

DOI:10.16356/j.1005-2615.2025.06.009

## 基于改进人工蜂群算法的无人飞行器路径协同规划

马梓元<sup>1</sup>, 刘伟鹏<sup>2</sup>, 胡春朝<sup>2</sup>, 龚华军<sup>1</sup>, 王新华<sup>1</sup>

(1. 南京航空航天大学自动化学院,南京 211106; 2. 上海机电工程研究所,上海 201109)

**摘要:** 为解决空战场景下的高亚声速无人飞行器(Unmanned aerial vehicle, UAV)的路径规划问题,提出一种改进的人工蜂群(Improved artificial bee colony, IABC)算法。综合考虑三维空间障碍与无人飞行器路径规划的协同问题,建立作战场景模型与目标函数;在雇佣蜂阶段引入了粒子群算法(Particle swarm optimization, PSO)降低搜索的盲目性,增强算法的搜索能力;在观察蜂阶段基于动态贪婪准则对迭代初期的蜜源进行局部平滑处理,进一步提升算法的收敛速度。为了验证算法有效性,对算法进行了仿真对比实验。仿真实验表明,IABC 算法继承了 ABC 与 PSO 算法的搜索优点,相较 ABC 算法,平均算法收敛速度提升 47.83%,算法收敛精度平均提升 53.49%。

**关键词:** 路径协同规划;三维空间障碍;改进人工蜂群;粒子群算法;动态贪婪准则

**中图分类号:** V279      **文献标志码:** A      **文章编号:** 1005-2615(2025)06-1124-11

## Path Cooperative Planning for Unmanned Aerial Vehicles Based on Improved Artificial Bee Colony Algorithm

MA Ziyuan<sup>1</sup>, LIU Weipeng<sup>2</sup>, HU Chunchao<sup>2</sup>, GONG Huajun<sup>1</sup>, WANG Xinhua<sup>1</sup>

(1. College of Automation Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 211106, China;  
2. Shanghai Institute of Mechanical and Electrical Engineering, Shanghai 201109, China)

**Abstract:** To solve the path planning problem of high-subsonic unmanned aerial vehicles (UAVs) in air combat scenarios, an improved artificial bee colony algorithm (IABC) is proposed. Firstly, by comprehensively considering the obstacles in three-dimensional space and the coordination problem of UAV's path planning, a combat scenario model and an objective function are established. Secondly, in the employed bee stage, the particle swarm optimization (PSO) algorithm is introduced to reduce the blindness while searching and enhance the search ability of the algorithm. Finally, in the onlooker bee stage, local smoothing processing is carried out on the food sources in the early stage of iteration based on the dynamic greedy criterion, which further improves the convergence speed of the algorithm. In order to verify the effectiveness of the algorithm, a simulation comparison experiment on the algorithm is conducted. The simulation experiment shows that the IABC algorithm inherits the search advantages of the ABC and PSO algorithms. Compared with the ABC algorithm, the average convergence speed of the algorithm is increased by 47.83%, and the average convergence accuracy of the algorithm is increased by 53.49%.

**Key words:** path cooperative planning; three-dimensional space obstacle; improved artificial bee colony (IABC); particle swarm optimization (PSO); dynamic greedy criterion

**基金项目:** 上海航天科技创新基金(SAST2023-012)。

**收稿日期:** 2024-11-26; **修订日期:** 2025-04-16

**通信作者:** 龚华军,男,教授,博士生导师,E-mail: ghj301@nuaa.edu.cn。

**引用格式:** 马梓元,刘伟鹏,胡春朝,等. 基于改进人工蜂群算法的无人飞行器路径协同规划[J]. 南京航空航天大学学报(自然科学版),2025,57(6):1124-1134. MA Ziyuan, LIU Weipeng, HU Chunchao, et al. Path cooperative planning for unmanned aerial vehicles based on improved artificial bee colony algorithm[J]. Journal of Nanjing University of Aeronautics & Astronautics(Natural Science Edition),2025, 57(6):1124-1134.

随着信息技术以及人工智能技术在航空领域的广泛应用,未来空中无人装备作战场景向集群化、智能化发展<sup>[1-3]</sup>。在未来空战任务中,高亚声速无人飞行器(Unmanned aerial vehicle, UAV)使用频率将大幅增高,而作为指挥官的有人机作战任务处理能力有限,因此高亚声速无人飞行器应当具备一定的自主能力,特别是自主路径规划能力。文献[4]使用合作评估机制与合作个体选取机制对自适应差分多目标进化算法进行改进,引入协同进化的思想,提升了种群的进化质量,提高寻找全局最优的能力。文献[5]对羊群算法进行改进,简化羊群位置更新方式提高算法速度,并使用莱维飞行策略替代牧羊监督机制,对算法进一步进行了简化。文献[6]通过对蝴蝶优化算法进行改进,通过引入对数型自适应惯性权重调整策略、全局动态更新调节策略与动态余弦概率选择策略增加算法的全局与局部探索能力以防止算法陷入局部最优。文献[7]对人工势场算法进行改进,根据导弹航向在障碍物周围生成环流斥力场以解决人工势场算法易陷入局部最优的问题。文献[8]引入人工蜂群(Artificial bee colony, ABC)算法雇佣蜂、观察蜂、侦察蜂的概念对蝙蝠算法进行改进,为了增加算法的局部搜索能力,还在观察蜂阶段通过随机选取3个雇佣蜂的蜜源位置对蜜源进行更新,同时为了压缩收敛速度,在观察蜂阶段只考虑距离损失函数。文献[9]对ABC算法进行改进,在蜜源初始化阶段采用帐篷混沌映射初始化种群以提高种群初始多样性,增加全局搜索能力;在雇佣蜂阶段,引入引导搜索策略,增加搜索的目的性;在跟随蜂和侦察蜂阶段采用大步长探索策略与正切随机进化策略,防止搜索陷入局部最优同时保持种群的活力。

上述算法虽然已经在路径规划问题取得了一定的成果,但多数算法在规划过程中仍存在一些问题:

- (1) 部分算法将路径规划问题简化为二维问题,忽略了空间中地形对路径规划的影响;
- (2) 在目标函数的设计中未考虑飞行过程中其余飞行器路径的影响,不适用于战场集群化的发展趋势;
- (3) 对威胁区或禁飞区的设计过于简单,仅考虑地面上的禁飞区,未考虑处于空中的威胁区或禁飞区。

基于上述问题,提出一种改进人工蜂群(Improved artificial bee colony, IABC)算法以实现高亚声速无人飞行器的路径规划,通过粒子群(Particle swarm optimization, PSO)算法在雇佣蜂阶段引入个体最优与种群最优对探索的影响,增强搜索的

目的性与搜索能力,加速算法收敛速度;在跟随蜂阶段引入局部平滑算法,进一步加快收敛速度。

## 1 问题描述

### 1.1 战场空间建模

对于战场空间,本文使用代表战场范围的立方体左下的下顶点作为观察站,建立站心坐标系<sup>[10]</sup>,坐标系x轴正方向为观测站正东方向,坐标系y轴正方向为过观测站子午线的北方向,坐标系z轴为观测站指向天空的法线方向。除去划定战场范围,进行路径规划时还需考虑在战场中的地形与空中需要规避的禁飞区域。其中,禁飞区与威胁区由随机生成的标准正交立方体组成,而地形可使用数学表达为<sup>[5]</sup>

$$z(x, y) = \sum_{i=1}^n h_{m_i} e^{-\frac{(x - x_{m_i})^2}{a} - \frac{(y - y_{m_i})^2}{b}} \quad (1)$$

式中: $n$ 为地形中山峰的总数; $h_{m_i}$ 为第 $i$ 座山峰的高度; $x_{m_i}$ 为第 $i$ 座山峰中心点的 $x$ 坐标; $y_{m_i}$ 为第 $i$ 座山峰中心点的 $y$ 坐标; $a$ 与 $b$ 分别为山峰 $x$ 方向与 $y$ 方向上的坡度影响因子。

当 $n=50$ ,战场范围划定为东西边长60 km、南北边长为60 km、高度范围为14 km的立方体,禁飞区生成约束为 $x \in [1, 10]$ 、 $y \in [1, 10]$ 、 $z \in [0, 4]$ 时,生成的战场空间如图1所示。

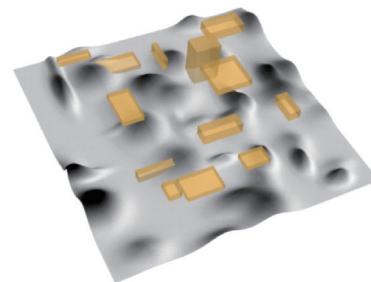


图1 战场空间模型

Fig.1 Battlefield space model

### 1.2 目标函数

路径规划问题是一个多约束问题,在对空战中高亚声速无人飞行器进行路径规划时,需要考虑油量、地形、威胁区、转向能力以及友方飞行器对飞行路线的影响,本文根据这些约束,将算法目标函数分为以下4部分。

#### (1) 路程损失

飞行速度一定时,飞行路程的长短正比于飞行油耗与飞行时长,路程损失的数学表达为

$$J_{\text{len}} = \sum_{i=1}^n l_i \quad (2)$$

式中: $n$ 为路径总段数; $l_i$ 为第 $i$ 段路径的欧氏距离。

### (2) 碰撞损失

在路径规划时,为了防止无人飞行器飞行高度过低与地面产生碰撞,或闯入禁飞区与威胁区,本文根据碰撞情况设计碰撞损失函数为

$$J_{\text{col}} = \sum_{i=1}^n J_{\text{col}_i} \quad (3)$$

式中: $J_{\text{col}_i}$ 为第*i*段路径的碰撞损失,数学表达为

$$J_{\text{col}_i} = \begin{cases} 0 & \text{未碰撞} \\ 1 & \text{碰撞} \end{cases} \quad (4)$$

### (3) 角度损失

飞行路径的曲折程度将极大影响无人飞行器的机动效率,同时受限于无人飞行器的飞行性能,每段路径的夹角应满足其最大转弯能力,角度损失的数学表达为

$$J_{\text{ang}} = \sum_{i=1}^n J_{\text{ang}_i} \quad (5)$$

式中 $J_{\text{ang}_i}$ 为第*i*段路径的角度损失,其数学表达为

$$J_{\text{ang}_i} = \begin{cases} 0 & \alpha_i < \alpha_{\text{max}} \\ 1 & \alpha_i \geq \alpha_{\text{max}} \end{cases} \quad (6)$$

式中: $\alpha_i$ 为第*i*段路径与第*i+1*段路径的夹角; $\alpha_{\text{max}}$ 为无人飞行器的最大转角。

### (4) 路径干涉损失

随着未来空战向集群化发展,空中无人飞行器间的路径规划协同能力将更加重要。因此,在对无人飞行器进行路径规划时,应考虑到在相同时间窗口下所规划的路径是否会与其余飞行器的路径发生干涉。第*i*段路径的时间窗口由开始时间 $T_{i_{\text{start}}}$ 与结束时间 $T_{i_{\text{end}}}$ 组成,数学表达为

$$T_{i_{\text{start}}} = \begin{cases} 0 & i = 0 \\ T_{(i-1)_{\text{end}}} & i \neq 0 \end{cases} \quad (7)$$

$$T_{i_{\text{end}}} = T_{i_{\text{start}}} + \frac{l_i}{v_i} \quad (8)$$

式中 $v_i$ 为第*i*段航路飞行速度。

根据干涉情况,路径干涉损失函数为

$$J_{\text{int}} = \sum_{i=1}^n J_{\text{int}_i} \quad (9)$$

式中 $J_{\text{int}_i}$ 为第*i*段路径在其时间窗口内的碰撞损失,数学表达为

$$J_{\text{int}_i} = \begin{cases} 0 & \text{未干涉} \\ 1 & \text{干涉} \end{cases} \quad (10)$$

### (5) 目标函数

综合考虑以上约束,所提算法的目标函数可表示为

$$J = \omega_{\text{len}} J_{\text{len}} + \omega_{\text{col}} J_{\text{col}} + \omega_{\text{ang}} J_{\text{ang}} + \omega_{\text{int}} J_{\text{int}} \quad (11)$$

式中: $\omega_{\text{len}}$ 、 $\omega_{\text{col}}$ 、 $\omega_{\text{ang}}$ 和 $\omega_{\text{int}}$ 分别为路程损失权重、碰撞损失权重、角度损失权重和路径干涉权重。其

中,本算法 $\omega_{\text{len}}$ 取 0.35、 $\omega_{\text{col}}$ 与 $\omega_{\text{int}}$ 取 6,为增大算法前期探索能力, $\omega_{\text{ang}}$ 采取分段取值,数学表达为

$$\omega_{\text{ang}} = \begin{cases} 0 & \text{iter} \leq \gamma \text{iter}_{\text{max}} \\ 3 & \text{iter} > \gamma \text{iter}_{\text{max}} \end{cases} \quad (12)$$

式中:iter 为当前迭代次数;iter<sub>max</sub> 为最大迭代次数。

## 2 改进人工蜂群算法

ABC 算法是 2005 年由 Karaboga<sup>[11]</sup>提出的一种模仿蜂群相互协作完成采蜜行为的启发式优化算法,该算法模型由蜜源、雇佣蜂与未雇佣蜂 3 部分组成。其中,蜜源代表着需优化问题的可行解,蜜源的花蜜量意味着解的质量。雇佣蜂数量与蜜源一致,每个蜜源有且只被一只雇佣蜂开发,当蜜源的花蜜量为零时,雇佣蜂将转化为未雇佣蜂中的侦查蜂以探索新的蜜源。而未雇佣蜂由两种蜜蜂组成,除去侦察蜂,第二种未雇用蜂为跟随蜂,跟随蜂会根据自身观察到的雇佣蜂分享的蜜源信息,以轮盘赌的形式选择质量最优蜜源进行开发。通过不断地迭代,以寻找全局质量最优的蜜源,进而完成对最优解的探索。

虽然传统 ABC 算法在处理低维优化问题时能够较快地完成收敛,但随着优化问题维度的升高,收敛速度会大幅下降,同时因为算法对邻域的搜索能力较弱,容易陷入局部最优<sup>[12-14]</sup>。为解决这些问题,本文对算法中雇佣蜂阶段与观察蜂阶段进行改进。

### 2.1 雇佣蜂阶段改进

为提升雇佣蜂邻域探索的目的性、加快算法的收敛速度,在雇佣蜂对新蜜源的探索阶段引入粒子群算法<sup>[15]</sup>。粒子群算法通过模拟鸟群捕食行为,在解的更新迭代过程中综合考虑自身惯性、个体最优解与群体最优解,以具备在保留自身信息的前提下还朝着群体最优不断收敛能力。引入粒子群算法后雇佣蜂的探索速度更新与探索位置更新公式为

$$\begin{cases} v_i^{t+1} = \omega v_i^t + c_1 r_1 (p_i^t - x_i^t) + c_2 r_2 (g^t - x_i^t) \\ x_i^{t+1} = f_e(x_i) = x_i^t + v_i^{t+1} \end{cases} \quad (13)$$

式中: $v_i^{t+1}$  与  $v_i^t$  分别为  $t+1$  时刻与  $t$  时刻下第  $i$  只雇佣蜂的探索速度; $x_i^{t+1}$  与  $x_i^t$  分别为  $t+1$  时刻与  $t$  时刻下第  $i$  只雇佣蜂的探索位置; $\omega$  为惯性权重,本算法取 0.9; $c_1$  和  $c_2$  为学习因子,本算法均取 2; $r_1$  和  $r_2$  为  $[0, 1]$  的随机数; $p_i^t$  为  $t$  时刻下第  $i$  只雇佣蜂的最佳探测位置; $g^t$  为从开始到  $t$  时刻蜂群探索到的最佳位置。

## 2.2 观察蜂阶段改进

传统人工蜂群算法虽然在观察蜂阶段通过分别选取两只不同的雇佣蜂作为起点蜜源与探索方向,为邻域内新蜜源的搜索提供了自适应搜索步长,但在迭代初期,由于初始化的随机性,使得路径的高频波动是导致路径损失函数过大的重要原因之一。

为进一步降低算法收敛速度,基于贪婪准则在迭代初期的观察蜂阶段引入了局部平均平滑处理,使用滑动窗口对观察蜂搜索的结果路径的局部进行平均平滑处理,降低路径局部波动,进一步加快收敛速度。

### (1) 局部平均平滑

局部平均平滑处理方式为

$$x'_i = f_s(x_i) = \frac{\sum_{j=\max(0, a)}^{\min(n, b)} x_j}{l} \quad (14)$$

$$a = i - l_{\text{half}} \quad (15)$$

$$b = i + l_{\text{half}} \quad (16)$$

$$l_{\text{half}} = \text{round}\left(\frac{l}{2}\right) \quad (17)$$

式中: $x'_i$ 为平滑后第*i*个路径点坐标;*a*为滑动窗口下索引;*b*为滑动窗口上索引;*l*为滑动窗口包含路径点数,本算法取3;*l<sub>half</sub>*为滑动窗口半径。

### (2) 观察蜂探索方式

由于路径高频波动随着迭代过程对收敛速度造成的影响逐渐降低,故在观察蜂阶段基于动态贪婪准则选择蜜源邻域的探索方式,其数学表达为

$$x'_i = \begin{cases} f_e(x_i) & \varphi \leq \varphi_0 + (1 - \varphi_0) \frac{\text{iter}}{\text{iter}_{\text{max}}} \\ f_s(x_i) & \varphi > \varphi_0 + (1 - \varphi_0) \frac{\text{iter}}{\text{iter}_{\text{max}}} \end{cases} \quad (18)$$

式中: $\varphi$ 为[0,1]的随机数; $\varphi_0$ 为贪婪准则的阈值常量,本算法取0.7。

## 2.3 算法基本流程

综上所述,IABC的算法流程如图2所示。算法具体步骤为:

**Step 1** 对蜜源进行初始化。

**Step 2** 雇佣蜂根据式(13)基于PSO算法对所属蜜源进行邻域搜索,并对搜索到的新蜜源适应度进行计算。

**Step 3** 观察蜂根据新蜜源的适应度基于轮盘赌原则选择自己要跟随的雇佣蜂。

**Step 4** 观察蜂根据式(18)对跟随的雇佣蜂所属蜜源进行进一步的邻域搜索,并对搜索到的新蜜源的适应度进行计算。

**Step 5** 将雇佣蜂与观察蜂搜索到的新蜜源

的适应度与旧蜜源进行比较,使用质量更优的新蜜源替代旧蜜源,并对全局最优蜜源进行更新。

**Step 6** 对未被替代的旧蜜源开采度进行更新,同时判断其开采度是否大于阈值,若大于阈值则派出侦察蜂开发新的蜜源,代替旧蜜源。

**Step 7** 对迭代次数进行判断,若小于阈值则进入Step 2进行新一轮迭代。若大于阈值,则将当前全局最优蜜源作为规划结果输出。

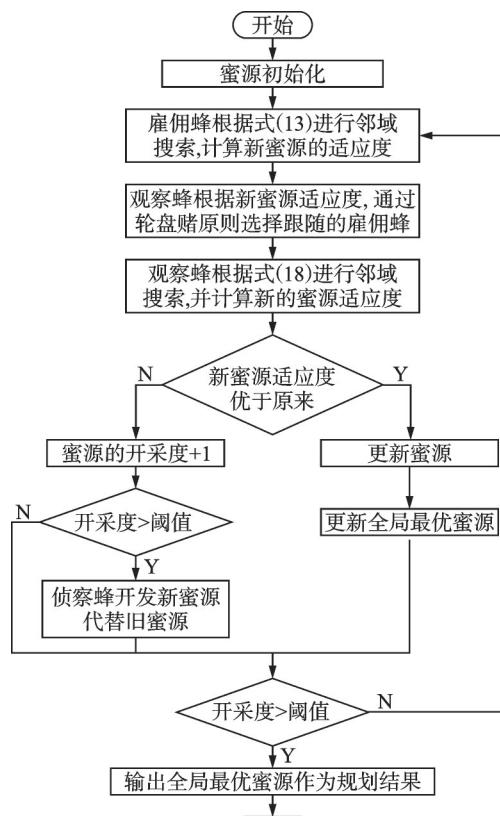


图2 IABC算法流程图

Fig.2 Flow chart of IABC algorithm

## 3 仿真实验

为验证算法的有效性,通过控制变量法控制不同路径点生成数量与不同待规划高亚声速无人飞行器数量,对所提IABC算法与传统ABC算法、PSO算法、基于PSO改进的ABC-PSO算法进行对比仿真实验。

实验中,高亚声速无人飞行器的性能参数如表1所示。

表1 高亚声速无人飞行器性能参数

Table 1 Performance parameters of high subsonic UAV

名称	取值
最大速度/ <i>Ma</i>	0.7
最大航程/km	250
相邻两段航路最大夹角/(°)	60

### 3.1 路径点生成数量不同的对比实验

参考文献[4]与文献[6]的作战场景较为简单,其路径点数量均取值在 10 以下。但经过验证,路径点数量在 10 以下时,无法满足本算法应用的当前作战场景需求,而路径点数量在 20 以上时,在本算法应用场景中会在降低算法效率的同时增加路径的曲折程度。故本算法在以路径点生成数量为变量的对比仿真实验中,路径点数量选择 10、15、20 进行比较,具体固定参数如表 2 所示。

表 2 路径点对比仿真实验固定参数

Table 2 Fixed parameters of comparative simulation experiments for path points

名称	取值
战场范围	$x \in [0, 60], y \in [0, 60], z \in [0, 14]$
起点 ( $x, y, z$ )	$[[0, 5, 7], [2, 0, 6]]$
终点 ( $x, y, z$ )	$[[50, 40, 2], [40, 55, 3]]$
禁飞区坐标 ( $x, y, z$ )	$[(24.34, 49.16, 9.64), (45.67, 10.87, 9.61), (10.95, 37.24, 7.15), (38.09, 38.74, 0.0), (40.4, 20.75, 5.06), (43.29, 49.05, 8.34), (29.21, 10.97, 4.79), (7.96, 9.06, 5.81), (32.3, 45.18, 2.63), (24.16, 38.72, 7.1), (17.1, 24.66, 9.55), (17.51, 15.36, 0.0), (32.69, 5.87, 0.0)]$
禁飞区范围 ( $\Delta x, \Delta y, \Delta z$ )	$[(5.22, 3.53, 1.24), (7.08, 1.7, 1.3), (2.69, 5.79, 2.2), (8.42, 1.91, 1.92), (4.93, 8.74, 1.29), (2.6, 3.46, 1.57), (1.25, 6.1, 2.66), (8.99, 4.6, 2.52), (8.18, 5.29, 1.33), (8.83, 3.12, 2.52), (8.27, 8.1, 1.7), (6.13, 4.43, 10), (9.67, 4.61, 1.42)]$
最大迭代次数	200
UAV 数量	2

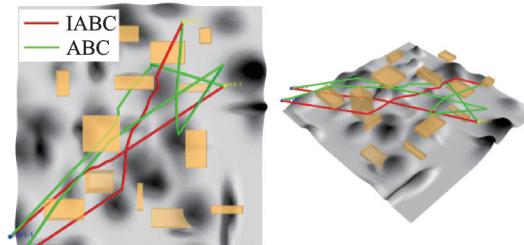
#### 3.1.1 路径点数量为 10 的对比实验

当路径点生成数量设为 10 时仿真结果如图 3 所示,各算法目标函数的迭代情况如图 4 所示,IABC 算法目标函数的各类损失值迭代情况如图 5 所示。

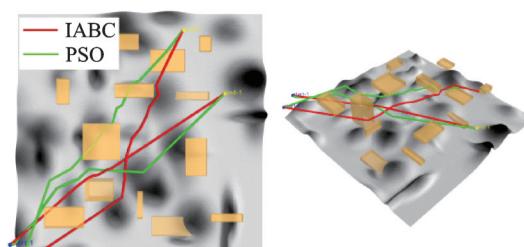
各算法具体的收敛情况如表 3 所示,各算法在收敛时碰撞损失、角度损失、路程干涉损失均为 0。相较 ABC 算法,IABC 算法收敛速度提升 31.96%,收敛精度提升 26.59%,规划结果路径距离缩短 26.59%。

#### 3.1.2 路径点数量为 15 的对比实验

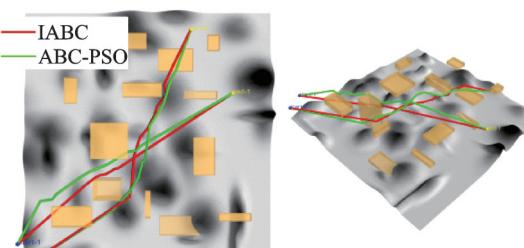
当路径点生成数量设为 15 时仿真结果如图 6 所示,各算法目标函数的迭代情况如图 7 所示,IABC 算法目标函数的各类损失值迭代情况如图 8 所示。



(a) Comparison diagram of simulation results between IABC and ABC algorithms



(b) Comparison diagram of simulation results between IABC and PSO algorithms



(c) Comparison diagram of simulation results between IABC and ABC-PSO algorithms

Fig.3 Comparison diagrams of simulation results when the number of path generation is 10

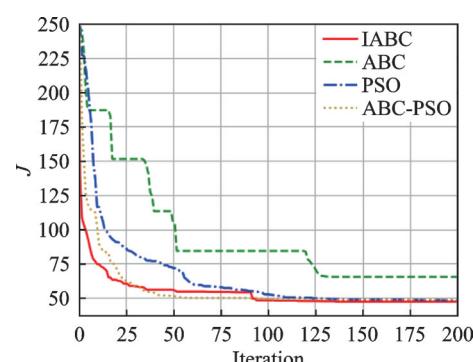


图 4 路径生成数量为 10 时目标函数迭代对比

Fig.4 Comparison diagram of objective functions iterations when the number of path generation is 10

各算法具体的收敛情况如表 4 所示,除碰撞损失外,各算法在收敛时角度损失、路程干涉损失均为 0。相较 ABC 算法,IABC 算法收敛速度提升 40.23%,收敛精度提升 54.6%,规划结果路径距离缩短 51.95%。

#### 3.1.3 路径点数量为 20 的对比实验

当路径点生产数量设为 20 时仿真结果如图 9

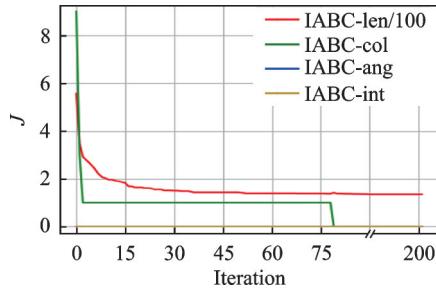
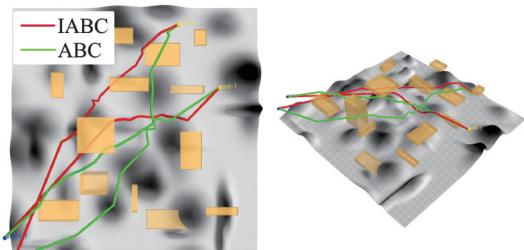


图5 路径生成数量为10时IABC算法目标函数各损失值  
Fig.5 Iteration diagram of each loss value of the objective function of the IABC algorithm when the number of path generations is 10

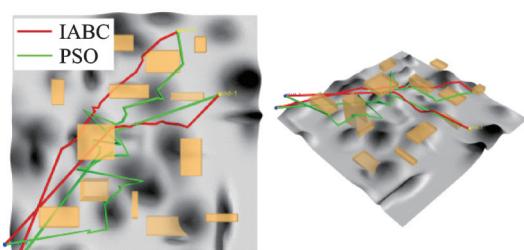
表3 路径生成数量为10时收敛情况对比

Table 3 Comparison of convergence situations when the number of path generation is 10

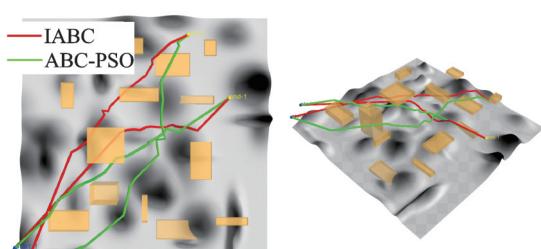
算法名称	ABC	PSO	ABC-PSO	IABC
收敛速度/s	25.53	17.56	26.68	17.37
收敛值	65.24	48.19	48.63	47.89
碰撞次数	0	0	0	0
路径距离/km	186.40	137.68	138.94	136.84
直线距离/km			128.16	



(a) Comparison diagram of simulation results between IABC and ABC algorithms



(b) Comparison diagram of simulation results between IABC and PSO algorithms



(c) Comparison diagram of simulation results between IABC and ABC-PSO algorithms

图6 路径生成数量为15时仿真结果对比

Fig.6 Comparison diagrams of simulation results when the number of path generation is 15

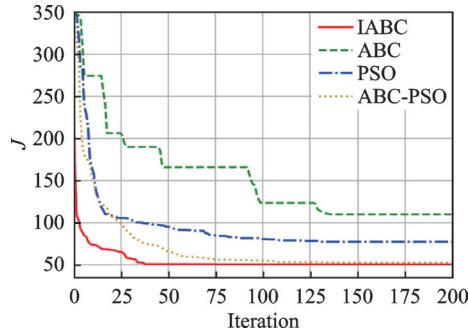


图7 路径生成数量为15时目标函数迭代对比

Fig.7 Comparison diagram of objective function iterations when the number of path generation is 15

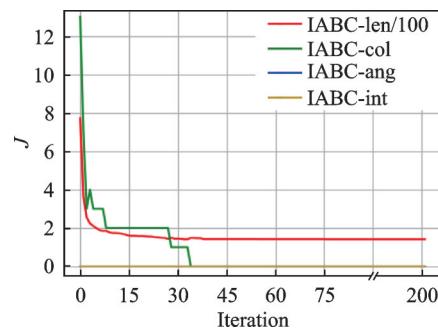


图8 路径生成数量为15时IABC算法目标函数各损失值迭代  
Fig.8 Iteration diagram of each loss value of the objective function of the IABC algorithm when the number of path generations is 15

表4 路径生成数量为15时收敛情况对比

Table 4 Comparison of convergence situations when the number of path generation is 15

算法名称	ABC	PSO	ABC-PSO	IABC
收敛速度/s	41.16	16.68	41.49	24.60
收敛值	108.98	76.39	51.64	49.48
碰撞次数	1	0	0	0
路径距离/km	294.24	218.25	147.55	141.37
直线距离/km			128.16	

所示,各算法目标函数的迭代情况如图10所示,IABC算法的各类损失值迭代情况如图11所示。

各算法具体的收敛情况如表5所示,除碰撞损失外,各算法在收敛时角度损失、路程干涉损失均为0。相较ABC算法,IABC算法收敛速度提升47.23%,收敛精度提升63.24%,规划结果路径距离缩短61.64%。

### 3.1.4 路径点生成数量不同的对比实验结果分析

由3组以路径点生成数量为变量的对比实验可知,上述几种方法,除ABC算法在路径点生成数量较多时规避碰撞能力下降外,其余算法在收敛后均能有效规避碰撞、路径干涉与角度损失。相较ABC算法,IABC算法收敛速度平均提升39.81%,收敛精度平均提升48.14%,规划结果路径距离平

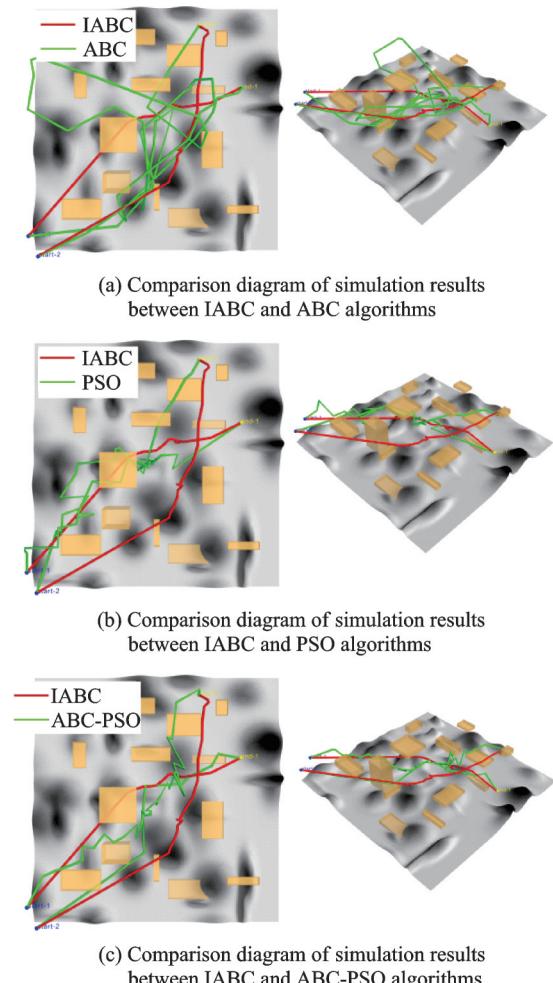


图 9 路径生成数量为 20 时仿真结果对比

Fig.9 Comparison diagrams of simulation results when the number of path generation is 20

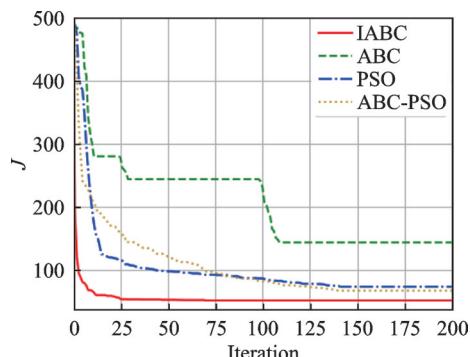


图 10 路径生成数量为 20 时目标函数迭代对比

Fig.10 Comparison diagram of objective function iterations when the number of path generation is 20

均缩短 46.73%，且提升效果随着路径点生成数量的增加而增加。

### 3.2 待规划高亚声速无人飞行器数量不同的对比实验

本文研究方向主要着重于 2v2、4v4 小规模的作战场景,故在以待规划高亚声速无人飞行器数量为变量的对比仿真实验中无人机数量取 2、3、4 进行比较,具体固定参数如表 6 所示。

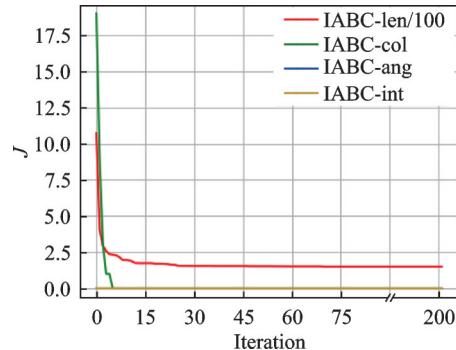


图 11 路径生成数量为 20 时 IABC 算法目标函数各损失值迭代

Fig.11 Iteration diagram of each loss value of the objective function of the IABC algorithm when the number of path generation is 20

表 5 路径生成数量为 20 时收敛情况对比

Table 5 Comparison of convergence situations when the number of path generation is 20

算法名称	ABC	PSO	ABC-PSO	IABC
收敛速度/s	50.33	30.00	66.68	26.56
收敛值	144.22	74.08	67.77	53.02
碰撞次数	1	0	0	0
路径距离/km	394.9	211.65	193.63	151.50
直线距离/km				128.16

表 6 无人飞行器数量对比仿真实验固定参数

Table 6 Fixed parameters of comparative simulation experiments for the number of UAV

名称	取值
战场范围	$x \in [0, 60], y \in [0, 60], z \in [0, 14]$
禁飞区坐标 $(x, y, z)$	$[(24.34, 49.16, 9.64), (45.67, 10.87, 9.61), (10.95, 37.24, 7.15), (38.09, 38.74, 0.0), (40.4, 20.75, 5.06), (43.29, 49.05, 8.34), (29.21, 10.97, 4.79), (7.96, 9.06, 5.81), (32.3, 45.18, 2.63), (24.16, 38.72, 7.1), (17.1, 24.66, 9.55), (17.51, 15.36, 0.0), (32.69, 5.87, 0.0)]$
禁飞区范围 $(\Delta x, \Delta y, \Delta z)$	$[(5.22, 3.53, 1.24), (7.08, 1.7, 1.3), (2.69, 5.79, 2.2), (8.42, 1.91, 1.92), (4.93, 8.74, 1.29), (2.6, 3.46, 1.57), (1.25, 6.1, 2.66), (8.99, 4.6, 2.52), (8.18, 5.29, 1.33), (8.83, 3.12, 2.52), (8.27, 8.1, 1.7), (6.13, 4.43, 10), (9.67, 4.61, 1.42)]$
最大迭代次数	200
航路点数量	15

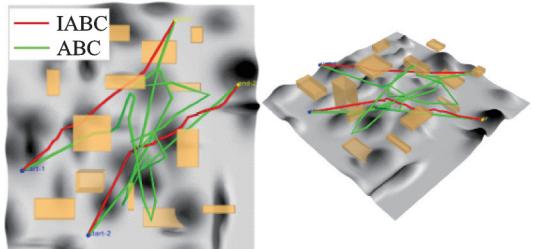
#### 3.2.1 UAV 数量为 2 的对比实验

实验变量参数如表 7 所示,仿真结果如图 12 所示,各算法目标函数的迭代情况如图 13 所示, IABC 算法目标函数的各类损失值迭代情况如图 14 所示。

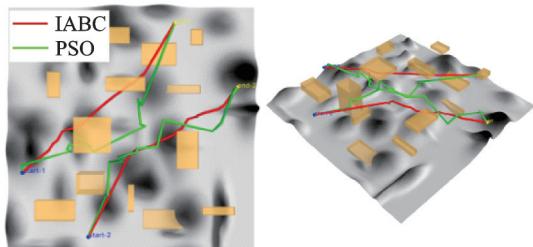
表7 无人飞行器数量为2时变量参数

Table 7 Variable parameters when the number of UAVs is 2

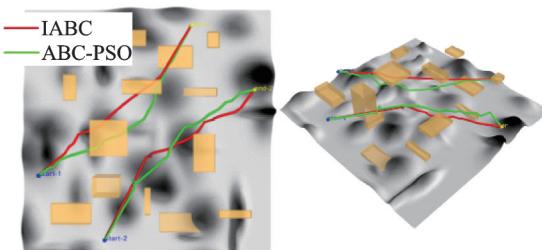
名称	取值
起点( $x, y, z$ )	$[[5, 20, 7], [20, 5, 6]]$
终点( $x, y, z$ )	$[[40, 55, 4], [55, 40, 3]]$
UAV数量	2



(a) Comparison diagram of simulation results between IABC and ABC algorithms



(b) Comparison diagram of simulation results between IABC and PSO algorithms



(c) Comparison diagram of simulation results between IABC and ABC-PSO algorithms

图12 无人飞行器数量为2时仿真结果对比

Fig.12 Comparison diagrams of simulation results when the number of UAVs is 2

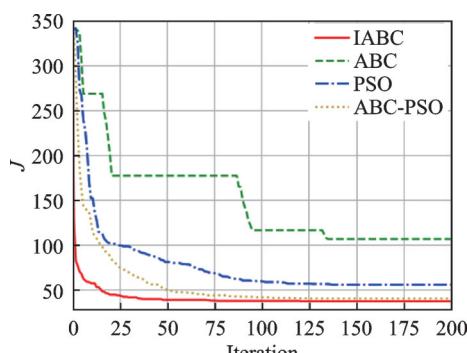


图13 无人飞行器数量为2时目标函数迭代对比

Fig.13 Comparison diagram of objective function iterations when the number of UAVs is 2

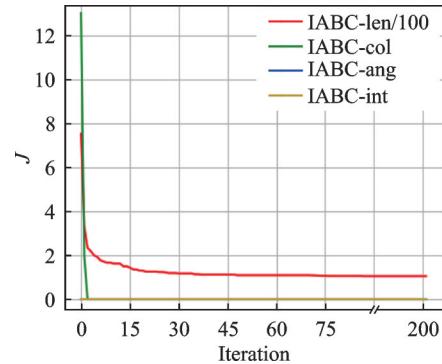


图14 无人飞行器数量为2时IABC算法目标函数各损失值

Fig.14 Iteration diagram of each loss value of the objective function of the IABC algorithm when the number of UAVs is 2

各算法具体的收敛情况如表8所示,除碰撞损失外,各算法在收敛时角度损失、路程干涉损失均为0。相较ABC算法,IABC算法收敛速度提升66.61%,收敛精度提升64.00%,规划结果路径距离缩短61.85%。

表8 无人飞行器数量为2时收敛情况对比

Table 8 Comparison of convergence situations when the number of UAVs is 2

算法名称	ABC	PSO	ABC-PSO	IABC
收敛速度/s	41.09	18.14	33.33	13.72
收敛值	105.98	55.04	39.54	38.15
碰撞次数	1	0	0	0
路径距离/km	285.66	157.25	112.97	108.99
直线距离/km				99.18

### 3.2.2 UAV数量为3的对比实验

实验变量参数如表9所示,仿真结果如图15所示,各算法目标函数的迭代情况如图16所示,IABC算法目标函数的各类损失值迭代情况如图17所示。

表9 无人飞行器数量为3时变量参数

Table 9 Variable parameters when the number of UAVs is 3

名称	取值
起点( $x, y, z$ )	$[[5, 20, 7], [5, 5, 5], [20, 5, 6]]$
终点( $x, y, z$ )	$[[40, 55, 4], [40, 40, 5], [55, 40, 3]]$
UAV数量	3

各算法具体的收敛情况如表10所示,各算法在收敛时碰撞损失、角度损失、路程干涉损失均为0。相较ABC算法,IABC算法收敛速度提升53.31%、收敛精度提升55.65%,规划结果路径距离缩短55.65%。

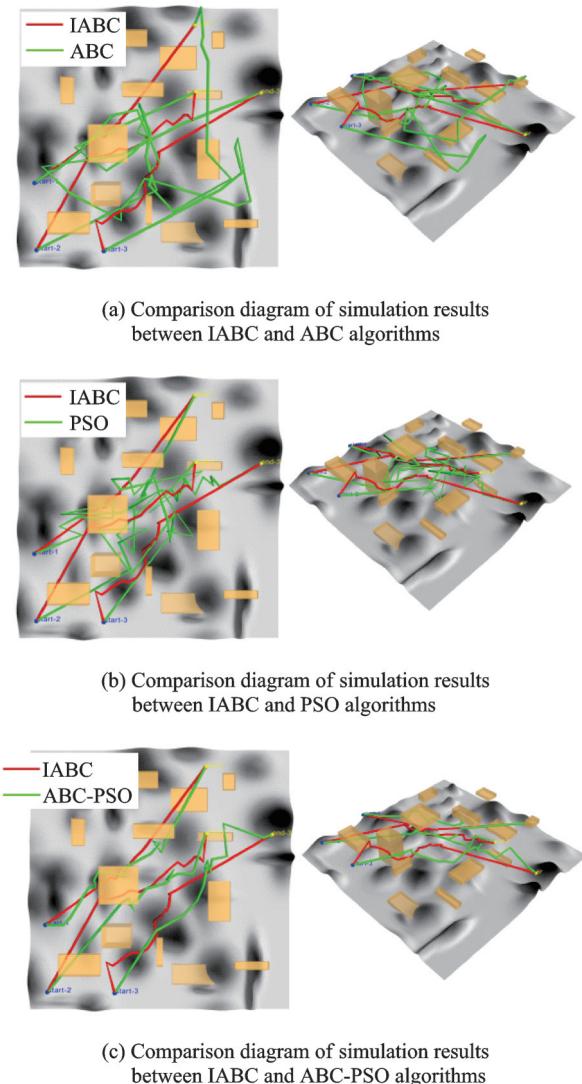


图 15 无人飞行器数量为 3 时仿真结果对比

Fig.15 Comparison diagrams of simulation results when the number of UAVs is 3

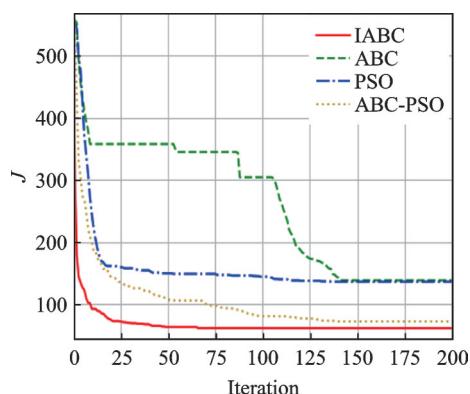


图 16 无人飞行器数量为 3 时目标函数迭代对比

Fig.16 Comparison diagram of objective function iterations when the number of UAVs is 3

### 3.2.3 UAV 数量为 4 的对比实验

实验变量参数如表 11 所示, 仿真结果如图 18 所示, 各算法目标函数的迭代情况如图 19 所示, IABC 算法目标函数的各类损失值迭代情况如图

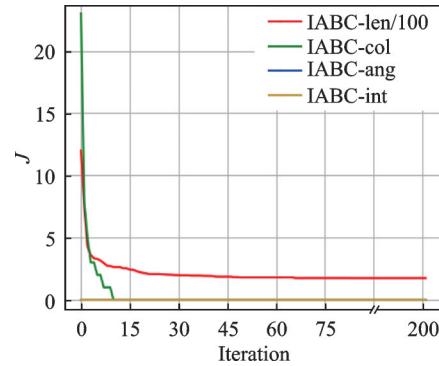


图 17 无人飞行器数量为 3 时 IABC 算法目标函数各损失值

Fig.17 Iteration diagram of each loss value of the objective function of the IABC algorithm when the number of UAVs is 3

表 10 无人飞行器数量为 3 时收敛情况对比

Table 10 Comparison of convergence situations when the number of UAVs is 3

算法名称	ABC	PSO	ABC-PSO	IABC
收敛速度/s	100.25	41.01	103.49	46.81
收敛值	137.94	135.92	71.83	61.18
碰撞次数	0	0	0	0
路径距离/km	394.12	388.36	205.23	174.81
直线距离/km			148.67	

表 11 无人飞行器数量为 4 时变量参数

Table 11 Variable parameters when the number of UAVs is 4

名称	取值
起点( $x, y, z$ )	$[[5, 20, 7], [5, 10, 5], [10, 5, 5], [20, 5, 6]]$
终点( $x, y, z$ )	$[[40, 55, 4], [40, 50, 5], [50, 40, 5], [55, 40, 3]]$
UAV 数量	4

20 所示。各算法的收敛情况如表 12 所示, 各算法在收敛时碰撞损失、角度损失、路程干涉损失均为 0。相较 ABC 算法, IABC 算法收敛速度提升 47.65%, 收敛精度提升 56.86%, 规划结果路径距离缩短 56.86%。

#### 4.2.4 待规划高亚声速无人飞行器数量不同的对比实验结果分析

由 3 组以待规划高亚声速无人飞行器数量及其起终点为变量的对比实验可知, 上述几种方法, 除 ABC 算法规避碰撞的能力较弱外, 其余算法都能有效规避碰撞、路径干涉与角度损失。相较 ABC 算法, IABC 算法收敛速度平均提升 55.86%, 收敛精度平均提升 58.84%, 规划结果路径距离平均缩短 58.12%。

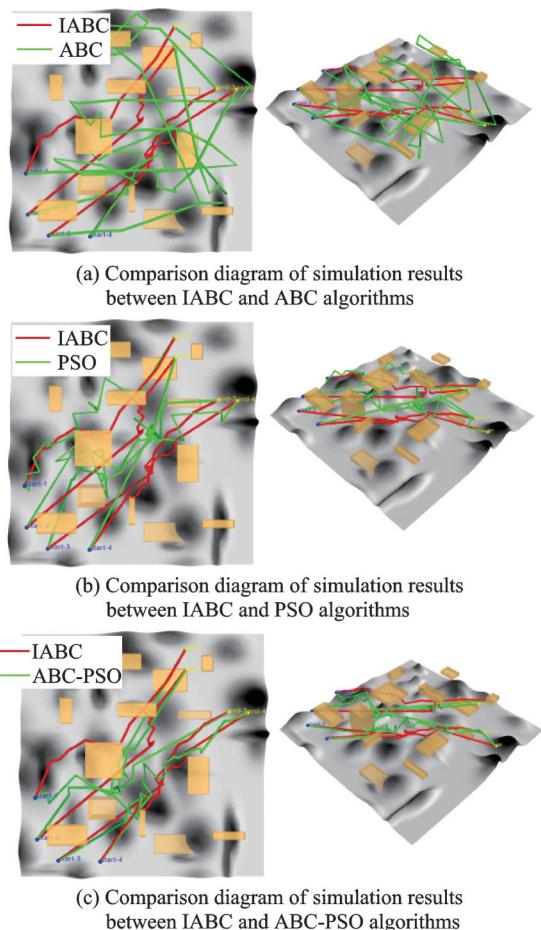


图18 无人飞行器数量为4时仿真结果对比

Fig.18 Comparison diagrams of simulation results when the number of UAVs is 4

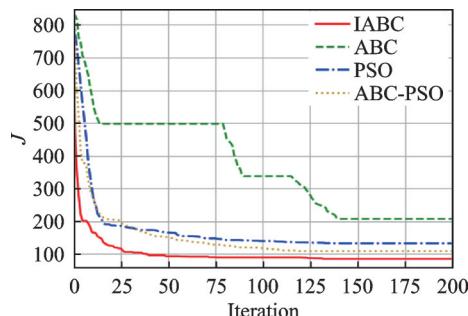


图19 无人飞行器数量为4时目标函数迭代对比

Fig.19 Comparison diagram of objective function iterations when the number of UAVs is 4

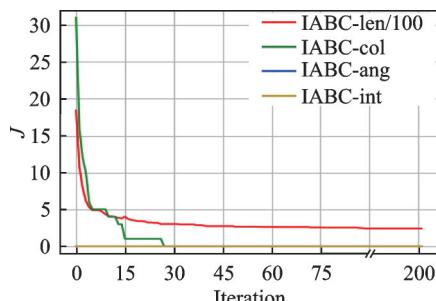
图20 无人飞行器数量为4时IABC算法目标函数各损失值  
Fig.20 Iteration diagram of each loss value of the objective function of the IABC algorithm when the number of UAVs is 4

表12 无人飞行器数量为4时收敛情况对比

Table 12 Comparison of convergence situations when the number of UAVs is 4

名称	ABC	PSO	ABC-PSO	IABC
收敛速度/s	170.82	78.63	147.41	89.42
收敛值	207.24	132.45	108.80	89.4
碰撞次数	0	0	0	0
路径距离/km	592.11	378.43	310.85	255.43
直线距离/km				205.48

## 4 结 论

本文重点讨论空战场景下的高亚声速无人飞行器的路径协同规划问题,首先,在目标函数中综合考虑了油量、三维空间障碍、转弯能力与友机路径干涉,对路径协同规划问题进行约束;其次,在雇佣蜂阶段引入了PSO算法,增强了算法搜索的目的性、提高了算法的收敛速度;最后,在观察蜂阶段基于动态贪婪准则对迭代初期的蜜源搜索结果使用局部平均平滑算法,进一步提高算法的收敛速度。

通过仿真对比实验可以看出,IABC算法继承了ABC与PSO算法的搜索能力,并且无论是在收敛速度还是收敛精度都得到了显著的提升。相较ABC算法,IABC算法收敛速度平均提升47.83%,收敛精度平均提升53.49%,规划结果路径距离平均缩短52.42%。

目前的研究只针对小规模集群,未来将在大规模集群中做进一步的研究,以适应更加高动态、高复杂的作战场景。

## 参考文献:

- [1] 梁晓龙,胡利平,张佳强,等.航空集群自主空战研究进展[J].科技导报,2020,38(15): 74-88.  
LIANG Xiaolong, HU Liping, ZHANG Jiaqiang, et al. Research progress of autonomous air combat for aerial swarm[J]. Science & Technology Review, 2020, 38(15): 74-88.
- [2] 范晋祥,陈晶华.未来空战新概念及其实现挑战[J].航空兵器,2020,27(2): 15-24.  
FAN Jinxiang, CHEN Jinghua. New concepts of future air warfare and the challenges for its realization [J]. Aero Weaponry, 2020, 27(2): 15-24.
- [3] 孙智孝,杨晟琦,朴海音,等.未来智能空战发展综述[J].航空学报,2021,42(8): 35-49.  
SUN Zhixiao, YANG Shengqi, PIAO Haiyin, et al. A survey of air combat artificial intelligence[J]. Acta Aeronautica et Astronautica Sinica, 2021, 42(8): 35-49.

- [4] 管祥民, 吕人力. 基于协同进化的复杂低空下多飞行器协同航迹规划方法[J]. 南京航空航天大学学报, 2017, 49(S1): 89-95.  
GUAN Xiangmin, LÜ Renli. Complex low-altitude airspace multi-aircraft cooperation path planning method based on cooperative coevolution ary[J]. Journal of Nanjing University of Aeronautics & Astronautics, 2017, 49(S1): 89-95.
- [5] YANG Liuqing, WANG Pengfei, ZHANG Yong. Coordinated path planning for UAVs based on sheep optimization[J]. Transactions of Nanjing University of Aeronautics & Astronautics, 2020, 37(5): 816-830.
- [6] 丁敏, 夏兴宇, 邹永杰, 等. 基于改进蝴蝶优化算法的无人机 3-D 航迹规划方法[J]. 南京航空航天大学学报, 2023, 55(5): 851-858.  
DING Min, XIA Xingyu, ZOU Yongjie, et al. 3-D track planning method of UAV based on improved butterfly optimization algorithm[J]. Journal of Nanjing University of Aeronautics & Astronautics, 2023, 55 (5): 851-858.
- [7] 卢发兴, 戴秋洋, 许俊飞, 等. 基于环流斥力势场的改进 APF 导弹路径规划[J]. 航空动力学报, 2023, 38(9): 2288-2298.  
LU Faxing, DAI Qiuyang, XU Junfei, et al. Improved APF missile route planning based on circulation repulsion potential field[J]. Journal of Aerospace Power, 2023, 38(9): 2288-2298.
- [8] ZHOU X, GAO F, FANG X, et al. Improved bat algorihm for UAV path planning in three-dimensional space[J]. IEEE Access, 2021, 9: 20100-20116.
- [9] LIN S, LI F, LI X, et al. Improved artificial bee colony algorithm based on multi-strategy synthesis for UAV path planning[J]. IEEE Access, 2022, 10: 119269-119282.
- [10] 孙旭民. 矢量闭合差在空间直角坐标系和站心坐标系下的转换[J]. 物探装备, 2017, 27(1): 48-49.  
SUN Xumin. Transformation of vector closed difference in space rectangular coordinates and topocentric coordinate system [J]. Equipment for Geophysical Prospecting, 2017, 27(1): 48-49.
- [11] KARABOGA D. An idea based on honey bee swarm for numerical optimization[R]. Kaysen: Erciyes University, 2005.
- [12] KARABOGA D, BASTURK B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm[J]. Journal of global optimization, 2007, 39: 459-471.
- [13] CUI Y, HU W, RAHMANI A. Fractional-order artificial bee colony algorithm with application in robot path planning[J]. European Journal of Operational Research, 2023, 306(1): 47-64.
- [14] LI G, LIU C, WU L, et al. A mixing algorithm of ACO and ABC for solving path planning of mobile robot[J]. Applied Soft Computing, 2023, 148: 110868.
- [15] KENNEDY J, EBERHART R. Particle swarm optimization[C]//Proceedings of International Conference on Neural Networks. [S.l.]: IEEE, 1995: 1942-1948.

(编辑:夏道家)