明贪心算法的最优性。

5.1.3.2 结果分析

根据贪心算法, 最终求得当 $x_i = 1, i = 1, 2, \dots, 6$ 时, 无人机执行任务的次数即为我们所求最优值, 故至少需要 5 架无人机, 飞行路线分别为

表 3 无人机飞行路线

无人机	飞行路线
A	$H \rightarrow F \rightarrow G \rightarrow H$
B	$H \rightarrow G \rightarrow D \rightarrow H$
C	$H \rightarrow D \rightarrow B \rightarrow H$
D	$H \rightarrow E \rightarrow H$
E	$H \rightarrow C \rightarrow H$

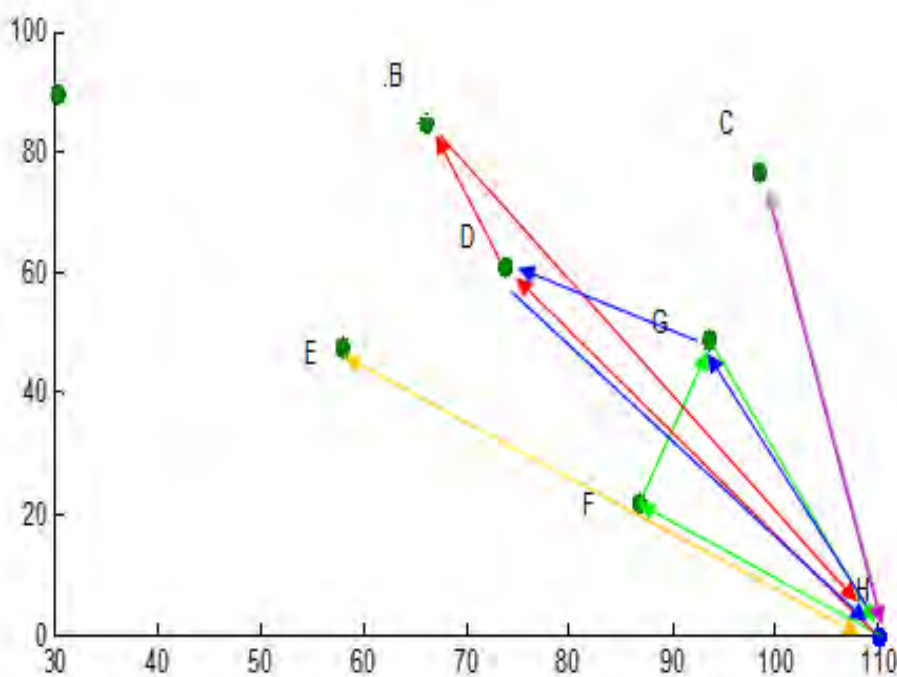


图 3 无人机飞行路线图

覆盖率 μ 可以通过如下公式计算

$$\mu = (1 - \frac{S_{\text{无法扫描区域占比}}}{S_{\text{巡查区面积占比}}}) \times 100\%$$

最后计算得 $\mu = 94.44\%$ 。

针对第二小问

5.1.3.3 地形图简化方法

为了简化计算任务量并且使简化后计算结果不至于产生过大的误差，于是，在合理范围内提出了一种简化地形图的方法。

基于此问，取无人机最大扫描直径 $2R_{\max}$ 和最小扫描直径 $2R_{\min}$ 做平均得到网格步长 $\tau = R_{\max} + R_{\min}$ ，然后对地形图以步长 τ 进行剖分，得到网格。若每个网格点中海拔高度低于 4000 的点至少占网格所有点的 50%，就可以认为这个网格点的平均海拔高度低于 4000。以此为依据，做出如下简化地形图，并且接下来的计算也是在简化地形图基础上。

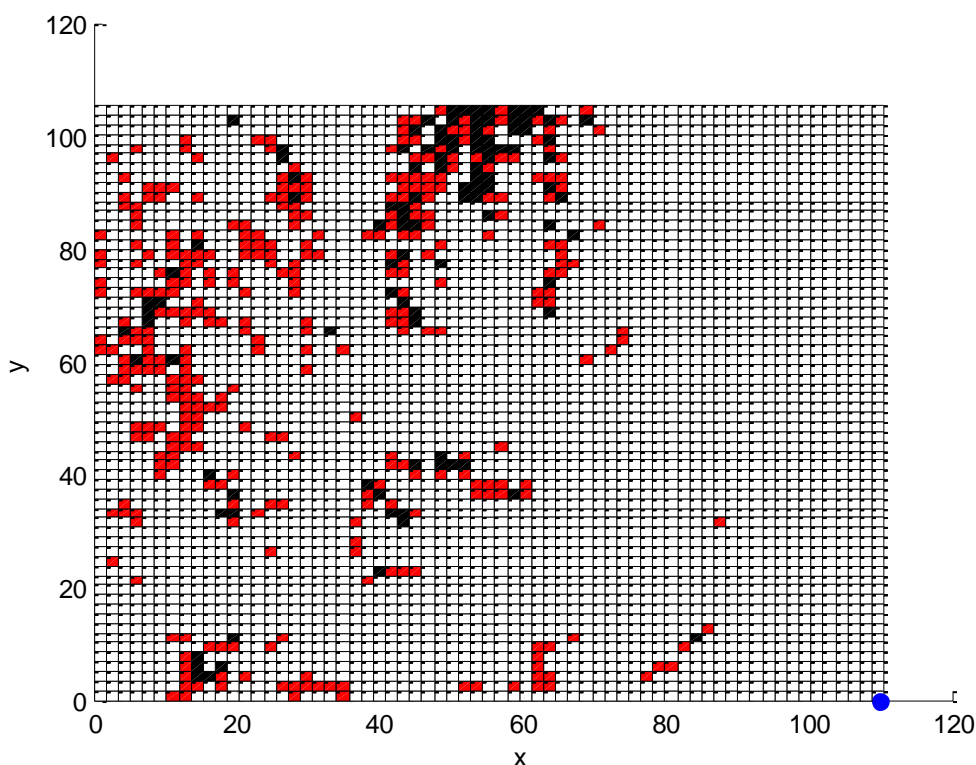


图4 海拔低于4000米点简化地形图

从图4可以得到, 中白色区域海拔高度 $h_{ij} < 4000$, 红色区域海拔高度 $4000 \leq h_{ij} < 4200$, 黑色区域为禁飞区, 海拔高度 $h_{ij} \geq 4200$ 。

5.1.3.4 禁忌搜索算法

禁忌搜索算法是组合优化算法的一种, 是局部搜索算法的扩展。禁忌搜索算法是人工智能在组合优化算法中的一个成功应用。禁忌搜索算法的特点是采用了禁忌技术。所谓禁忌就是禁止重复前面的工作。禁忌搜索算法用一个禁忌表记录下已经到达过的局部。

禁忌搜索算法流程如下:

STEP1 选定一个初始解 x^s 及给以禁忌表 $H = \Phi$;

STEP2 若满足停止条件, 停止计算; 否则, 在 x^s 的领域 $N(H, x^s)$ 中选出满足禁忌要求的候选集 $CN(x^s)$, 在 $CN(x^s)$ 中选一个评价价值最佳的解 x^{next} , $x^s = x^{next}$, 更新禁忌表 H , 重复 STEP2。

解决本问题的步骤如下:

STEP1 划分一个子图 (见图5), 要达到比较均衡, 使每架无人机飞行时间接近8小时。

STEP2 对划分的子图 G_i 采用禁忌搜索算法求得其最短巡查路线 ψ_i 和最短巡查

时间 $t(\psi_1)$

STEP3 $t(\psi_1)/480$ 接近于 1，否则重复 STEP1。

STEP4 将划分的子图从整个图中删除，对于剩下的部分，重复 STEP1

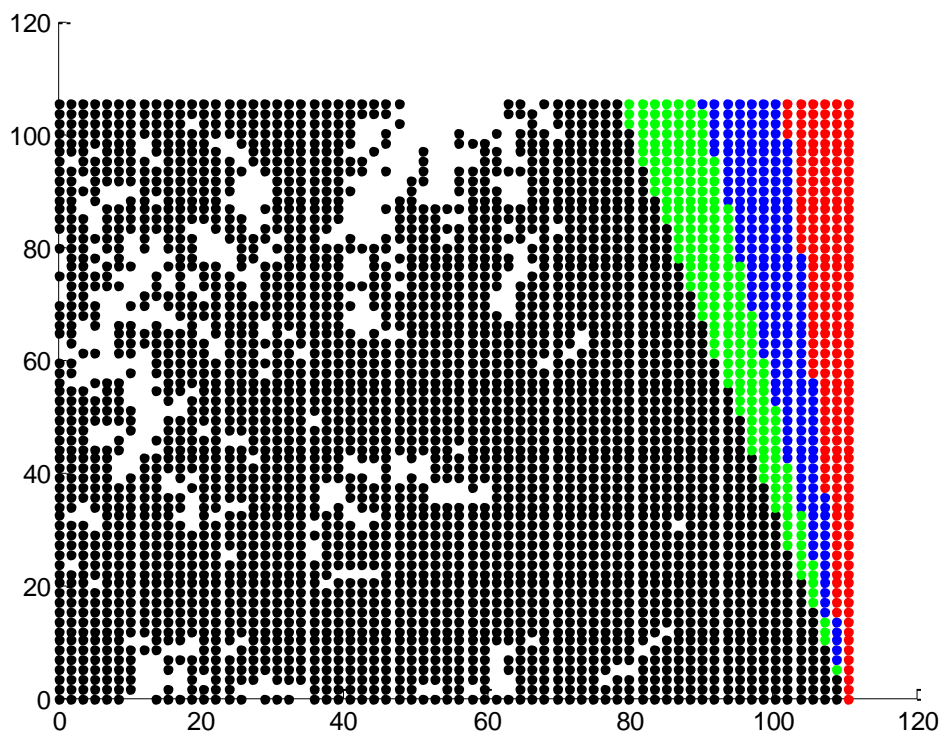


图 5 子图划分示意图

5.1.3.5 结果分析

基于上述算法流程，结合地形图地势考虑，我们将 $t(\psi_i)/480$ 的值假定为 0.8 进行计算，最终可以近似分为 23 个区域。由于每个区域每架无人机巡查的时间 $t_i \approx 480\text{min}$ ，为了保证在 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，且无人机均需从 H 出发并在 8 小时内回到 H，再出发的时间间隔不小于 1 小时，那么一架无人机的执行周期为 9 小时。我们只需每隔三小时从基地 H 分别向这 23 个区域派出 23 架无人机，即可满足条件，则共需要无人机 69 架。对于飞行时间及路线，各自区域内的无人机相同。

5.2 问题 2 的建立与求解

5.2.1 问题 2 的建立

多无人机协同任务分配问题是为了给无人机分配目标，让其执行一定的任务，使得多无人机协同任务执行的效能总和最大，所付出的代价最小。它是一种

约束规则众多的一种目标优化选择。

本题中，从基地 H,J 派出的无人机数量相同，所执行的任务是有效探测海拔 3000 米以下的生命迹象。假定 N_v 为无人机数量， M 为无人机所执行的任务，决策变量为 $x_{i,j}$, $i = \{1, 2, \dots, N_v\}$, $j = \{1, 2, \dots, M\}$ 。下面给出多无人机协同任务分配模型。

步骤一：目标函数

多无人机协同任务分配问题的一个重要思想就是使多无人机执行任务的时间尽可能短，对应到题目中就是使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可能短。不妨假设无人机同时从基地 H,J 飞出，那么最后一架完成任务的无人机回到基地的时间，即为多无人机协同任务执行的总时间。 M_i 是第 i 架无人机分配到的所有目标点， f_L 为最后一架无人机返回基地的时间， T_k 为完成第 k 个任务所需要时间。要实现题设条件，必须让 f_L 完成任务的时间最少。

$$f_L = \min_{i=1,2,\dots,N_v} \sum_{k=1}^{M_i} T_k \quad (9)$$

a. 约束规则

1) 续航时间

多无人机执行任务时，必须要在最大续航时间内执行

$$\sum_{k=1}^{M_i} T_k \leq T_{\max} \quad (10)$$

其中 T_{\max} 为最大续航时间。

2) 平面探测半径

由于每架无人所携带的仪器具有不同的探测范围，探测面积与完成任务时间有关，所以要考虑平面探测半径（单位：米）

$$r \leq \sqrt{\frac{3}{4} r_{\max}^2} \quad (11)$$

其中 r_{\max} 为最大有效探测距离。

3) 完成探测任务约束

对任意目标 $k \in M$ 进行且只进行一次探测，即

$$\text{num}(\{(i, j) | x_{i,j} = k, i = 1, 2, \dots, N_v, j = 1, 2, \dots, M_i\}) = 1 \quad (12)$$

5.2.2 问题 2 的求解

针对多无人机协同任务分配的建模,首先对复杂的地形图做简化处理,提取出有用的任务点信息(方法见问题一)。其次描述多无人机协同任务分配的建模的具体算法。

方法一:模拟退火算法

模拟退火算法得益于材料的统计力学的研究成果。统计力学表明材料中粒子的不同结构对应于粒子的不同能量水平。在高温条件下,粒子的能量较高,可以自由运动和重新排列。在低温条件下,粒子能量较低。如果从高温开始,非常缓慢地降温(这个过程被称为退火),粒子就可以在每个温度下达到热平衡。当系统完全被冷却时,终形成处于低能状态的晶体。本文基于这种最优思想,将多目标问题转化为局部单目标问题,通过控制区域内任务目标的个数,结合模拟退火算法,得到局部最优解。

求解的模拟退火算法描述如下:

(1)解空间

解空间 S 可表示为 $\{1,2,\dots,6\}$ 的所有固定起点和终点的循环排列集合,即 $S = \{(\pi_1, \pi_2, \dots, \pi_6) \mid \pi_1 = 1, (\pi_2, \pi_3, \dots, \pi_6) \text{ 为 } \{2, 3, \dots, 6\} \text{ 的循环排列}, \pi_6 = 6\}$ 其中每一个循环排列表示扫描 6 个重要基地的一个回路, $\pi_i = j$ 表示第 i 次侦察 j 点,初始解可选为 $(1, 2, \dots, 6)$, 本文使用平均法求得一个较好的初始值。

a. 目标函数

此时的目标函数为侦察所有目标的路径长度或称代价函数。建立的模型

$$\min f(\pi_1, \pi_2, \dots, \pi_6) = \sum_{i=1}^6 d_{\pi_i \pi_{i+1}} \quad (13)$$

1)新解的产生

任选序号 $u, v (u < v)$ 交换 u 与 v 的顺序,此时的新路径为

$$\pi_1 \cdots \pi_{u-1} \pi_v \pi_{v+1} \cdots \pi_{u+1} \pi_u \pi_{v+1} \cdots \pi_6$$

2)变换法

任选序号 u, v 和 w , 将 u 和 v 之间的路径插到 w 之后,对应的新路径为(设

$$u < v < w) \quad \pi_1 \cdots \pi_{u-1} \pi_{v+1} \cdots \pi_w \pi_u \cdots \pi_v \pi_{w+1} \cdots \pi_6$$

b. 代价函数差

对于 2 变化法,路径差可表示为

$$\Delta f = (d_{n_{u-1}n_v} + d_{n_u n_{v+1}}) - (d_{n_{u-1}n_v} + d_{n_v n_{v+1}}) \quad (14)$$

1)接受准则

$$P = \begin{cases} 1 & \Delta f < 0 \\ \exp(-\Delta f / T) & \Delta f \geq 0 \end{cases} \quad (15)$$

如果 $\Delta f < 0$ ，则接受新的路径。否则，以概率 $\exp(-\Delta f / T)$ 接受新的路径，即若 $\exp(-\Delta f / T)$ 大于 0 到 1 之间的随机数则接受。

c. 降温

利用选定的降温系数 α 进行降温即： $T \leftarrow \alpha T$ ，得到新的温度，这里我们取

$$\alpha = 0.999$$

d. 结束条件

用选定的终止温度 $e = 10^{-30}$ ，判断退火过程是否结束。若 $T < e$ ，算法结束，输出当前状态。

方法二：基于模拟退火算法的多目标局部优化算法

结合问题，下面介绍一种基于模拟退火算法的多目标局部优化算法。已知基地坐标为 H (110, 0)，J (110, 55)，以简化后的地形图为依据，不妨以基地所在位置为界，将区域划分为两部分，记划分上域为区域 I，划分下域为区域 II，见图 6：

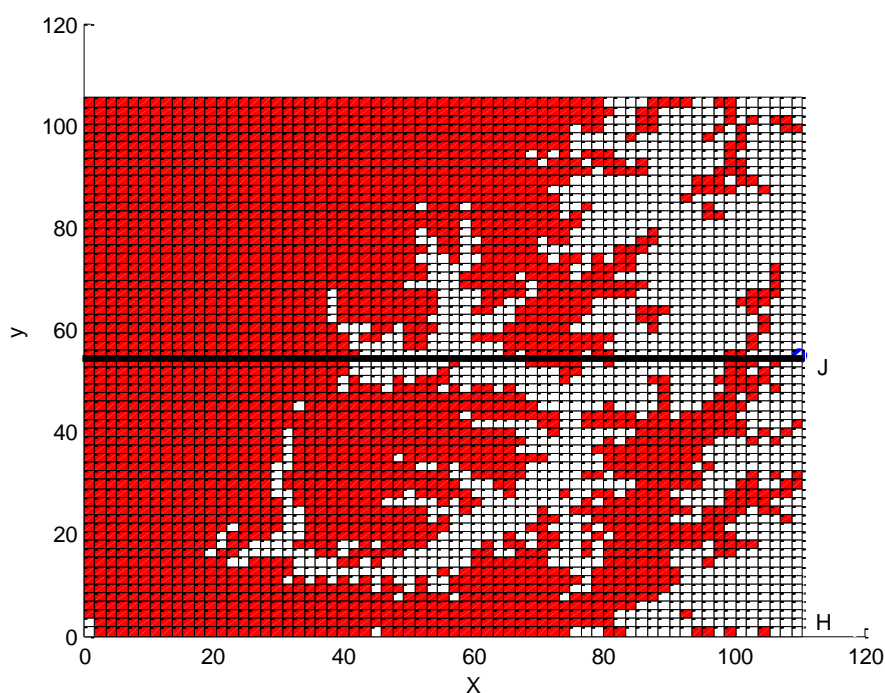


图 6 简化地形图按照基地分区

其中白色区域为海拔低于 3000 米的区域，即需要探测的区域。红色区域为海拔

高于 3000 米的区域。由于无人机最大飞行高度可以达到 5000 米，而图 1 区域最高海拔为 4800 米，故此问暂不考虑无人机被山体阻隔无法通过的问题。

以基地 H, J 为起始点，向区域 I、II 分别派出 15 架无人机执行任务。为了使最后一架无人机返回基地的时间较短，对于这个问题，基本思想是，先根据探测区域的分布特点将 3000 米以下的点分为 15 个子区域，分别让 15 架无人机去探测这 15 个子区域，求取各自区域的最短时间，然后根据 15 个子区域的最长时间，对任务不均衡子区域之中的点做适当的调整。

算法步骤如下：

(1) 划分子区域 $G_i (i=1,2,\dots,15)$ ，要达到比较均衡，应该让每架无人机完成任务的时间基本相同，由于探测区域分布不均匀，先按角度划分，将区域划分为夹角相等的 15 个子区域。

(2) 对每个子区域 G_i 分别运用模拟退火算法求得其最短探测路径 ψ_i 和最短探测时间 $t(\psi_i) (i=1,2,\dots,15)$ 。

(3) 均衡任务。定义均衡率

$$\eta = \frac{\min\{t(\psi_1), t(\psi_2), \dots, t(\psi_{15})\}}{\max\{t(\psi_1), t(\psi_2), \dots, t(\psi_{15})\}} \times 100\% \quad (17)$$

若 η 接近与 1，则上面的划分可以接受，否则的话，根据 $t(\psi_i) (i=1,2,\dots,15)$ 的大小调整每个区域的角度，从而调整各区域内的任务点的数量，直至任务达到均衡。

最后计算结果如下

表 4 区域 I 各子区域探测时间（单位：分钟）

子区域	1	2	3	4	5	6	7
探测时间	178.79	178.79	184.45	169.87	180.12	179.80	175.07
8	9	10	11	12	13	14	15
163.76	156.19	140.18	142.15	143.44	139.12	139.24	139.52

$$\text{区域 I 均衡率 } \eta = \frac{\min\{t(\psi_1), t(\psi_2), \dots, t(\psi_{15})\}}{\max\{t(\psi_1), t(\psi_2), \dots, t(\psi_{15})\}} = \frac{139.12}{184.45} \times 100\% = 75.42\%。$$

表 5 区域 II 各子区域探测时间（单位：分钟）

子区域	1	2	3	4	5	6	7
探测时间	187.46	210.85	208.91	215.60	227.29	224.94	222.24
8	9	10	11	12	13	14	15
189.02	181.13	184.84	192.76	175.04	177.21	178.56	182.29

$$\text{区域 II 均衡率 } \eta = \frac{\min\{t(\psi_1), t(\psi_2), \dots, t(\psi_{15})\}}{\max\{t(\psi_1), t(\psi_2), \dots, t(\psi_{15})\}} = \frac{175.04}{227.29} \times 100\% = 77.01\%$$

因此，要保证题中给出的全区域内海拔 3000 米以下部分能被探测到的面积尽可能大，还要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，那么最后一架完成任务的无人机回到基地的时间应为 227.29 分钟。

5.2.3 结果分析

观察区域 I、区域 II 均衡率发现，与 1 还是有一定的误差。分析发现，首先对地形图简化时，采用无人机平均探测直径 r_A 为步长对原始地形图进行网格剖分限制了计算的精度。其次，由于地形图中各点海拔高度均低于 5000m，无人机飞行并不受阻挡，但是没有考虑飞行过程中最大爬升(俯冲)角度为 $\pm 15^\circ$ 这个条件。最后，探测面积采用了平均探测直径，也是误差导致的原因。

5.3 问题 3 的建立与求解

5.3.1 问题 3 的建立

无人机的任务是作为通信中继保证地面人员之间信息及时准确的传输。在无人机执行中继任务之间，必须找到地面终端，才能建立起通信网络进行通信中继。因此本文根据任务需要将无人机分成两种状态：搜索状态和中继状态。对于搜索地面终端区域，本文将无人机的搜索方式分为两种情况

a. 假设地面终端用户可以自由移动

地震后，地面终端只能以自身为圆心，以 2000 米为半径的安全区域内移动，无人机之间的最大通信距离不能为 6000 米，与地面终端的距离为 3000 米，从而才能保证无人机之间互相通信，地面与地面之间的通信由无人机转接，并且能保证无人机与地面终端的正常通信。因此无人机在对地面终端传送信息时，应该全部覆盖地面终端的活动区域。因此，在地面终端可以移动的情况下，无人机应该扫描到该区域的所有范围，扫描方式如图 7 所示，

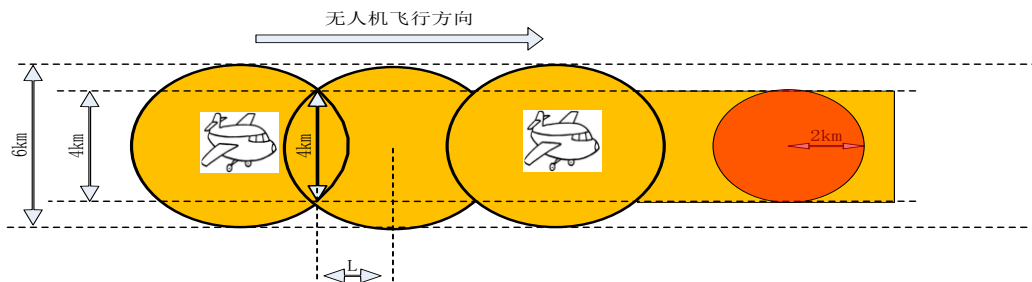


图 7 无人机扫描区域相扣式扫描

从图 7 可以看出，宽度为 4000 米的黄色袋装区域为无人机无楼扫描区域，这个区域可以覆盖地面终端自由移动范围，即半径为 2000 米的圆。此时无人机扫描为半径 3000 米的原型区域，无人机之间项链区域最大宽度为 4000 米，则宽度为 $L = \sqrt{R^2 - r^2}$

b. 假设地面终端用户按照指挥部给定的通信点进行移动

当地面终端用户按照指挥部给定的通信点进行移动时，首先，将第一架无人机按照假设 1 中的路线进行宽度 400 米的扫描方式进行信息传输，给每一个地面终端用户传输一个信号位置，使其地面终端用户移动到信号点处，并保持不变。随后派出无人机对移动后的地面终端用户位置进行新路线的扫描，扫描方式为相切式扫描如图 8 所示，

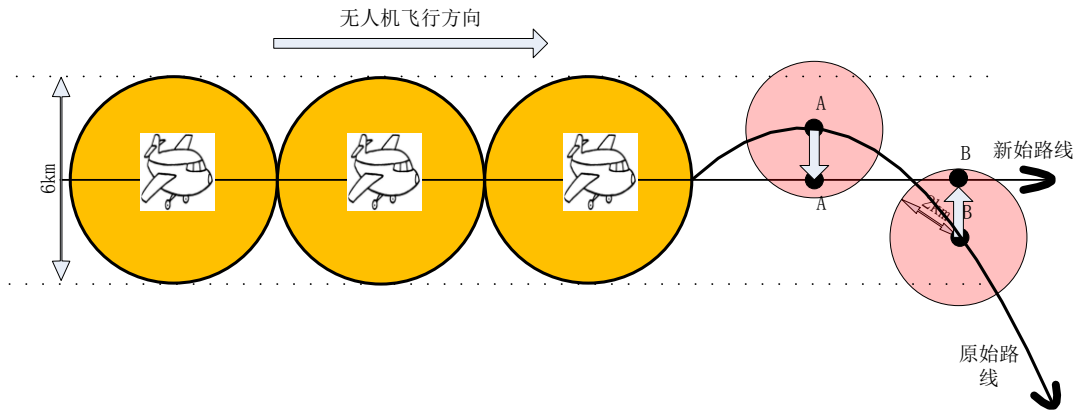


图 8 无人机扫描区域相切式扫描

从图 8 可以看出，A，B 两点的地面终端按指挥部给出通信点进行移动。优化后新路线的距离比原始路线距离缩短，从而可以派出少量无人机可以对 72 个地面终端进行中继。

5.3.2 问题 3 的求解

无人机将作为地面终端的通信中继，为灾区提供持续的通信保障。本文使用狼群算法求解出扫描地面终端的最少的无人机数量，以及每架无人机最佳的飞行路线。

灰狼算法描述

STEP1: 灰狼与猎物之间的距离。在捕食过程中，灰狼首先需要对猎物之间的距离：

$$D = |CX_p(t) - X(t)|$$

式中， $X_p(t)$ 表示第 t 代时猎物的位置； $X(t)$ 表示第 t 代时灰狼个体的位置；常

数 C 为摆动因子，由式 $C=2r_1$ 所决定， r_1 表示 $[0,1]$ 之间的随机数

STEP2: 灰狼位置更新: $X_{(t+1)}=X_p(t)-AD$

$$A=2ar_2-a$$

式中， A 为收敛因子； r_2 为 $[0,1]$ 的随机数； a 随着迭代次数的增加从 2 递减到 0。

STEP3: 猎物位置定位

灰狼判断出猎物所在位置时，将由头狼 α 带领 β ， δ 发动追捕行为。在狼群中 α, β, δ 三者最靠近猎物，可以利用这三者的位置来判断猎物所在方位。

STEP4: 狼群中个体跟踪猎物方位的数学描述如下:

$$D_\alpha = |C_1 X_\alpha(t) - X(t)| \quad (18)$$

$$D_\beta = |C_2 X_\beta(t) - X(t)| \quad (19)$$

$$D_\delta = |C_3 X_\delta(t) - X(t)| \quad (20)$$

$$X_1 = X_\alpha - A_1 D_\alpha \quad (21)$$

$$X_2 = X_\beta - A_2 D_\beta \quad (22)$$

$$X_3 = X_\delta - A_3 D_\delta \quad (23)$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \quad (24)$$

由式 (18-24) 计算出群内个体与 α, β, δ 的距离，然后由式 (24) 即可综合判断出个体向猎物移动的方向。

第一种情况：地面终端只能以自身为圆心，以 2000 米为半径的安全区域内移动，应用蚁群算法求解得到无人机遍历 72 个地面终端的最佳路径如图 9 所示，得到的最短路径为 514.1388 千米，无人机数量为 116 台。

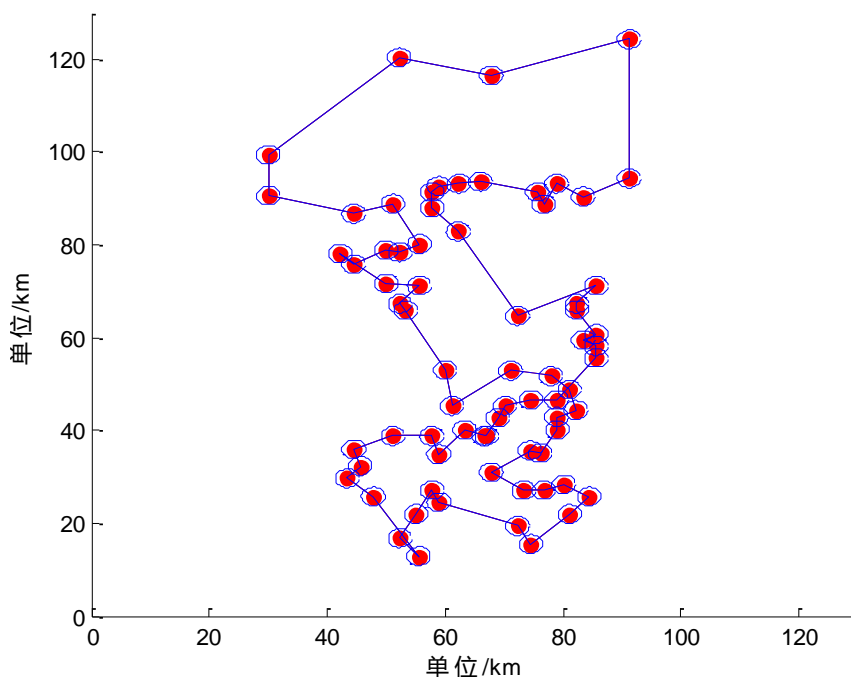


图 9 蚁群算法对 72 个地面终端的最佳路径

第二种情况：地面终端用户按照指挥部给定的通信点进行移动，第一架无人机确定 72 个地面终端所要移动的位置之后，随后的无人机进行遍历，所得到无人机飞行路线如图 10 所示，得到的最短路径为 446.1388 千米，无人机之间的距离为无人机数量为 75 台。

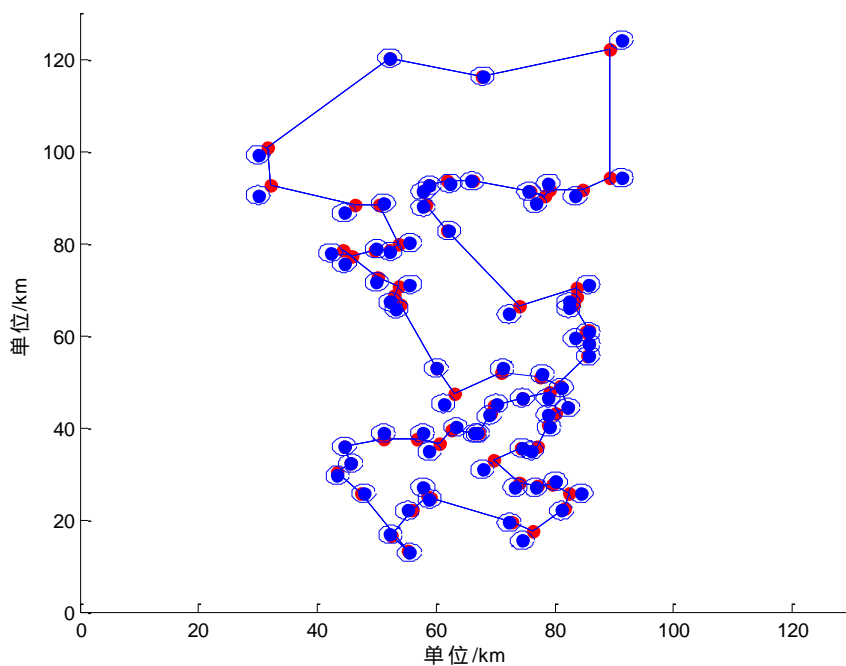


图 10 地面终端用户按照指挥部给定的通信点进行移动无人机所飞行路线

5.3.3 结果分析

本文对无人机协同中的搜索路径规划进行研究，在无人机能够完成通信中继任务时得到无人机的最少架数。首先通过假设地面终端只在一定的区域内移动，使用蚁群算法得到所需无人机数量为 116 台，为了得到更好了解决方案，使得无人机的数量减少，其次假设无人机按照指挥部的移动方式移动到信号点，再将剩余的地面终端进行通讯中继，从而得到需要 75 台无人机就可以遍历所有地面终端。

5.4 问题 4 的建立与求解

5.4.1 问题 4 的建立

对于这个问题，先根据地面终端用户的分布特点将地面终端用户大体分为三个区域，并使三架无人机分别对这三个区域的用户进行探测，求取各自的最短时间，然后对任务不均衡区域进行适当的调整。具体的步骤如下：

STEP1: 划分子图

要达到区域划分的均衡，应使得每架无人机的巡航时间基本相同，可以将地方终端用户分为面积相等的三部分。过初始点 $H(110,0)$ 以斜率 $k_1 = \frac{\sqrt{3}}{3}$ 做一条斜线，过 $H(110,0)$ 点以斜率 $k_1 = \sqrt{3}$ 做一条斜线，将地面终端所在区域划分成 G_1, G_2, G_3 三部分，初始的划分结果如图 11 所示：

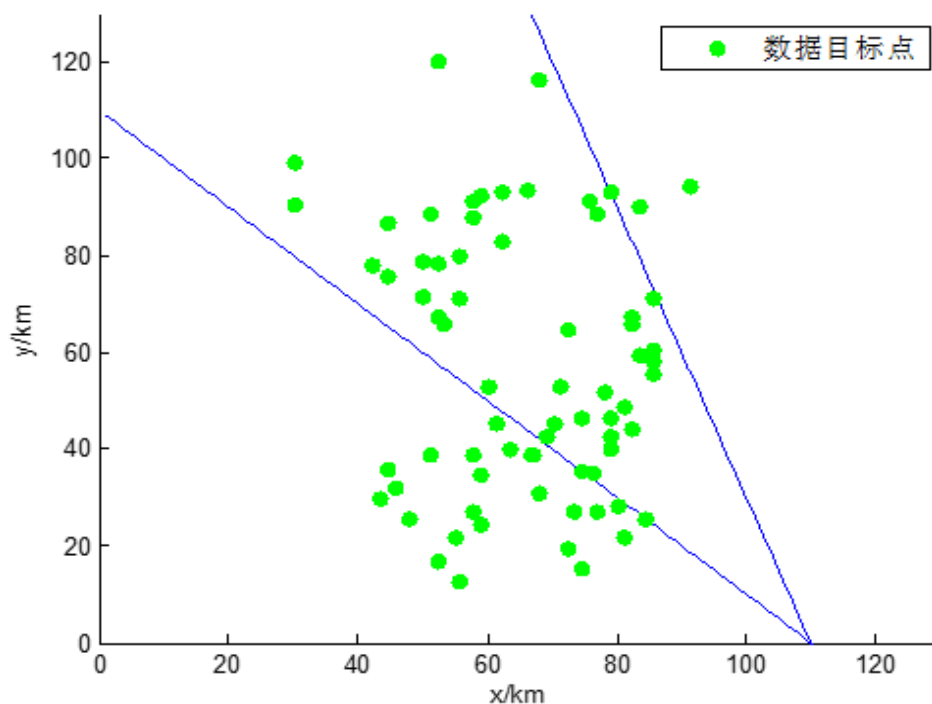


图 11 地面终端的初始划分子图

从图 11 可以看出，三个区域中的点的分布不均衡，使得无人机探测三个区域中的路径不能达到最短，因此使用蚁群算法将划分的三个区域进行调整，使得用户数量分布误差达到最小。

运用蚁群算法对地面终端用户的最终的划分方法如图 3 所示。

划分所依据的约束规划为：

$$Obj: \min \frac{1}{3} \sum_{i=1}^3 \sum_{j=1}^3 (k_i - k_j)^2 \quad (24)$$

$$s.t. \begin{cases} R_1 < a_{im}(k_1, 1) < R_1 \\ R_2 < a_{im}(k_2, 2) < R_2 \\ R_3 < a_{im}(k_3, 3) < R_3 \\ R_i = \left[\tan 30^\circ + \sum_{i=1}^m (i/2) \right] \cdot (x-110) + \sum_{j=1}^n (j/100) \end{cases}$$

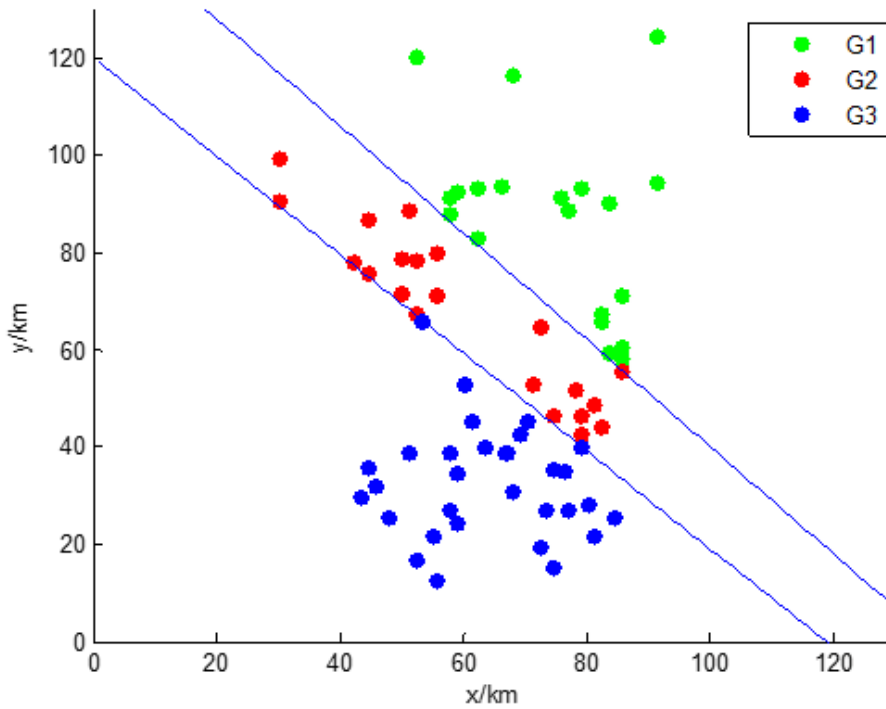


图 12 蚁群算法对地面终端的划分子图

从图 12 可以看出， G_1 区域中的点分布比较集中， G_2 和 G_3 区域中的点分布比较分散，因此 G_1 区域比 G_2 、 G_3 区域所划分的地面终端用户较多，选择合适的初始点，使 3 架无人机分别从出发点出发遍历区域中所有的点，使得无人机传播信息经过的路径最小。

STEP2: 选择最短路径

采用蚁群算法对三个区域中所有路径进行分析，选择出每个区域中用户的最短路径，其中 G_1 区域中的最短路径的距离为 219.0503 千米， G_2 区域中的最短路径的距离为 197.4493 千米， G_3 区域中最短路径的距离为 228.3028 千米。由于蚁群算法具有随机性，从图 12 可以看出，小方块中的点虽然属于不同的区域，但是距离小于同一区域中相邻点之间的 $1/5$ ，从而对蚁群算法进行优化处理，将图中不同区域中距离小于同一区域中 $1/5$ 的点进行优化，得到图 14，优化后， G_1 区域中的最短路径的距离为 198.5020 千米，距离减少了 20.5483 千米。 G_2 区域中的最短路径的距离为 191.9085 千米，距离减少了 5.5408 千米。 G_3 区域中最短路径的距离为 228.5233 千米，距离增加了 5.5408 千米。

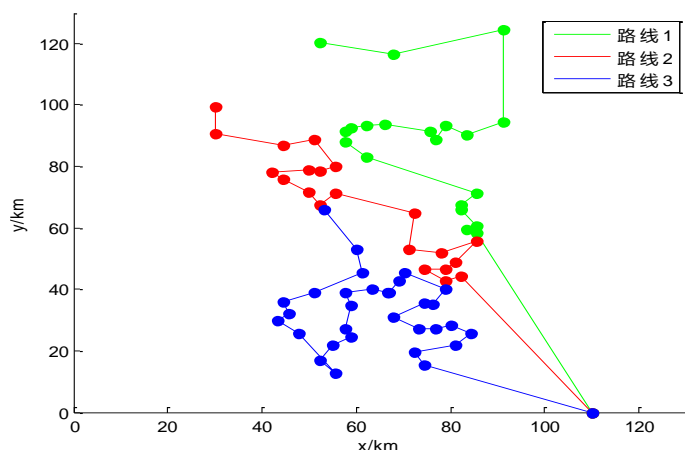


图 13 蚁群算法初始划分区域

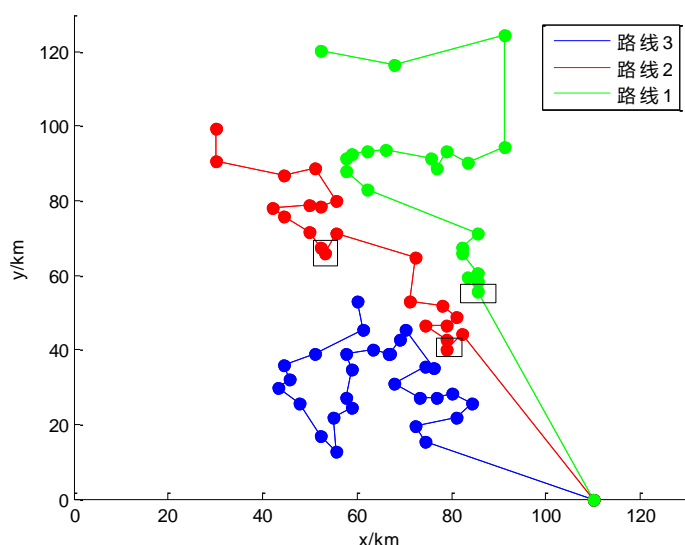


图 14 进行优化后蚁群算法的区域划分

$$\text{Obj: } \min[\max(t_a, t_b, t_c)] \quad (24)$$

$$\begin{aligned}
 L_a &= D_{Ha} + \sum_{i=1}^n D_{i,i+1,a} \quad n = 29 \\
 L_b &= D_{Hb} + \sum_{i=1}^m D_{i,i+1,b} \quad m = 22 \\
 L_c &= D_{Hc} + \sum_{i=1}^k D_{i,i+1,c} \quad k = 21 \\
 L &= \frac{500 \cdot 1024 \cdot 1024}{r_{1500,5}} \cdot v_{\min} \\
 S.T. \quad & \begin{cases} D_{i,j,a} = \sqrt{(aim_{i,1,a} - aim_{j,1,a})^2 + (aim_{i,2,a} - aim_{j,2,a})^2} \quad i, j = 1, 2, \dots, 29 \\ D_{i,j,b} = \sqrt{(aim_{i,1,b} - aim_{j,1,b})^2 + (aim_{i,2,b} - aim_{j,2,b})^2} \quad i, j = 1, 2, \dots, 22 \\ D_{i,j,c} = \sqrt{(aim_{i,1,c} - aim_{j,1,c})^2 + (aim_{i,2,c} - aim_{j,2,c})^2} \quad i, j = 1, 2, \dots, 21 \\ t_a = \frac{L_a - \cos 30^\circ \cdot 1.5}{v_{\max}} + \frac{\cos 30^\circ \cdot 1.5 + a_num1 \cdot (L - \cos 30^\circ \cdot 1.5)}{v_{\min}} \\ t_b = \frac{L_b - \cos 30^\circ \cdot 1.5}{v_{\max}} + \frac{\cos 30^\circ \cdot 1.5 + b_num1 \cdot (L - \cos 30^\circ \cdot 1.5)}{v_{\min}} \\ t_c = \frac{L_c - \cos 30^\circ \cdot 1.5}{v_{\max}} + \frac{\cos 30^\circ \cdot 1.5 + c_num1 \cdot (L - \cos 30^\circ \cdot 1.5)}{v_{\min}} \\ a_num1 = 14 \quad b_num1 = 9 \quad c_num1 = 12 \end{cases}
 \end{aligned}$$

式中, a_num1 , b_num1 , c_num1 表示 a/b/c 三个分组中子信道带宽小于 150 的点的个数。求得 $L_a = 3.7445h$, $L_b = 3.1720h$, $L_c = 3.6671h$, L 表示子信道带宽为 100 的目标点需要传输数据时移动的距离。

STEP3: 求解飞机最优扫描高度

3 架无人机携带通信设备向地面终端发送内容时,地面终端 i 看无人机的仰角大于 30° , 最大距离不超过 3000 米且没有山体阻隔, 为了使得无人机从目标点的上方穿过时传输的信息总量最大, 因此传输所需的时间需最长, 因为在无人机传输速率相同的情况下, 传输的时间越长, 传输信息的总量就越多。因此控制无人机的最低速度保持 60 千米/小时, 无人机到地面终端的距离不同, 传输数据的总量也就不同。根据题意可知, 飞机离障碍物的最低距离为 50 米, 欧氏距离保持在 50-3000 米之间, 因此寻找传输信息最优高度。

当无人机与地面终端的欧氏距离低于 50 米时, 仰角将小于 30° , 无人机无法向地面终端传输信息; 当无人机与地面终端的欧氏距离为最低 50 米时, 仰角最

小为 30° ，无人机与地面的垂直高度为 1000 米，与地面终端的水平距离为 $1000 \tan 30^\circ$ ，传输的信息总量为 $2000 \tan 30^\circ$ ；当无人机与地面终端的欧式距离最高为 3000 米的时，仰角最小 30° ，可知，无人机与地面终端的水平距离为 $1500 \tan 30^\circ$ ，传输的信息总量为 $3000 \tan 30^\circ$ ；当无人机与地面终端的欧式距离高于 3000 米时，仰角最小 30° ，无人机无法向地面终端传输信息，只有当无人机飞到与地面终端的欧氏距离等于 3000 米的时，才开始传输信息，因此传输的信息总量较少，当无人机在地面终端的垂直上方，并且距离为 3000 时，等于一瞬间传输信息量。解决无人机向地面终端传输信息的问题可分为两种情况进行计算，具体计算如下所示，

(1) 无人机传输信息时垂直高度为 1500 米以下：

首先计算无人机进入地面终端的半程扫描时间为：

$$t = \frac{h \cdot \tan 60^\circ}{v}$$

式中， v 表示无人机扫描时的水平速度， h 表示无人机扫描时的垂直高度。

无人机与地面终端的欧式距离为：

$$d(t_i) = \sqrt{h^2 + (h \cdot \tan 60^\circ - vt_i)^2}$$

(2) 无人机传输信息时垂直高度为 1500 米以上：

计算无人机进入地面终端的半程扫描时间为：

$$t = \frac{\sqrt{(3000^2 - h^2)}}{v}$$

无人机与地面终端的欧式距离为：

$$d(t_i) = \sqrt{h^2 + ((3000^2 - h^2) - vt_i)^2}$$

根据题已知无人机飞过地面终端 i 用户的信息速率为

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

式中， B_i 表示无人机服务 i 的子信道带宽， $p_i(t)$ 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，取 $5W$ ， ρ_0 为信道特性参数，取 4.314×10^{-10} ，从而根据最大的海拔高度可以求解到所用的最长时间 t_{\max} 。信息速率 $r_i(t)$ 对 t 进行积分

$$\int_0^{t_{\max}} r_i(t) dt = \int_0^{t_{\max}} B_i \log_2 \left(1 + \frac{P_i(t)}{\rho_0 d^2(u, i)} \right) dt = \int_0^{t_{\max}} B_i \log_2 \left(1 + \frac{P_i(t)}{\rho_0 \left[(h \tan 30^\circ)^2 - (vt)^2 \right]} \right) dt$$

即为无人机向地面终端的最大传输量。

5.4.2 问题 4 的求解

表 6 无人机分组参数最终结果

序号	无人机	X 坐标	Y 坐标	速度	功率 w
26	第三架	85.600	55.600	64.292	5
22	第三架	85.600	58.200	58.499	5
25	第三架	83.400	59.500	56.965	5
24	第三架	85.600	60.800	57.510	5
19	第三架	82.300	66	61.160	5
21	第三架	82.300	67.300	59.732	5
20	第三架	85.600	71.200	58.516	5
40	第三架	62.200	82.900	63.308	5
6	第三架	57.800	88	60.852	5
4	第三架	57.800	91.400	60.497	5
11	第三架	58.900	92.700	64.171	5
12	第三架	62.300	93.200	57.858	5
39	第三架	66	93.600	62.572	5
5	第三架	75.600	91.400	62.537	5
3	第三架	76.700	88.800	58.804	5
49	第三架	78.900	93.200	60.678	5
2	第三架	83.400	90.400	55.758	5
1	第三架	91.200	94.300	55.539	5
63	第三架	91.200	124.300	60.307	5
50	第三架	67.800	116.500	62.791	5
52	第三架	52.200	120.400	64.340	5
36	第二架	85.600	55.600	61.550	5
23	第二架	85.600	58.200	56.626	5
70	第二架	83.400	59.500	56.189	5
64	第二架	85.600	60.800	59.983	5
46	第二架	82.300	66	64.597	5
68	第二架	82.300	67.300	58.403	5
27	第二架	85.600	71.200	60.852	5
41	第二架	62.2000	82.900	57.238	5
18	第二架	57.800	88	62.512	5
15	第二架	57.800	91.400	57.550	5
66	第二架	58.900	92.700	60.059	5

"华为杯"第十四届中国研究生数学建模竞赛



17	第二架	62.300	93.200	61.990	5
30	第二架	66	93.600	63.909	5
34	第二架	75.600	91.400	64.5929	5
10	第二架	76.700	88.800	60.472	5
13	第二架	78.900	93.200	56.386	5
33	第二架	83.400	90.400	56.492	5
53	第二架	91.200	94.300	57.575	5
7	第二架	91.200	124.300	63.407	5
14	第二架	67.800	116.500	57.542	5
8	第二架	52.200	120.400	63.142	5
9	第二架	85.600	55.600	57.435	5
43	第一架	74.500	15.500	64.057	5
48	第一架	72.300	19.400	56.269	5
59	第一架	81.100	22	64.133	5
47	第一架	84.400	25.800	61.323	5
37	第一架	80.100	28.400	55.975	5
60	第一架	76.7000	27.100	57.784	5
72	第一架	73.300	27.100	60.468	5
42	第一架	67.800	31	64.575	5
29	第一架	74.400	35.700	64.648	5
54	第一架	76.100	35.100	56.576	5
45	第一架	70.100	45.300	64.705	5
69	第一架	68.900	42.700	64.571	5
16	第一架	67	38.800	59.853	5
65	第一架	66.500	38.800	63.002	5
35	第一架	63.300	40.100	56.418	5
67	第一架	57.700	38.800	59.217	5
58	第一架	58.900	34.900	64.157	5
44	第一架	57.800	27.100	62.922	5
57	第一架	58.900	24.600	64.594	5
38	第一架	55.100	22	61.557	5
62	第一架	55.500	12.900	55.357	5
32	第一架	52.300	16.800	63.4912	5
51	第一架	47.800	25.800	64.339	5
56	第一架	43.400	29.700	61.787	5
31	第一架	45.600	32.300	62.577	5
28	第一架	44.500	36	62.431	5
71	第一架	51.100	38.800	58.922	5

5.4.3 结果分析

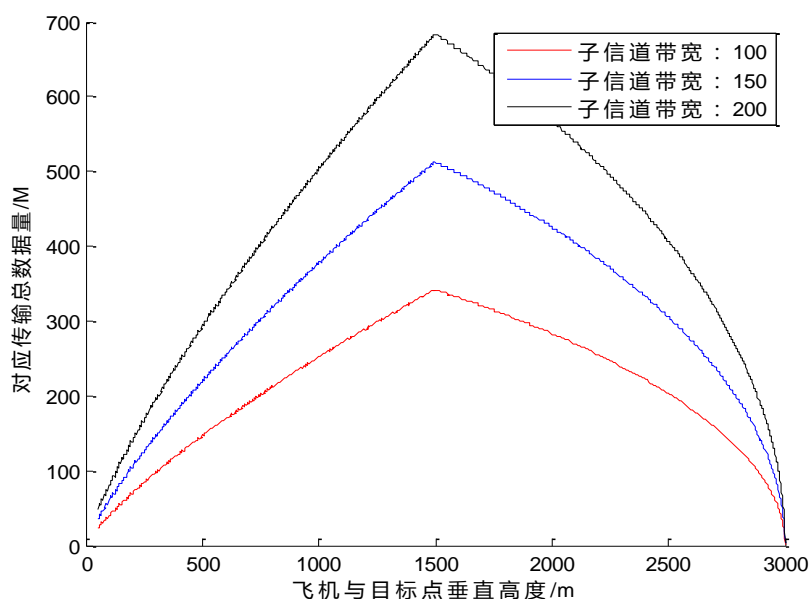


图 16 无人机与目标点的垂直高度与对应传输总数据量的关系

从图 16 可以看出，无论无人机的子信道宽度为多少，当无人机与地面终端的垂直距离在 1500 米以下时，无人机对地面终端的传输信息量随着无人机与目标点的垂直高度的升高而增加；当无人机与地面终端的距离在 1500 米以上时，无人机对地面终端的传输信息随着无人机与目标点的垂直高度的升高而降低，当在 1500 米处达到了最大值，所对应的无人机传输总数据量最大。在 1500 米处时，无人机服务 i 的子信道宽度与对应传输总数量之间的关系图如下所示，

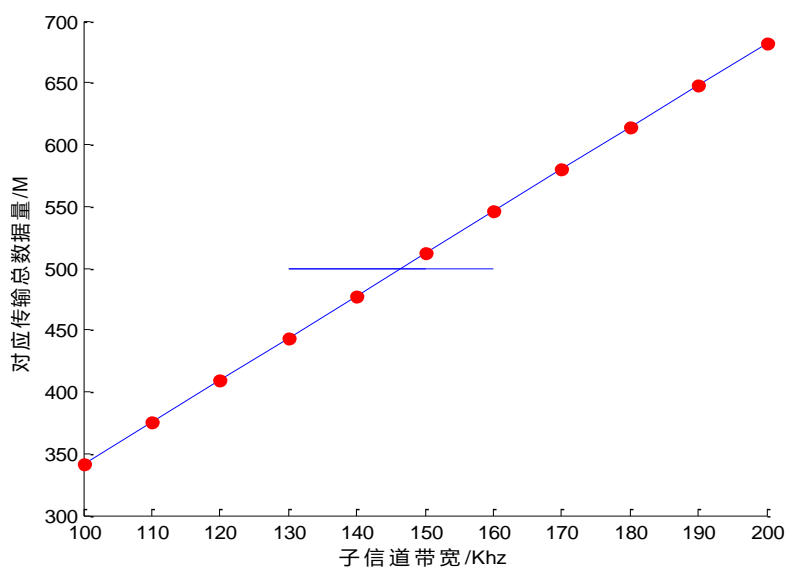


图 17 子信道宽度与对应传输总数据量关系

从图 17 可以看出,无人机飞行的垂直高度为 1500 米,当无人机传输总数据量为 500M 时,服务 i 子信道宽度大于等于 140Khz,无人机才能将信息全部传输成功。当无人机服务 i 的子信道宽度小于 140Khz 时,传输的总数据量也小于 500M,无人机从地面终端的圆心穿过无法将信息全部传输。因此对无人机服务 i 的子信道宽度最小为 100Khz 时,对应传输总数据量为 341.1772M,因此只要让最小的子信道宽度对地面终端的传输信息完成后,才能使其余的无人机完成对地面终端的信息输入。72 个地面终端用户的子信道宽度的分布图如图 18 所示:

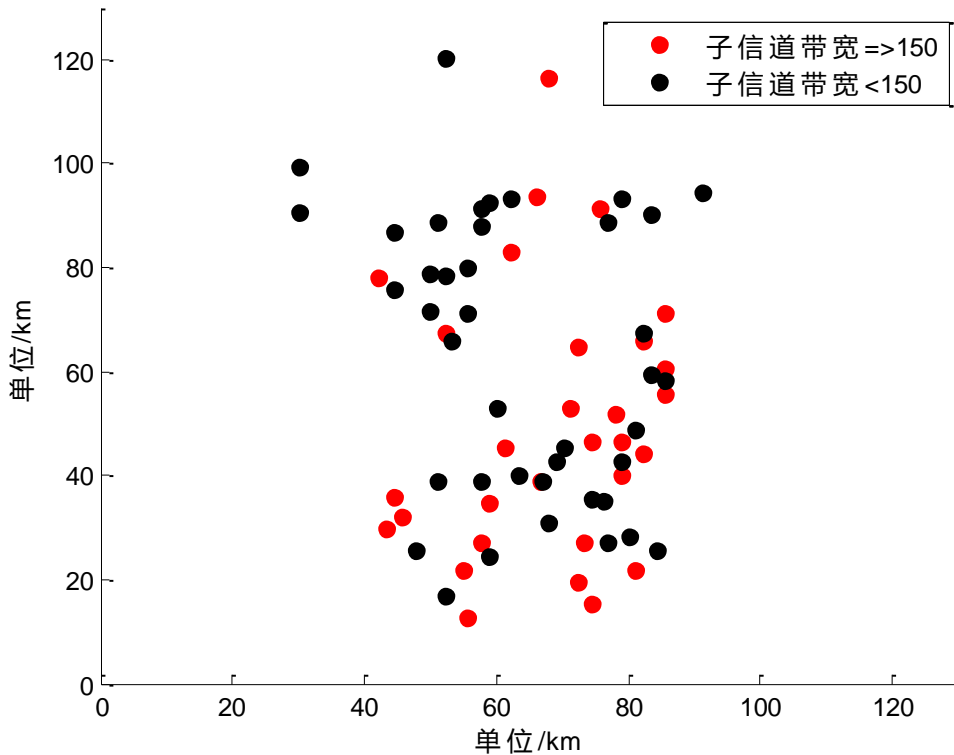


图 18 72 个地面终端用户子信道宽度的分布

从图 8 可以看出,72 个地面终端用户中子信道宽度小于等于 140 的用户有 31 个,也就是说,有 31 个用户从用户的圆心穿过不能将信息传送完成。因此以子信道宽度为 100Khz 为例,无人机与地面终端的欧氏距离达到最大 3000 米,每台通信装备的总功率 5 瓦,可知, i 用户在时刻 t 接收到无人机发送的信息速率为 $r_i(t)$ 为定值 $1.032 \times 10^6 \text{ bit/s}$,传输的总数据量最小为 500M 时,可得: $500 = \int_0^1 r \cdot t dt$,求得 $t = 508s$ 从而得到无人机对地面用户传输信息时,在用户区域内所经过的长度 $L = v \cdot t = 8.4682$ 千米。因此当服务于 i 的子信道宽度小于 140Khz 时,无人机先进入用户所在区域,在区域内以某一个圆心为半径的圆中进行剩余信息量的传输,将剩余信息全部完成后离开用户所在区域。

若两地面终端之间的距离大于一个范围 L ，则在离开一个地面终端后进行加速飞行，当加速度达到速度限定最大值（100km/h）时，匀速飞行，在接近下一个地面终端时进行减速飞行，使得进入下一个传输区域时无人机的时速保持在

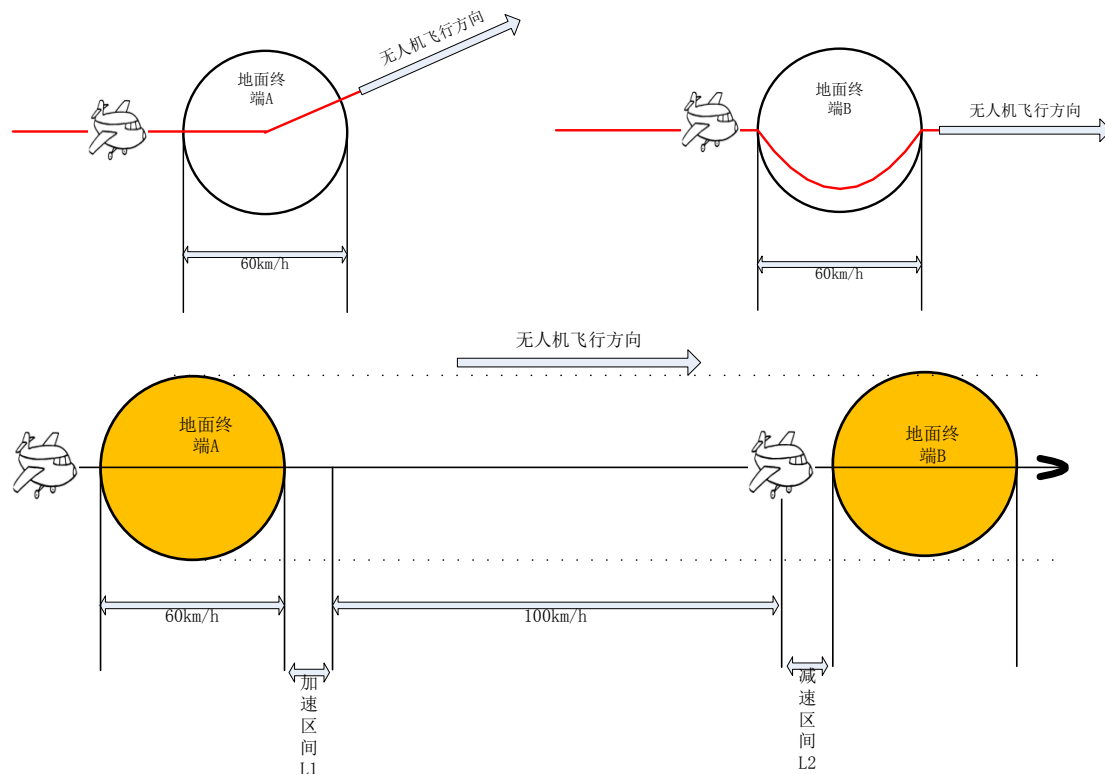


图 19: 无人机在地面终端之间的加速与减速

加速区间 L_1 , L_2 为:

$$L_1 = v_{\min} \cdot t + \frac{a_1 \cdot t^2}{2}, \quad L_2 = v_{\max} \cdot t - \frac{a_1 \cdot t^2}{2}$$

求解得 60km/h 加速到 100km/h 所需要的最短时间为 2.22 秒，最短加速距离为 49.377m。若无人机离开上一个地面终端时的速度大于 60km/h,则加速距离相应减小。

6 模型的评价与改进

6.1 模型的优点

(1) 蚁群算法：算法具有较强的鲁棒性，参数数目少，设置简单，易于应用优化问题求解。

(2) 本文采用多种启发式算法对问题的不同角度进行求解，

6.2 模型的缺点

- (1) 蚁群算法：陷入局部最优，运行时间长。
- (2) 贪心算法：注重局部最优从而达到整体最优。

6.3 模型的推广

本文为解决无人机线路规划问题，基于蚁群算法，贪心算法，狼群算法建立了寻找最优路线的优化模型以及研究思路。

本文在问题的模型建立上充分利用了蚁群算法易理解、参数少、操作简单等优点，蚁群算法后期快速收敛，易于与其他方法结合，在解决目标优化类型的问题上有较好的前景

7 参考文献

- [1] 符小卫,程思敏,高晓光,无人机协同中继过程中的路径规划与通信优化[J], 36 (5): 891-893, 2010.
- [2] 陈海,何开封,钱炜祺,多无人机协同覆盖路径规划[J], 3 (37): 929-932, 2016.
- [3] Seyedali Mirjalili,Seyed Mohammad Mirjalili,Andrew Lewis,Grey Wolf Optimizer[J],ScienceDirect,69 (3): 46-61,2014.
- [4] 李士勇,陈永强,李研.蚁群算法及其应用[M].哈尔滨工业大学出版社, 28(1): 42-45,2004.
- [5] 李大卫,王莉,王梦光.遗传算法与禁忌搜索算法的混合策略[J].系统工程学报,28(3):28-34, 1998.
- [6] 田菁,多无人机协同侦察任务规划问题建模与优化技术研究[D].国防科学技术大学研究生院, 2007: 4-9.
- [7] 高晖.无人机航路规划研究[J].南京航空航天大学学报, 33(2): 135-138, 2001.
- [8] 李华伟,多无人机协同任务规划研究与实现[D].西安电子科技大学, 27-33, 2014.
- [9] 张雷,多无人机协同侦察任务决策研究[D].西北工业大学, 9-38,2016.

附录

```
hold on
%%%%%%%%%起始点
% b = [110,0]*1000/38.2;
b = [110,0];
scatter(b(:,1),b(:,2),'filled'k');
dd = [110,55];
scatter(dd(:,1),dd(:,2),'filled'k');
%%%%%%%%%读取地形数据
z = xlsread('1.xlsx');
%%%%%%%%%剖分网格
x = 1:44.5:2913;
x1 = 38.2*(x-1)/1000;
y = 1:44.5:2775;
y1 = 38.2*(y-1)/1000;
count = 1;
for i = 1:length(y)
    for j = 1:length(x)
        if (z(floor(y(i)),floor(x(j)))<3000 && y(i)>100)
            u(i,j,1) = x1(j);
            u(i,j,2) = y1(i);
            test(:,count) = u(i,j,:);
            count=count+1;
%            u(i,j) = 1;
        else
            u(i,j,1) = 0;
            u(i,j,2) = 0;
        end
    end
end
end
%%%%%%%%%网格按出发点分两块
count = 1;
count1 = 1;
for i = 1:length(test)
    if (test(2,i)<52)
        test_half_0(:,count) = test(:,i);
        count=count+1;
    else
        test_half_1(:,count1) = test(:,i);
        count1=count1+1;
    end
end
```

```
end
%
% % % % % % 每块网格分为食物小块，按6度角划分
% for i = 1:15
%     j = 6*i;
%     a{i} = DIVIDE(test_half_0,j,110,0,0,0);
% end
for i = 1:15
    j = 6*i;
    c{i} = DIVIDE(test_half_0,j,110,55,0,0);
end
% % % % % 模拟退火算法
% for i = 1:15
%     result1(i) = mnthsf(a{i},110,0);
% end
% result2 = result1 - sum(result1)/15;
for i = 1:15
    result3(i) = mnthsf(c{i},110,55);
end
% result4 = result3 - sum(result3)/15;
% % % % % % % 模型最优值求解
% bi = 0;
% while ( bi < 0.7)
% for i = 1:15
%     if result1(i) == min(result1)
%         min_a = i;
%     else if result1(i) == max(result1)
%         max_a = i;
%     end
% end
% end
% k=0;
% for i = 1:15
%     if k == i
%         continue;
%     end
%     j = 6*i;
%     if i == max_a && i < 15
%         a{i} = DIVIDE(test_half_0,j,110,0,0,-3);
%         a{i+1} = DIVIDE(test_half_0,j,110,0,3,0);
%         i = i+1;
%         k = i;
%     else if i+1 == max_a && i == 14
%         a{i} = DIVIDE(test_half_0,j,110,0,0,3);
```

```
%      a{i+1} = DIVIDE(test_half_0,j,110,0,-3,0);
%      end
%      end
%      if i == min_a && i < 15
%          a{i} = DIVIDE(test_half_0,j,110,0,0,3);
%          a{i+1} = DIVIDE(test_half_0,j,110,0,-3,0);
%          i = i+1;
%          k = i;
%      else if i+1 == min_a && i == 14
%          a{i} = DIVIDE(test_half_0,j,110,0,0,-3);
%          a{i+1} = DIVIDE(test_half_0,j,110,0,3,0);
%      end
%      end
% end
% for i = 1:15
%     [result1(i),b] = mnthsf(a{i},110,0);
% end
% result2 = result1 - sum(result1)/15;
% bi = min(result1)/max(result1)
% end

bi=0
while ( bi <0.6)
for i = 1:15
    if result3(i) == min(result3)
        min_a = i;
    else if result3(i) == max(result3)
        max_a = i;
    end
end
end
end
k=0;
for i = 1:15
    if k == i
        continue;
    end
    j = 6*i;
    if i == max_a && i < 15
        c{i} = DIVIDE(test_half_0,j,110,55,0,-3);
        c{i+1} = DIVIDE(test_half_0,j,110,55,3,0);
        i = i+1;
        k = i;
    else if i+1 == max_a && i == 14
        c{i} = DIVIDE(test_half_0,j,110,55,0,3);
```



```

        c{i+1} = DIVIDE(test_half_0,j,110,55,-3,0);
    end
end
if i == min_a && i < 15
    c{i} = DIVIDE(test_half_0,j,110,55,0,3);
    c{i+1} = DIVIDE(test_half_0,j,110,55,-3,0);
    i = i+1;
    k = i;
else if i+1 == min_a && i == 14
    c{i} = DIVIDE(test_half_0,j,110,55,0,-3);
    c{i+1} = DIVIDE(test_half_0,j,110,55,3,0);
end
end
end
for i = 1:15
    [result3(i),aaa] = mnthsf(c{i},110,55);
end
result4 = result3- sum(result3)/15;
bi = min(result3)/max(result3);
end

%%%%%%按角度分区域
function [ P ] = DIVIDE( test_half_0,j,a,b,q,p)
P = [];
count = 1;
for i = 1:length(test_half_0(1,:))
    if (abs((test_half_0(2,i)-b)/(test_half_0(1,i)-a)) < tand(j+p) &&
abs((test_half_0(2,i)-b)/(test_half_0(1,i)-a)) >tand(j-6+q))
        P(:,count) = test_half_0(:,i);
        count=count+1;
    end
end
end
if isempty(P)
    P(:,1) = [a b];
end

%%%%%%按斜率分区域(queation4)
% function [ P,Q,R ] = DIVIDE(test_half_0,k,j)
% P = [];
% count = 1;
% count1 = 1;
% count2 = 1;

```

```
% for i = 1:length(test_half_0(1,:))
%     if ( test_half_0(2,i) < (-1.5+k)*test_half_0(1,i)+120 && test_half_0(2,i)>=0 )
%         P(:,count) = test_half_0(:,i);
%         count=count+1;
%     else if( test_half_0(2,i) > (-1.5+k)*test_half_0(1,i)+120 && test_half_0(2,i)<
(-1+j)*test_half_0(1,i)+150 )
%         Q(:,count1) = test_half_0(:,i);
%         count1=count1+1;
%     else if( test_half_0(2,i)>(-1+j)*test_half_0(1,i)+150 )
%         R(:,count2) = test_half_0(:,i);
%         count2=count2+1;
%     end
% end
% end
% end
% if isempty(P)
% P(:,1) = [110 0];
% end
% if isempty(P)
% Q(:,1) = [110 0];
% end
% if isempty(P)
% Q(:,1) = [110 0];
% end
% end
% end
```

```
%%%%%%模拟退火算法
function [S0,Sum] = mnthsf(Data,a,b)
% Data = load('sj.txt');
% Data = load('sj1.txt');
% Data = load('sj2.txt');
% x = Data(1,:);
% y = Data(2,:);
x = [a;Data(1,:)'];
y = [b;Data(2,:)'];
d=zeros(length(x));
for i=1:length(x)
    for j=1:length(x)
        if i == j
            d(i,j) = 0;
        else
            d(i,j) = ((x(i)-x(j))^2+(y(i)-y(j))^2)^(1/2);
        end
    end
end
end
```

```

end
S0=[];Sum=inf;
rand('state',sum(clock));
for j=1:1000
    S=[1 1+randperm(length(x)-2,length(x));
    temp=0;
    for i=1:length(x)-1
        temp=temp+d(S(i),S(i+1));
    end
    if temp<Sum
        S0=S;Sum=temp;
    end
end
end
e=0.1^30;L=20000;at=0.999;T=1;
% 计算最短路径
for k=1:L
    c=2+floor((length(x)-2)*rand(1,2));
    c=sort(c);
    c1=c(1);c2=c(2);
    df=d(S0(c1-1),S0(c2))+d(S0(c1),S0(c2+1))-d(S0(c1-1),S0(c1))-d(S0(c2),S0(c2+1));
    if df<0
        S0=[S0(1:c1-1),S0(c2:-1:c1),S0(c2+1:length(x))];
        Sum=Sum+df;
    elseif exp(-df/T)>rand(1)
        S0=[S0(1:c1-1),S0(c2:-1:c1),S0(c2+1:length(x))];
        Sum=Sum+df;
    end
    T=T*at;
    if T<e
        break;
    end
end
end
S0;
Sum;
end

function [Shortest_Route,Shortest_Length] = yq(C)
m = 10;
Alpha = 1;
Beta = 5;
Rho = 0.1;

max_iter = 300;
Q = 100;

```

```

n = size(C,1);      %n 表示问题的规模（城市个数）
D = zeros(n,n);     %D 表示完全图的赋权邻接矩阵
D_2 = zeros(1,n);

start = [110,0];

for i = 1:n
    D_2(1,i) = ((C(i,1)-start(1,1))^2+(C(i,2)-start(1,2))^2)^0.5;
    for j = 1:n
        if i ~= j
            D(i,j) = ((C(i,1)-C(j,1))^2+(C(i,2)-C(j,2))^2)^0.5;
        else
            D(i,j) = eps;      %i=j 时不计算，应该为 0，但后面的启发因子要取倒数，用 eps（浮点相对精度）表示
        end
        D(j,i)=D(i,j);      %对称矩阵
    end
end

Eta = 1./D;          %Eta 为启发因子，这里设为距离的倒数

Tau = ones(n,n);     %Tau 为信息素矩阵

Tabu = zeros(m,n);   %存储并记录路径的生成

iter = 1;            %迭代计数器，记录迭代次数

R_best = zeros(max_iter,n); %各代最佳路线

L_best = inf.*ones(max_iter,1); %各代最佳路线的长度

L_ave=zeros(max_iter,1); %各代路线的平均长度

while iter <= max_iter %停止条件之一：达到最大迭代次数，停止

%%第二步：将 m 只蚂蚁放到 n 个城市上

    Randpos = []; %随即存取

    for i = 1:(ceil(m/n))
        Randpos = [Randpos,randperm(n)];
    end

    Tabu(:,1) = (Randpos(1,1:m))'; %此句不太理解？

```

%% 第三步: m 只蚂蚁按概率函数选择下一座城市, 完成各自的周游

for j = 2:n %所在城市不计算

 for i=1:m

 visited = Tabu(i,1:(j-1)); % 记录已访问的城市, 避免重复访问

 J = zeros(1,(n-j+1)); % 待访问的城市

 P = J; % 待访问城市的选择概率分布

 Jc = 1;

 for k = 1:n

 if length(find(visited==k)) == 0 % 开始时置 0

 J(Jc) = k;

 Jc = Jc+1; % 访问的城市个数自加 1

 end

 end

% 下面计算待选城市的概率分布

 for k = 1:length(J)

 P(k) = (Tau(visited(end),J(k))^Alpha)*(Eta(visited(end),J(k))^Beta);

 end

 P = P/(sum(P));

% 按概率原则选取下一个城市

 Pcum = cumsum(P); % cumsum, 元素累加即求和

 Select = find(Pcum>=rand); % 若计算的概率大于原来的就选择这条路线

 to_visit = J(Select(1));

 Tabu(i,j) = to_visit;

 end

end

if iter>=2

 Tabu(1,:) = R_best(iter-1,:);

end

%% 第四步: 记录本次迭代最佳路线

L = zeros(m,1); % 开始距离为 0, m*1 的列向量

for i=1:m

 R = Tabu(i,:);

 for j=1:(n-1)

 L(i)=L(i) + D(R(j),R(j+1)); % 原距离加上第 j 个城市到第 j+1 个城市的距离

 end

```
L(i) = L(i) + D(R(1),R(n));
%   if D_2(1,Tabu(i,1)) < D_2(1,Tabu(i,n))
%       L(i)= L(i) + D_2(1,Tabu(i,1));
%   else
%       L(i)= L(i) + D_2(1,Tabu(i,n));
%   end

% L(i)=L(i)+D(R(1),R(n));      %一轮下来后走过的距离

end

L_best(iter)=min(L);          %最佳距离取最小

pos=find(L==L_best(iter));

R_best(iter,:)=Tabu(pos(1),:); %此轮迭代后的最佳路线

L_ave(iter)=mean(L);          %此轮迭代后的平均距离

iter=iter+1                    %迭代继续

%%第五步：更新信息素

Delta_Tau = zeros(n,n);       %开始时信息素为 n*n 的 0 矩阵

for i=1:m

    for j=1:(n-1)

        Delta_Tau(Tabu(i,j),Tabu(i,j+1))=Delta_Tau(Tabu(i,j),Tabu(i,j+1))+Q/L(i);

        %此次循环在路径 (i, j) 上的信息素增量

    end

    Delta_Tau(Tabu(i,n),Tabu(i,1))=Delta_Tau(Tabu(i,n),Tabu(i,1))+Q/L(i);

    %此次循环在整个路径上的信息素增量

end
```

```
Tau=(1-Rho).*Tau+Delta_Tau; %考虑信息素挥发，更新后的信息素
```

```
%%第六步：禁忌表清零
```

```
Tabu=zeros(m,n); %直到最大迭代次数
```

```
end
```

```
%%第七步：输出结果
```

```
Pos=find(L_best==min(L_best)); %找到最佳路径（非0为真）
```

```
Shortest_Route=R_best(Pos(1),:); %最大迭代次数后最佳路径
```

```
Shortest_Length = L_best(Pos(1)); %最大迭代次数后最短距离
```

```
% subplot(1,2,1) %绘制第一个子图形
```

```
%画路线图的子函数
```

```
% subplot(1,2,2) %绘制第二个子图形
```

```
% plot(L_best)
```

```
%
```

```
% hold on %保持图形
```

```
%
```

```
% plot(L_ave,'r')
```

```
%
```

```
% title('平均距离和最短距离') %标题
```

```
end
```

```
clc
```

```
clear
```

```
SearchAgents_no = 50;
```

```
Max_iter = 300;
```

```
dim = 2;
```

```
ub = 2;
```

```
lb = -2;
```

```
Alpha_pos = zeros(72,dim);
```

```
Alpha_score = inf;
```



```
Beta_pos = zeros(72,dim);
Beta_score = inf;

Delta_pos = zeros(72,dim);
Delta_score = inf;

Positions = initialization(SearchAgents_no,dim,ub,lb);

Convergence_curve = zeros(1,Max_iter);

l=0;

while l < Max_iter
    for i=1:size(Positions,3)

        Flag4ub = Positions(:,i)>ub;
        Flag4lb = Positions(:,i)<lb;
        Positions(:,i) = (Positions(:,i).*(~(Flag4ub+Flag4lb))) + 1.4.*Flag4ub+1.4.*Flag4lb;

        fitness = fun(Positions(:,i));

        if fitness < Alpha_score
            Alpha_score = fitness;
            Alpha_pos = Positions(:,i);
        end

        if fitness > Alpha_score && fitness < Beta_score
            Beta_score = fitness;
            Beta_pos = Positions(:,i);
        end

        if fitness>Alpha_score && fitness>Beta_score && fitness<Delta_score
            Delta_score = fitness;
            Delta_pos = Positions(:,i);
        end
    end

    a = 2-1*((2)/Max_iter);
```

```
for i = 1:size(Positions,3)
    for j=1:size(Positions,1)

        r1 = rand();
        r2 = rand();

        A1 = 2*a*r1-a;
        C1 = 2*r2;
        D_alpha = abs(C1*Alpha_pos(j,:)-Positions(j,:,i));
        X1 = Alpha_pos(j,:)-A1*D_alpha;

        r1=rand();
        r2=rand();
        A2=2*a*r1-a;
        C2=2*r2;
        D_beta = abs(C2*Beta_pos(j,:)-Positions(j,:,i));
        X2 = Beta_pos(j,:)-A2*D_beta;

        r1=rand();
        r2=rand();
        A3=2*a*r1-a;
        C3=2*r2;
        D_delta = abs(C3*Delta_pos(j,:)-Positions(j,:,i));
        X3 = Delta_pos(j,:)-A3*D_delta;

        Positions(j,:,i) = (X1+X2+X3)/3;
    end
end
l = l+1
Convergence_curve(l) = Alpha_score;
end
```

Convergence_curve(l)/6

function [R] = rate(B,p_t,x)

```
t = 0;    %% 记录飞机进入时间
d = [];   %% 每一秒钟的数据传输率
p0 = 4.314e-10;
```

```

if x <= 1500                                %% 1500 是个极限高
度 超过这个高度 30 度角的欧式距离大于 3Km
    while (3^0.5*x - (60/3.6)*t) >= 0        %% 跳出条件= 飞机飞过目
标点正上，到达圆心点
        D = (x^2 + (3^0.5*x - (60/3.6)*t)^2)^0.5;    %% D 表示每一秒对应的欧氏
距离
        r = B*log2(1+p_t/(p0*D^2));                %% 每一秒中传出的数据
量
        d = [d,r];
        t = t + 1;
    end
    R = sum(d)*2;                                %% 开始-》正上空-》
离开，两个半径距离
    R = R/1024/1024;
end

if x >= 1500
    while (3000^2 - x^2)^0.5 - (60/3.6)*t >= 0
        D = (x^2 + ((3000^2 - x^2)^0.5 - (60/3.6)*t)^2)^0.5;
        r = B*log2(1+p_t/(p0*D^2));                %% 每一秒中传出的数据
量
        d = [d,r];
        t = t + 1;
    end
    R = sum(d)*2;                                %% 开始-》正上空-》
离开，两个半径距离
    R = R/1024/1024;
end
end
end

```