

# 无人机在边境勘测中的路径优化问题研究

于海宝<sup>1,2</sup>, 许卫东<sup>1</sup>, 魏 洪<sup>1</sup>, 谢雨谋<sup>3</sup>, 叶雨佳<sup>1</sup>,  
刘悦宝<sup>1</sup>, 郑龙华<sup>1</sup>, 刘 庆<sup>1</sup>

<sup>1</sup>陆军工程大学野战工程学院, 江苏 南京

<sup>2</sup>陆军边防69348部队, 新疆 霍城

<sup>3</sup>新疆察布查尔县人民武装部, 新疆 察布查尔县

收稿日期: 2024年9月14日; 录用日期: 2024年10月7日; 发布日期: 2024年10月16日

## 摘 要

使用复合翼无人机实施边境勘测, 已经成为部分边防部队边境勘测的重要手段。采用无人机进行边境勘测, 极大提高了作业效率, 实现高山、河流、雪地等恶劣艰苦环境的巡逻, 尤其对担负新疆、西藏边防线上的巡逻任务来说, 无人机代替分队巡逻、勘测具有重要意义。针对无人机对边防线上多个边境目标实施勘测时的路径优化问题, 本文通过建立数学模型, 在人工蜂群算法(Artificial Bee Colony Algorithm, ABC)、模拟退火算法(Simulated Annealing, SA)、遗传算法(Generation Algorithm, GA)、蚁群算法(Ant Colony Algorithm, ACA)等智能算法基础上进行优化, 尝试采用头脑风暴优化算法(Brain Storm Optimization Algorithm, BSO)进行路径优化, 并仿真实验求解, 通过对求解结果、收敛度进行综合分析, 得出BSO算法较其他算法收敛度较好, 路径更优化, 极大节省了巡逻时间、提高作业效率、为当前部分队采用无人机进行边境勘测的路径优化问题提供新方法。

## 关键词

边境勘测, 路径优化, BSO

# A Study of the Path Optimisation Problem of Unmanned Aerial Vehicles in Border Survey

Haibao Yu<sup>1,2</sup>, Weidong Xu<sup>1</sup>, Hong Wei<sup>1</sup>, Yumou Xie<sup>3</sup>,  
Yujia Ye<sup>1</sup>, Yuebao Liu<sup>1</sup>, Longhua Zheng<sup>1</sup>, Qing Liu<sup>1</sup>

<sup>1</sup>College of Field Engineering, Army Engineering University of PLA, Nanjing Jiangsu

<sup>2</sup>Army Border Defence Unit 69348, Huocheng Xinjiang

<sup>3</sup>People's Armed Forces Department of Chabchal County, Chabchal County Xinjiang

Received: Sep. 14<sup>th</sup>, 2024; accepted: Oct. 7<sup>th</sup>, 2024; published: Oct. 16<sup>th</sup>, 2024

**文章引用:** 于海宝, 许卫东, 魏洪, 谢雨谋, 叶雨佳, 刘悦宝, 郑龙华, 刘庆. 无人机在边境勘测中的路径优化问题研究[J]. 应用数学进展, 2024, 13(10): 4563-4571. DOI: 10.12677/aam.2024.1310437

## Abstract

The use of composite-winged drones to carry out border surveys has become an important means of border surveys for some border defence forces. The use of UAVs to carry out border surveys greatly improves operational efficiency and achieves the patrolling of harsh and difficult environments such as high mountains, rivers, snow, etc. Especially for patrolling tasks on the borderline of Xinjiang and Tibet, UAVs are of great significance, unlike detachment patrolling and surveying. This paper addresses the path optimization problem for unmanned aerial vehicles (UAVs) conducting surveys on multiple border targets along the border defense line. By establishing a mathematical model, we optimize the problem based on various intelligent algorithms, including the Artificial Bee Colony Algorithm (ABC), Simulated Annealing (SA), Genetic Algorithm (GA), and Ant Colony Algorithm (ACA). Additionally, we explore path optimization using the Brain Storm Optimization Algorithm (BSO) and conduct simulation experiments for problem-solving. Through a comprehensive analysis of the results and convergence performance, we find that the BSO algorithm demonstrates better convergence and more optimized paths compared to the other algorithms, significantly reducing patrol time and enhancing operational efficiency. This provides a new approach to the path optimization issue for UAVs used in border surveying in some current units.

## Keywords

Border Survey, Path Optimisation, BSO

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 概述

在近期发生的纳卡冲突、俄乌战争和巴以冲突等军事对抗中，多种类型无人机(Unmanned Aerial Vehicles, UAVs)出现在战场中，执行多种任务。比如俄乌冲突中，俄军调配“海鹰”-10、“海鹰”-30、“前哨”、“石榴”等无人机，用于侦察、火力校射等任务，开辟了大规模、高强度的无人战场；巴以冲突中，哈马斯武装组织利用无人机对以色列的“梅卡瓦 4M”主战坦克发动攻击，出现了世界上第一个装备有主动防御系统的先进主战坦克被无人机击毁的案例。无人机在战争中发挥着越来越重要的作用，其应用范围涵盖了侦察与监视、通信中继、打击目标、电子战等多个方面。随着技术的不断进步，无人机在军事领域的应用更加广泛和深入，在日常分队执勤、边境勘测中也将发挥重要意义。

对于边海防分队，常年驻守在高山海岛等艰苦恶劣地形条件下，不利于分队执行边境勘测任务，无人机可搭载高清摄像头和激光雷达等设备，快速获取边境地区的高精度地形数据，这些数据对于边境管理、军事防御和自然资源管理等方面都具有重要意义；对边境区域可进行实时监控，并通过高空悬停和长航时能力，可持续监测边境线的动态，及时发现非法越境、走私和偷渡等违法活动。与传统的人力步行勘测、骑马探测等方式相比，无人机勘测具有成本较低、覆盖区域较广、灵活性较强、采集数据较准确、效率较高等优点。当前，在遂行多目标边境勘测任务中，通常使用复合翼侦察无人机实施现地勘测，对于多目标勘测，如何实施勘测路径优化问题需要进一步进行关注和研究，以进一步提高勘测效率、缩短侦察时间，达到抢占先机的目的。

## 2. 建立数学模型

### 2.1. 基本模型的建立

无人机在边境勘测的路径优化问题，可参考旅行商问题(Traveling Salesman Problem, 简称 TSP)加以分析。

#### 2.1.1. TSP 问题

旅行商问题(又称货郎担问题), 可描述为: 有一人员去各地配送货物, 从城市  $M_1$  出发, 要遍访城市  $M_2, M_3, M_4, \dots, M_n$  各一次, 最后返回到城市  $M_1$ , 其中  $M_i$  到  $M_j$  的旅费为  $V_{ij}$ , 按照怎样的次序访问这些城市, 才能使总旅费最少。

定义变量  $X_{ij}$  表示人员是否从城市  $M_i$  卖货后去城市  $M_j$ ,

$$x_{ij} = \begin{cases} 1 & \text{人员从城市 } M_i \text{ 卖货后去城市 } M_j \\ 0 & \text{人员从城市 } M_i \text{ 卖货后不去城市 } M_j \end{cases}$$

定义总费用为  $Z$ :

$$\min z = \sum_{i=1}^n \sum_{j=1}^n v_{ij} x_{ij}$$

人员在遍访城市售货时, 为防止出现子回路, 需要附加约束:

$$w_i - w_j + nx_{ij} \leq n-1, \quad i=1, \dots, n, \quad j=2, \dots, n$$

$$w_1 = 0, \quad 1 \leq w_i \leq n-1, \quad i=2, 3, \dots, n$$

建立数学模型:

$$\begin{aligned} \min z &= \sum_{i=1}^n \sum_{j=1}^n v_{ij} x_{ij} \\ \text{s.t.} &\begin{cases} \sum_{j=1}^n x_{ij} = 1, & i=1, 2, \dots, n \\ \sum_{i=1}^n x_{ij} = 1, & j=1, 2, \dots, n \\ w_i - w_j + nx_{ij} \leq n-1, & i=1, 2, \dots, n; j=2, 3, \dots, n \\ w_1 = 0, 1 \leq w_i \leq n-1, & i=2, 3, \dots, n \\ x_{ij} = 0 \text{ 或 } 1, \end{cases} \end{aligned}$$

#### 2.1.2. TSP 问题目前研究现状

TSP 问题作为一个经典的组合优化问题, 在规模较小, 计算量较少的情况下可采用精确、近似算法, 例如可采用线性规划、分支定界算法, 但此方法不适用规模较大、计算量大的情况。后期发展采用混合人工蜂群算法(Artificial Bee Colony Algorithm, ABC)、模拟退火算法(Simulated Annealing, SA)、遗传算法(Generation Algorithm, GA)、蚁群算法(Ant Colony Algorithm, ACA)等智能算法进行优化, 王港华对物流配送类小规模 TSP 问题的遗传算法求解进行了研究, 王珺研究了基于模拟退火算法的民航运输路径优化技术, 孙鉴等人提出了基于 Spark 的并行模拟退火算法求解 TSP 的方法[1]。本文对无人机的多目标边境勘测任务的路径优化问题, 尝试采用头脑风暴优化算法(BSO)求解, 并进行收敛度分析, 得出无人机对边境目标实施勘测的路径优化问题的最优解。

## 2.2. 头脑风暴优化算法(BSO)

头脑风暴优化算法的灵感来自于人类社会性活动(头脑风暴), 遵循延迟评判、自由畅想、以量求质和综合改善原则的智能优化算法[2][3]。以求解问题过程为原型, 抽象成优化算法。其被广泛应用于解决多方面问题, 如机器人协作问题[4]、数值优化问题[5]、缓存优化问题[6]等, 算法流程图见图 1。

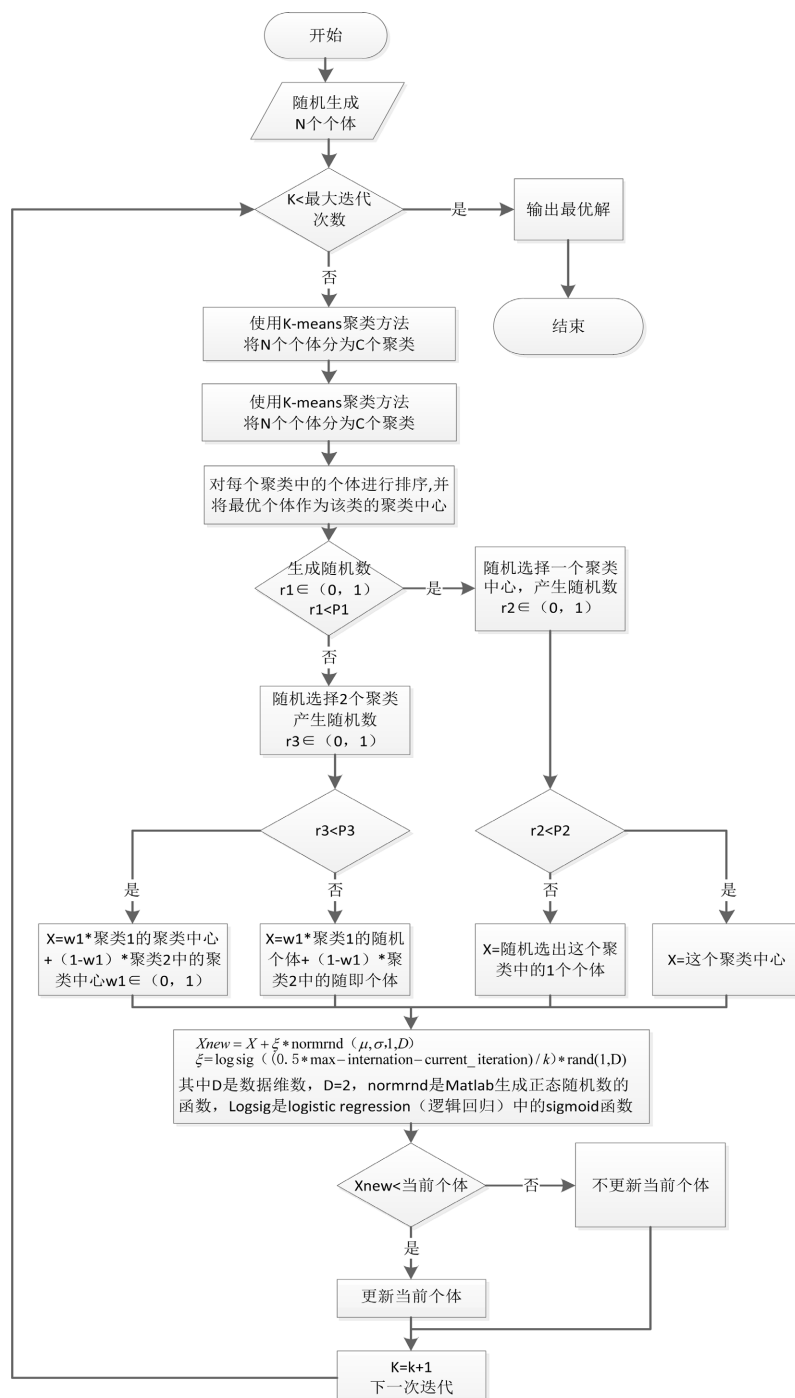


Figure 1. Flowchart of brainstorming optimisation (BSO) algorithm [7]

图 1. 头脑风暴优化(BSO)算法流程图[7]

3. 算例与结果分析

3.1. 问题描述

某边防分队根据上级任务部署，对 C 国与 H 国某段边境线之间的 22 个重点勘测目标(以距离分队战立点最近的点 O 为坐标原点，其余坐标为其相对距离，见表 1、图 2)进行勘测，形成边境勘测报告，为指挥员判断边境情况提供有利实际情报支撑。该边防分队携带复合翼侦察无人机已抵达原点 O(0,0)，该复合翼侦察无人机电池完好、电量充足，可持续工作 4 小时，假设无论采取何种路线勘测，均能在 4 小时之内完成任务并回到原点；该无人机可飞行高度大于勘测地的山脉高度，且均能按上级要求完成对重点目标的勘测。无人机分队应如何选择勘测路径，使得无人机勘测路径最短，较快完成边境勘测任务。

Table 1. Information on the coordinates of points to be surveyed along a borderline

表 1. 某边境线需勘测点坐标信息表

| 序号   | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    |
|------|------|------|------|------|------|------|------|------|------|
| X 坐标 | 0    | 1000 | 725  | 1127 | 1105 | 655  | 1203 | 1235 | 1686 |
| Y 坐标 | 0    | 525  | 1108 | 1758 | 2900 | 3125 | 3750 | 4803 | 5608 |
| 序号   | 10   | 11   | 12   | 13   | 14   | 15   | 16   | 17   | 18   |
| X 坐标 | 2203 | 2255 | 2304 | 3525 | 3728 | 4568 | 4672 | 6234 | 6783 |
| Y 坐标 | 5650 | 5050 | 5955 | 6050 | 5734 | 5487 | 6375 | 6735 | 5102 |
| 序号   | 19   | 20   | 21   | 22   | 23   |      |      |      |      |
| X 坐标 | 7593 | 7421 | 6893 | 7458 | 5374 |      |      |      |      |
| Y 坐标 | 3795 | 2232 | 1537 | 328  | 423  |      |      |      |      |

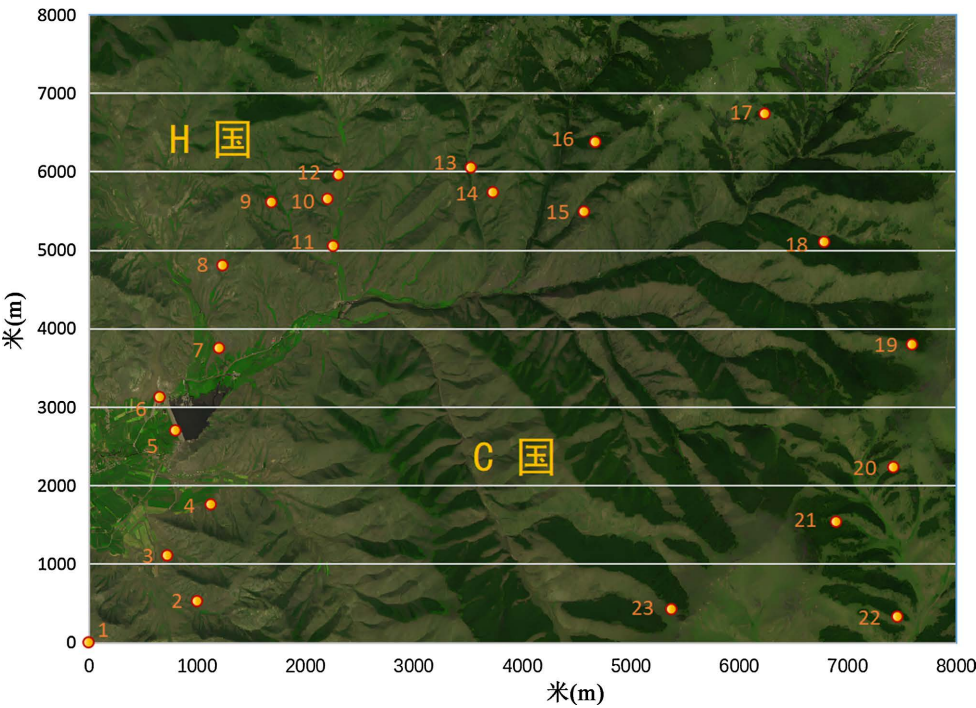


Figure 2. Distribution of points to be surveyed along a borderline  
图 2. 某边境线需勘测点分布图

### 3.2. 根据 TSP 问题模型建立数学模型

边防分队需利用复合翼无人机对重点目标完成勘测任务, 假设对  $n$  个目标实施边境勘测, 每两个目标点之间距离为  $d_{ij}$  ( $i, j = 1, 2, \dots, n$ ; 且  $i \neq j$ ), 求从点  $O(0, 0)$  (以出发点为原点, 正东方向为  $x$  轴, 正北为  $y$  轴) 出发, 遍历所有目标点有且仅有 1 次后回到出发点  $O(0, 0)$  的最短距离。

建立数学模型如下:

$$\min f = \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \quad (1)$$

$$d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, \quad i \neq j, i \in V, j \in V \quad (2)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i \in V \quad (3)$$

$$\text{s.t.} \quad \sum_{i=1}^n x_{ij} = 1, \quad j \in V \quad (4)$$

$$\sum_{j=s}^n \sum_{j \in s} x_{ij} \leq |s| - 1, \quad \forall s \subset V, 2 \leq |s| \leq n-1 \quad (5)$$

$$x_{ij} \in \{0, 1\} \quad (6)$$

(1) 是目标函数, 即从第一个点  $O(0, 0)$  出发, 遍历  $n$  个点后回到点  $O(0, 0)$  的最短距离; (2)  $d_{ij}$  为第  $i$  点到第  $j$  点的欧氏距离; (3)、(4) 为无人机经过每一个点有且仅有一次; (5) 无人机不能重复经过任何一个点(出发点除外); (6) 无人机经过某一点该点状态改为 1, 否则为 0。  $V$  为所有点的序号集合;  $s$  为第 2 个点到  $n-1$  点。

### 3.3. 利用 BSO 算法求解

采用最大迭代次数为 100, 种群为 50, 聚类数目为 5, 随机替换一个聚类中心的概率为 0.1, 选择 1 个聚类的概率 0.5, 选择两个聚类的概率 0.5, 选择 1 个聚类中心的概率 0.3, 选择两个聚类中心的概率 0.2。 Matlab 核心代码见图 3~6。

```
% 参数初始化
MAXGEN=100; %最大迭代次数
NIND=50; %种群数目
cluster_num=5; %聚类数目
p_replace=0.1; %用随机解替换一个聚类中心的概率
p_one=0.5; %选择 1 个聚类的概率
p_two=1-p_one; %选择 2 个聚类的概率。 p_two=1-p_one
p_one_center=0.3; %选择 1 个聚类中聚类中心的概率
p_two_center=0.2; %选择 2 个聚类中聚类中心的概率
%% 种群初始化
Population=InitPop(NIND,N);
%% 主循环
gen=1; %计数器初始化
BestPop=zeros(MAXGEN,N); %记录每次迭代过程中全局最优个体
BestL=zeros(MAXGEN,1); %记录每次迭代过程中全局最优个体的总距离
while gen<=MAXGEN
    %% 计算目标函数值
    Obj=ObjFunction(Population,dist);
    %% K-means 聚类
    [Idx,C,sumD,D]=kmeans(Obj,cluster_num,'Distance','cityblock','Replicates',2);
    cluster=cell(cluster_num,2); %将解储存在每一个聚类中
    order_cluster=cell(cluster_num,2); %将储存在每一个聚类中的解按照目标函数值排序
    for i=1:cluster_num
        cluster{i,1}=Population(Idx==i,:); %将个体按照所处的聚类编号储存在对应的聚类中
        cluster_row(i)=size(cluster{i,1},1); %计算当前聚类中个体数目
        for j=1:cluster_row(i)
            Individual=cluster{i,1}(j,:); %当前聚类中第 j 个个体
            cluster{i,2}(j,1)=ObjFunction(Individual,dist); %计算当前聚类中第 j 个个体的目标函数值
        end
        [order_cluster{i,2},order_index]=sort(cluster{i,2}); %将当前聚类中的所有个体按照目标函数值从小到大
        %的顺序进行排序
        order_cluster{i,1}=cluster{i,1}(order_index,:); %将当前聚类中的所有个体按照排序结果重新排列
        order_index=0; %重置排序序号
    end
    gen=gen+1;
end
```

Figure 3. Core code 1

图 3. 核心代码 1



```

%% 以一定的概率随机从 m 个聚类中心中选出一个聚类中心，并用一个新产生的随机解更新这个被选中的聚类中心
R1=rand(1,1);
if R1<=p_replace
%随机选择一个聚类中心
repalce_cluster_num=randi([1,cluster_num],1,1);
%随机产生一个解
replace_solution=randperm(N);
%并用这个新产生的随机解更新这个被选中的聚类中心
order_cluster{repalce_cluster_num,1}(1,:)=replace_solution;
%计算新解的目标函数值
replace_solution_fitness=ObjFunction(replace_solution,dist);
%将新解的目标函数值储存在 order_cluster 中
order_cluster{repalce_cluster_num,2}(1,:)=replace_solution_fitness;
end
%% 更新这 n 个个体
for i=1:NIND
%如果随机数小于选择 1 个聚类的概率，则随机选择 1 个聚类
if rand()<p_one
select_one_cluster=randi([1,cluster_num],1,1); %选择选择一个聚类
%如果随机数小于选择 1 个聚类中聚类中心的概率，或当前聚类中只有一个个体
if rand()<p_one_center||cluster_row(select_one_cluster)==1
select_ind=order_cluster{select_one_cluster,1}(1,:); %选择当前解的聚类中心（只有一个个体时，就选择该个体）
else
r_1= randi([2,cluster_row(select_one_cluster)],1,1); %随机选择当前聚类中除聚类中心外的其它个体序号
select_ind=order_cluster{select_one_cluster,1}(r_1,:); %随机选择当前聚类中除聚类中心外的其它个体
end
indi_temp=Swap(select_ind); %将该个体进行交换操作
else %如果随机数不小于选择 1 个聚类的概率，则随机选择 2 个聚类
cluster_two=[0,0]; %随机产生两个聚类的序号
while cluster_two(1,1)==cluster_two(1,2)
cluster_two=randi([1,cluster_num],1,2); %这两个聚类序号不能相同
end

```

Figure 4. Core code 2

图 4. 核心代码 2

```

%如果随机数小于选择 2 个聚类中聚类中心的概率，或当前两个聚类中都只有一个个体
if(rand()<p_two_center)|| (cluster_row(cluster_two(1,1))==1&&cluster_row(cluster_two(1,2))==1)
%选择这 2 个聚类中聚类中心
select_ind1=order_cluster{cluster_two(1,1),1}(1,:); %第一个被选择的聚类中心
select_ind2=order_cluster{cluster_two(1,2),1}(1,:); %第二个被选择的聚类中心
else
%如果第 1 个选择的聚类中只有一个个体
if cluster_row(cluster_two(1,1))==1
r_2=randi([2,cluster_row(cluster_two(1,2))],1,1); %选择第 2 个聚类中除聚类中心外的其它个体
select_ind1=order_cluster{cluster_two(1,1),1}(1,:); %第一个聚类中被选择的个体
select_ind2=order_cluster{cluster_two(1,2),1}(r_2,:); %第二个聚类中被选择的个体
elseif cluster_row(cluster_two(1,2))==1 %如果第 2 个选择的聚类中只有一个个体
r_3=randi([2,cluster_row(cluster_two(1,1))],1,1); %选择第 1 个聚类中除聚类中心外的其它个体
select_ind1=order_cluster{cluster_two(1,1),1}(r_3,:); %第一个聚类中被选择的个体
select_ind2=order_cluster{cluster_two(1,2),1}(1,:); %第二个聚类中被选择的个体
elseif cluster_row(cluster_two(1,1))>1&&cluster_row(cluster_two(1,2))>1
%%选择这 2 个聚类中除聚类中心外的其它个体
r_4=randi([2,cluster_row(cluster_two(1,1))],1,1);
r_5=randi([2,cluster_row(cluster_two(1,2))],1,1);
select_ind1=order_cluster{cluster_two(1,1),1}(r_4,:); %第一个聚类中被选择的个体
select_ind2=order_cluster{cluster_two(1,2),1}(r_5,:); %第二个聚类中被选择的个体
end
end
[child1,child2,min_index,start_c]=heuristic_crossover(select_ind1,select_ind2,dist); %启发式交叉算子
%如果子代个体 1 目标函数值更小，则将 indi_temp 赋值为 child1，否则赋值为 child2
if min_index==1
indi_temp=child1;
else
indi_temp=child2;
end

```

Figure 5. Core code 3

图 5. 核心代码 3

```

fit_indi_temp=ObjFunction(indi_temp,dist);
%如果 fit_indi_temp 比原来位置上的个体目标函数值更小, 则更新该位置上的个体
if fit_indi_temp<order_cluster{Idx(i),2}(1,:)
Population(i,:) =indi_temp(1,:);
end
end
%% 计算目标函数值
Obj=ObjFunction(Population,dist);
[min_len,min_index]=min(Obj); %当前种群中最优个体以及所对应的序号
BestL(gen,1)=min_len; %记录当前迭代中最优个体
BestPop(gen,:)=Population(min_index,:); %记录当前迭代中最优个体的总距离
%% 打印各代最优解
disp(['第',num2str(gen),'代最优个体的总距离为',num2str(min_len)]);
%% 计数器加 1
gen=gen+1;
end
%% 计算最终种群的目标函数值
Obj=ObjFunction(Population,dist);
[best_len,best_index]=min(Obj);
bestR=Population(best_index,:); %全局最优个体
PlotRoute(bestR,x,y);
%% 打印每次迭代的全局最优个体的总距离变化趋势图
figure;
plot(BestL,'LineWidth',1);
title('优化过程')
xlabel('迭代次数');
ylabel('总距离');
data1=BestL;
toc

```

Figure 6. Core code 4

图 6. 核心代码 4

仿真实验结果如下:

第 1 代最优个体的总距离为 62877.7623 米; 第 2 代最优个体的总距离为 52868.493 米; 第 3 代最优个体的总距离为 47296.452 米; 第 4 代最优个体的总距离为 37711.8539 米; 第 5 代最优个体的总距离为 36353.5003 米; 第 6 代最优个体的总距离为 33812.3365 米; 第 7 代最优个体的总距离为 30332.3101 米; 第 8 代最优个体的总距离为 30332.3101 米; 第 9 代最优个体的总距离为 29102.3541 米; 第 10 代最优个体的总距离为 29102.3541 米; 第 11 代最优个体的总距离为 29102.3541 米; 第 12 代最优个体的总距离为 29007.5297 米; 第 13 代最优个体的总距离为 28331.8461 米; 第 14 代最优个体的总距离为 28331.8461 米; 第 15 代至第 100 代最优个体的总距离均为 27995.9277 米, BSO 的收敛度图、路由图见图 7、图 8。

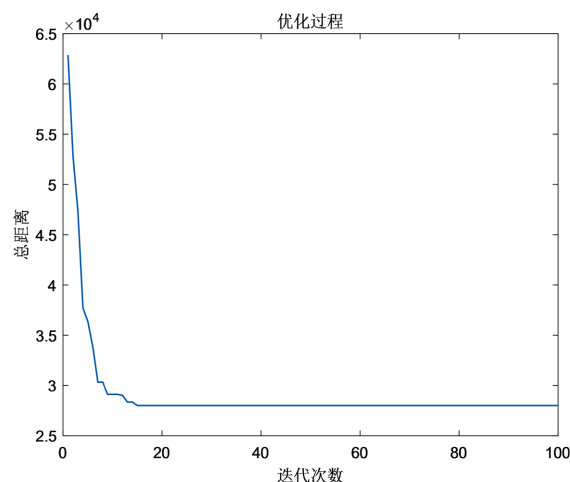
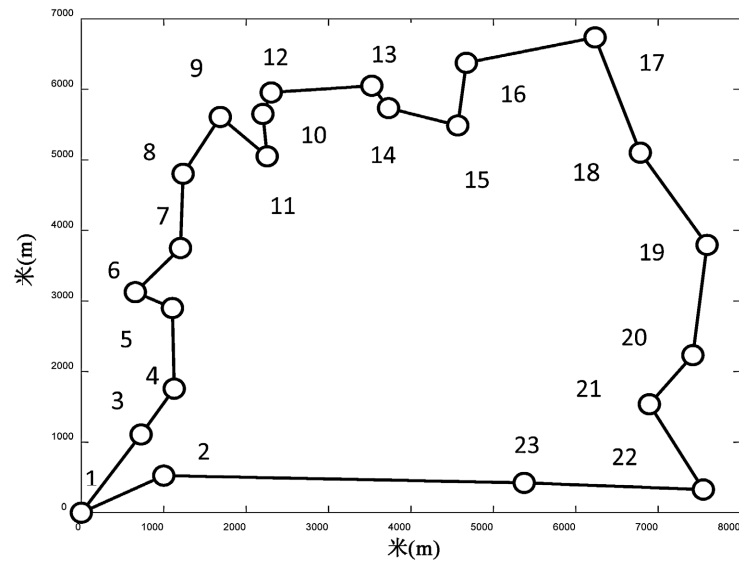


Figure 7. Convergence plot of the brainstorming optimisation algorithm BSO

图 7. 头脑风暴优化算法 BSO 的收敛度图





**Figure 8.** Routing diagram of brainstorming optimisation algorithm BSO  
**图 8.** 头脑风暴优化算法 BSO 的路由图

根据仿真实验结果, 无人机最优勘测路径为从原点  $O(0, 0)$  出发, 沿着勘测点  $3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 11 \rightarrow 10 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15 \rightarrow 16 \rightarrow 17 \rightarrow 18 \rightarrow 19 \rightarrow 20 \rightarrow 21 \rightarrow 22 \rightarrow 23 \rightarrow 2 \rightarrow 1$ , 使得无人机勘测路径最短, 为 27995.9277 米, 能够较快完成边境勘测任务。从 BOS 收敛度图可以看出其收敛度较好, 迭代到 15 代就能算出结果。利用 BOS 算法能较快解决当前部队使用无人机对边境多目标勘测时的路径优化问题, 尤其在高山海岛、不良天候等恶劣环境不便于人工勘测时, 极大地提高了边境勘测效率; 节省大量资源。

## 参考文献

- [1] 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, 2022.
- [2] 程适, 刘悦, 王雪萍, 等. 改进头脑风暴优化算法求解多模态多目标问题[J]. 华中科技大学学报(自然科学版), 2024, 52(6): 24-31.
- [3] 程适, 王雪萍, 刘悦, 等. 面向非线性方程组的学习型头脑风暴优化算法[J]. 计算机工程, 2023, 49(7): 47-54.
- [4] Yang, J. and Shi, Y. (2022) Automatic Synthesizing Multi-Robot Cooperation Strategies Based on Brain Storm Robotics. *Applied Soft Computing*, **120**, Article 108672. <https://doi.org/10.1016/j.asoc.2022.108672>
- [5] Cai, Z.H., Gao, S.C., Yang, X., et al. (2022) Alternate Search Pattern-Based Brain Storm Optimization. *Knowledge-Based Systems*, **238**, Article 107896. <https://doi.org/10.1016/j.knsys.2021.107896>
- [6] Jiang, Y.X., Chen, X., Zheng, F.C., et al. (2021) Brain Storm Optimization-Based Edge Caching in Fog Radio Access Networks. *IEEE Transactions on Vehicular Technology*, **70**, 1807-1820. <https://doi.org/10.1109/tvt.2021.3052971>
- [7] 李鑫, 余墨多, 姜庆超, 等. 基于分区搜索和强化学习的多模态多目标头脑风暴优化算法[J/OL]. 计算机应用研究, 1-12. <https://doi.org/10.19734/j.issn.1001-3695.2023.12.0588>, 2024-06-19.