2025 年 第 25 卷 第 13 期
2025, 25(13):05501-14

科 学 技 术 与 工 程
Science Technology and Engineering

ISSN 1671—1815
CN 11—4688/T

DOI:10.12404/j.issn.1671-1815.2404351
引用格式:巫光福, 王小林. 求解无人机三维路径规划问题的动态多子群樽海鞘群算法[J]. 科学技术与工程, 2025, 25(13): 5501-5514.
Wu Guangfu, Wang Xiaolin. Dynamic multi-subswarm salp swarm algorithm for solving unmanned aerial vehicles three-dimensional path planning problem[J]. Science Technology and Engineering, 2025, 25(13): 5501-5514.

求解无人机三维路径规划问题的动态多子群樽海鞘群算法

巫光福, 王小林
(江西理工大学信息工程学院, 赣州 341000)

摘 要 无人机三维路径规划问题是在复杂三维环境中找到起点与终点之间最优路径的组合优化问题,但大多数路径规划算法难以在可接受的时间和精度范围内找到可行路径,因此提出了一种基于 *K*-means + + 聚类优化的动态多子群樽海鞘群算法用于解决上述问题。首先,在三维环境模型中结合高度成本提出新的成本函数,将路径规划问题转化为多维函数优化问题。其次,采用 *K*-means + + 聚类算法对种群进行分群,并设计动态多子群机制均衡算法的全局搜索与局部开发;各子群结合多策略协同改进,在避免算法陷入局部最优的同时提高全局寻优能力。最后,在 12 个 CEC2017 基准测试函数中验证了该算法对比其他 5 种算法(ISSA、MSNSSA、IBSO、MBFPA、SSA)的性能后,将其应用于三维环境中对最优路径规划问题进行求解。在不同的环境模型下的仿真实验结果表明,该算法的平均有效路径率相较于其他 5 种算法分别提高了 15.5%、11%、23%、20.5% 和 18%,这证实了该算法在复杂环境下具有优秀的寻优能力。

关键词 三维路径规划; 成本函数; 樽海鞘群算法; *K*-means + + 聚类算法; 动态多子群; 协同改进
中图法分类号 TP301.6 V279; **文献标志码** A

Dynamic Multi-subswarm Salp Swarm Algorithm for Solving Unmanned Aerial Vehicles Three-dimensional Path Planning Problem

WU Guang-fu, WANG Xiao-lin
(School of Information Engineering, Jiangxi University of Science & Technology, Ganzhou 341000, China)

[**Abstract**] The Unmanned aerial vehicles three-dimensional path planning problem is a combinatorial optimization problem to find the optimal path between the starting point and the endpoint in complex three-dimensional environment, but most path planning algorithms struggle to find feasible paths within acceptable time and precision range, therefore, a dynamic multi-subswarm salp swarm algorithm based on *K*-means + + clustering optimization was proposed to address the aforementioned issue. Firstly, a new cost function incorporating height cost was proposed within the three-dimensional environment model. The path planning problem was converted into a multi-dimensional function optimization issue. Secondly, the population was clustered using the *K*-means + + clustering algorithm, and a dynamic multi-subswarm mechanism was designed to balance the algorithm's global search and local exploitation. Each subswarm collaborates with multiple strategies for improvement, avoiding the algorithm from being trapped in local optima while enhancing global optimization capability. Finally, after validating the algorithm against five algorithms ISSA, MSNSSA, IBSO, MBFPA, and SSA using 12 CEC2017 benchmark test functions, it was applied to solve the optimal path planning problem in three-dimensional environments. Simulation results under different environmental models demonstrate that the algorithm's average effective path rate is increased by 15.5%, 11%, 23%, 20.5% and 18% compared to the other five algorithms, confirming its excellent optimization capability in complex environments.

[**Keywords**] three-dimensional path planning; cost function; salp swarm algorithm; *K*-means + + clustering algorithm; dynamic multi-subswarm; collaborative improvement

收稿日期: 2024-06-12 **修订日期**: 2025-02-05
基金项目: 国家自然科学基金(11461031);江西省科技厅重点项目(20212ACB202003);密码科学技术国家重点实验室开放课题面上项目(MMKFKT202123)
第一作者: 巫光福(1977—),男,汉族,江西玉山人,博士,副教授,硕士研究生导师。研究方向:信息论与编码、密码学、信息安全、区块链等。E-mail:wuguangfu@126.com。

近年来,无人机(unmanned aerial vehicles,UAV)在各个领域、行业和应用中越来越受欢迎[1],由于其通用性强、机动性好、安装方便和运行成本适中等特点,无人机被广泛应用于军事、灾害管理、救援行动、公共服务、农业等各种领域[2]。由于无人机的特殊工作属性,在执行任务时将面临大跨度空间和复杂的飞行环境的问题,并且在遇到不同类型的威胁区域和障碍物时,也会在一定程度上增加任务的成本和飞行风险。因此,合理的路径规划不仅可以提高无人机的避障性能,还可以降低飞行成本,满足实际任务需求。无人机路径规划是指在飞行路径、飞行持续时间、燃料消耗和威胁数量或类型等一个或者多个约束条件的前提下,无人机在有障碍物的空间中找到起点到终点的无碰撞最优路径[3]。

针对路径规划提出一系列算法,这些算法通常被区分为经典、启发式、机器学习和元启发式算法。经典算法的代表有 Voronoi 图(VD)[4]、快速探索随机树(rapidly-exploring random tree,RRT)[5]和人工势场(artificial potential field,APF)[6]等。这些算法在简单障碍条件下以较少的计算资源就可以提供较好的路径规划结果,但是在复杂的环境条件下,它们很难规划出可行路径。启发式算法的代表有 A*[7]、贪婪启发式(greedy heuristics,GH)[8]等及其改进算法。与经典算法相比,启发式算法能在合理的时间内提供更优的路径,被认为是部分智能的,在复杂的环境条件和约束下,这些算法也很难获得可靠的路径。采用神经网络(neural network,NN)和强化学习(reinforcement learning,RL)等技术将机器学习引入路径规划中,这些方法已被用于静态和动态环境下以及在各种约束条件下获得无人机的最优飞行路径。然而,机器学习方法却常常受限于高昂的计算成本、大型数据集和大量的训练时间。元启发式算法因其快速收敛和在静态和动态环境条件下出色的寻优能力而被广泛应用于无人机的最优路径规划中,其中元启发式群智能算法更是备受国内外学者的瞩目,如粒子群算法(particle swarm optimization, PSO)[9]、灰狼优化算法(grey wolf optimization, GWO)[10]、蝴蝶优化算法(butterfly optimization algorithm, BOA)[11]等都属于这一类别。

樽海鞘群算法(salp swarm algorithm,SSA)是一种由 Mirjalili 等[12]于 2017 年提出的新型群智能算法。相比其他群体智能算法,SSA 的优点在于参数少、结构简单且易于实现。SSA 在解决大多数优化问题方面都表现出卓越的性能,但与同类群智能算法类似,在解决高维复杂优化问题时仍存在易陷入局部最优、全局搜索与局部开发不平衡等不足。为此,文献[13]提出一种基于 Levy 飞行和正余弦算子的改进樽海鞘群算法,结合改进的正余弦算子对领导者位置进行更新并在追随者位置引入 levy 飞行机制增强算法鲁棒性和稳定性。文献[14]将局部搜索算法(local search algorithm,LSA)和 SSA 结合使用,提高算法跳出局部最优能力。文献[15]将高斯随机行走集成到算法中平衡算法的开发与搜索能力,然后在局部最优停滞点引入一种双领导者再分散策略提高算法搜索效率最后运用于电动汽车的动态充电。文献[16]结合 levy 飞行策略更新领导者位置,并将其应用到多阈值图像分割,获得更高的分割效率和更精确的分割阈值。针对 SSA 在求解效率和寻优精度等方面取得一些成效,但 SSA 在解决三维路径规划问题时仍存在寻优能力差、求解效率低、易收敛于次优路径等问题。现提出一种基于*K*-means++聚类优化的动态多子群樽海鞘群算法(dynamic multi-subswarm salp swarm algorithm based on *K*-means++ clustering optimization,DMSSSA)用于解决上述问题。通过建立多约束三维环境模型,结合高度成本,提出一种三维路径规划成本函数,并采用三次 B 样条曲线插值技术处理路径坐标点,以保证路径的可飞性和平滑性。提出一种基于*K*-means++聚类优化的动态多子群樽海鞘群算法(DMSSSA),在 12 个 CEC2017 基准测试函数中验证DMSSSA对比其他 5 种对比算法的性能后,用 Friedman 检验检测其差异性。为验证 DMSSSA 在解决三维路径规划问题时的有效性,使 DMSSSA 与其他 5 种对比算法在三组不同复杂性的环境模型中进行仿真实验。

1 无人机路径规划中的数学模型

1.1 路径规划问题

路径规划问题主要包括两个部分:路径规划算法和路径规划模型。路径规划算法包括经典算法、启发式算法、机器学习算法和元启发式算法。路径规划模型涵盖了内外部约束,内部约束包括无人机的性能约束,例如飞行高度、飞行距离和飞行转角等;外部约束则包括地形和威胁约束,例如地理范围、地形条件、威胁位置和威胁级别等。所采用目标函数是通过无人机的性能约束和外部飞行环境约束来考虑可能影响路径质量的一些必要因素,路径规划算法结合目标函数获取路径坐标点,并通过三次 B 样条曲线插值技术平滑化坐标点曲线,最后形成无人机可飞行的路径。

1.2 三维无人机飞行环境建模

1.2.1 地形建模

采用复合函数对无人机飞行环境的地形进行

模拟,考虑了原始地形和障碍物区域等因素,形成三维数字高程地图模型,表达式为

$$Z_1(x,y) = \sin(y+a) + b\sin x + c\cos(d\sqrt{x^2+y^2}) + e\cos y + f\sin(f\sqrt{x^2+y^2}) + g\cos y \tag{1}$$

式(1)中:在地图模型的笛卡尔坐标系中,$Z_1(x,y)$为环境中的某些点的坐标,Z_1为对应点的海拔高度,$Z_1(x,y)$通过常数系数(a,b,c,d,e,f,g)的不同组合来模拟数字地图的多样化基础地形特征。

1.2.2　障碍区域建模

通过较高的天然山峰模拟无人机飞行环境中的障碍物,其数学模型可以表示为

$$Z_2(x,y) = \sum_{i=1}^{n} H_i \exp\left[-\frac{(x-x_i)^2}{a_i^2} - \frac{(y-y_i)^2}{b_i^2}\right] + H_0 \tag{2}$$

式(2)中:$Z_2(x,y)$用于描述数字地图的山峰坡度数据;H_i、H_0和(x_i,y_i)分别为山峰的海拔高度、水平面高度和山峰的中心坐标点;a_i、b_i为山峰沿着x和y轴方向的坡度衰减量用来控制山峰的坡度;n为环境模型中山峰的数量。威胁区域用空心圆柱体表示,无人机在飞行过程中需避让威胁区域,环境建模如图1所示。

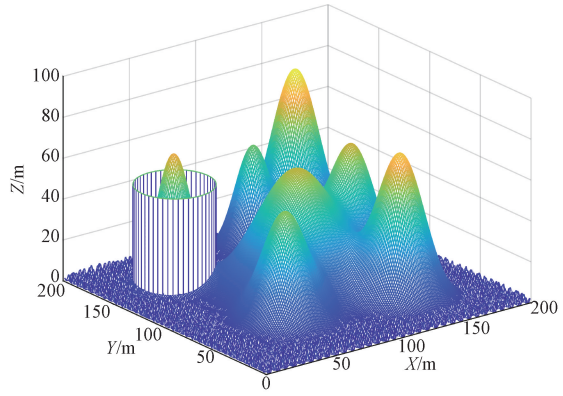


图1　环境模型

Fig. 1　Environmental model

1.2.3　三次B样条曲线路径优化

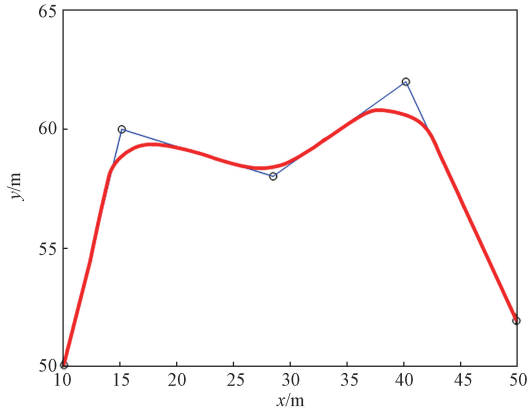
在路径规划中常用路径参数化技术来描述路径,路径可以由路径规划算法结合目标函数生成的一系列的有序路径坐标点来表示。基于多约束的三维环境模型,DMSSSA生成的路径会出现对应的拐点,为使规划的路径更贴切无人机的真实飞行路径,降低无人机飞行成本,引入了三次B样条曲线插值[17]对路径进行平滑处理如图2所示。B样条曲线是由贝塞尔曲线发展而来,它改进了贝塞尔曲线的优化方式,弥补了其缺陷,B样条曲线上点的方程为

$$P(u) = \sum_{i=0}^{n} d_i N_{i,k}(u) \tag{3}$$

式(3)中:$d_i(i=0,1,\cdots,n)$为第i个路径点;$N_{i,k}(u)$为由递归公式[式(4)]定义的k阶归一化B样条基函数,即

$$\begin{cases} N_{i,1}(u) = \begin{cases} 1, & u_i \leqslant u \leqslant u_{i+1} \\ 0, & \text{其他} \end{cases} \\ N_{i,k}(u) = \dfrac{u-u_i}{u_{i+k}-u_i} N_{i,k-1}(u) + \dfrac{u_{i+k+1}-u}{u_{i+k+1}-u_{i+1}} N_{i+1,k-1}(u),\ k>1 \end{cases} \tag{4}$$

该基本函数是由一个称为参数结的非递减序列确定,即$\{u_0 \leqslant u_1 \leqslant \cdots \leqslant u_{n+k}\}$,此外,$N_{i,k}(u)$是每个区间上的分段多项式,并且满足:$\sum_{i=0}^{n} N_{i,k}(t) \equiv 1$。



蓝色圈点为路径点;蓝色折线为连接各个路径点的路径;红色曲线则为经过三次B样条曲线插值技术处理后的路径,使路径更平滑

图2　三次B样条曲线插值

Fig. 2　Cubic B-spline curve interpolation

1.3　约束条件和成本函数

在三维空间中,高度的变化也会影响整体成本,所以在考虑整体成本函数时,高度成本应该是最有影响的因素。对于离线三维路径规划问题,地图信息和障碍物信息已给出,无人机可以在该环境中移动,路径成本和飞行转角成本也应该被纳入其中。由于路径规划方案是随机生成的,无人机可能会进入有威胁的区域,因此还必须考虑碰撞成本。

1.3.1　飞行高度成本

提出并引入飞行高度成本,其表示无人机根据估计路径需要改变飞行高度的次数所需的总成本,改变飞行高度需要改变推力,在实时的情况下这种推力的改变对于无人机的能效表现方面存在很大

影响。设 w_i、w_{i+1}、S 和 T 的海拔分量分别表示为 $w_i(z)$、$w_{i+1}(z)$、$S(z)$ 和 $T(z)$,那么路径中的总海拔高度变化可以由式(5)确定,因此飞行高度总成本如式(7)所示。

$$E_{\text{alt}} = \sum_{i=1}^{n}\left[\,|w_{i+1}(z) - w_i(z)|\,\right] + |T(z) - w_n(z)| + |w_1(z) - S(z)| \tag{5}$$

$$R_{\text{alt}} = |T(z) - S(z)| \tag{6}$$

$$J_1 = |E_{\text{alt}} - R_{\text{alt}}| \tag{7}$$

1.3.2 路径成本

在所有的路径规划问题中,较短的路径长度都具有较大的优势,这不仅减少了飞行时间,还降低了能源的消耗。考虑到无人机在整个任务期间以恒定速度飞行,路径成本(J_2)被转换为从起点(S)到目标(T)的最短路径长度问题,S 到 T 之间的路径点可以表示为 $W = (S, w_1, w_2, \cdots, w_n, T)$,将两个路径点之间的欧氏距离记为路径段 $|w_i w_{i+1}|$,那么 J_2 可表示为

$$J_2 = \sum_{i=0}^{n} |w_i w_{i+1}| \tag{8}$$

1.3.3 飞行转角成本

为保证规划路径的可飞性引入飞行转角成本函数,无人机的飞行角度主要由水平转角 α 和垂直俯仰角 β 控制,且必须符合无人机的实际角度约束。如图3所示,$|w_i w_{i+1}|$ 和 $|w_{i+1} w_{i+2}|$ 表示无人机飞行路径中的两个连续路径段,而 $|\boldsymbol{w}'_i \boldsymbol{w}'_{i+1}|$ 和 $|\boldsymbol{w}'_{i+1} \boldsymbol{w}'_{i+2}|$ 是它们在 xoy 平面上的投影。在这点上,如果将 $\boldsymbol{k}$ 标记为沿轴正方向的单位向量,则路径段投影 $\boldsymbol{w}'_i \boldsymbol{w}'_{i+1}$、水平转角 α_i 和垂直俯仰角 β_i 的计算公式如下。

$$\boldsymbol{w}'_i \boldsymbol{w}'_{i+1} = \boldsymbol{k} \times (\boldsymbol{w}_i \boldsymbol{w}_{i+1} \times \boldsymbol{k}) \tag{9}$$

$$\alpha_i = \arctan\left(\frac{\boldsymbol{w}'_i \boldsymbol{w}'_{i+1} \cdot \boldsymbol{w}'_{i+1} \boldsymbol{w}'_{i+2}}{\boldsymbol{w}'_i \boldsymbol{w}'_{i+1} \cdot \boldsymbol{w}'_{i+1} \boldsymbol{w}'_{i+2}}\right) \tag{10}$$

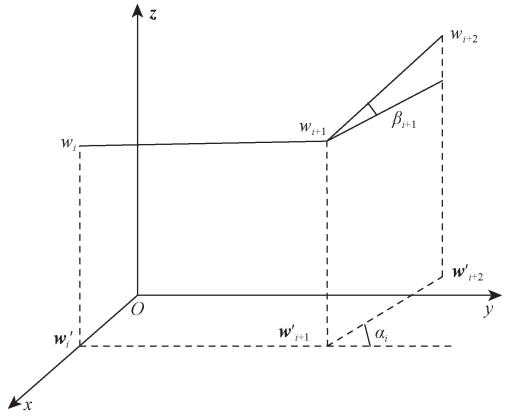


图3 飞行转角约束

Fig. 3 Flight angle constraint diagram

$$\beta_i = \arctan\left[\frac{w_{i+1}(z) - w_i(z)}{\|\boldsymbol{w}'_i \boldsymbol{w}'_{i+1}\|}\right] \tag{11}$$

式中:$w_i(z)$、$w_{i+1}(z)$ 为路径点 w_i、w_{i+1} 对应的海拔分量,将无人机水平转角和垂直俯仰角超出约束的惩罚系数记为 λ_1 和 λ_2,无人机飞行转角成本为

$$J_3 = \lambda_1 \sum_{i=1}^{n-2} \alpha_i + \lambda_2 \sum_{i=1}^{n-1} |\beta_i - \beta_{i-1}| \tag{12}$$

1.3.4 碰撞成本

在无人机飞行的环境中,除了一些不同高度天然山峰障碍物外,还包括一些用圆柱体表示的威胁区域,若无人机与威胁区域中心的距离为 d_{T},$d_{\text{T max}}$ 表示威胁区域能辐射到的最大距离,$d_{\text{T min}}$ 表示威胁区域内的危险区域,则无人机的碰撞成本(J_4)可以表示为如式(13)所示。

$$J_4 = \begin{cases} 0, & d_{\text{T}} > d_{\text{T}_{\max}} \\ \dfrac{1}{d_{\text{T}}}, & d_{\text{T}_{\min}} \leqslant d_{\text{T}} \leqslant d_{\text{T}_{\max}} \\ 1, & d_{\text{T}} < d_{\text{T}_{\min}} \end{cases} \tag{13}$$

1.4 目标函数

目标函数用来计算路径规划的成本从而分析路径的优劣,成本函数包括飞行高度成本、路径长度成本、飞行转角成本和碰撞成本4个方面,目标函数的表达式为

$$J = \sum_{k=1}^{4} k_i J_i \tag{14}$$

式(14)中:k_i 为特定因素在总成本函数中的权重,这些权重的取值取决于具体问题,并且对生成的最优路径起决定性作用的因素会被赋予更高的权重。

2 樽海鞘群算法

樽海鞘群以链式结构存在,领导者是位于链首“引导群”的樽海鞘个体,其余个体则被认为是“相互追随”的追随者,在觅食阶段,领导者负责引导种群向食物源位置游动,种群初始化为

$$\boldsymbol{X}_{i,j} = \text{rand}(N, D)(\boldsymbol{ub}_j - \boldsymbol{lb}_j) + \boldsymbol{lb}_j \tag{15}$$

式(15)中:$\boldsymbol{X}_{i,j}$ 为第 i 个樽海鞘在第 j 维空间的位置;N 为樽海鞘种群规模;D 为空间维度;$\boldsymbol{ub}_j$ 和 $\boldsymbol{lb}_j$ 分别为搜索空间的上下界。

在SSA中,食物源的位置是全局最优解,负责引导领导者的位置更新,领导者位置更新如下。

$$\boldsymbol{X}^t_{1,j} = \begin{cases} \boldsymbol{F}_j + c_1[(\boldsymbol{ub}_j - \boldsymbol{lb}_j)c_2 + \boldsymbol{lb}_j], & c_3 \geqslant 0.5 \\ \boldsymbol{F}_j - c_1[(\boldsymbol{ub}_j - \boldsymbol{lb}_j)c_2 + \boldsymbol{lb}_j], & c_3 < 0.5 \end{cases} \tag{16}$$

式(16)中:$\boldsymbol{X}^t_{1,j}$ 和 $\boldsymbol{F}_j$ 分别为第 t 次迭代第 j 维领导者和食物源的位置;c_2 和 c_3 为[0,1]区间内均匀分

布的随机数；c_1 为平衡算法全局搜索与局部开发的重要因素，并且参数 c_1 在迭代过程中自适应减小，如式(17)所示。

$$c_1 = 2e^{-(\frac{4t}{T_{max}})^m} \tag{17}$$

追随者的位置更新公式为

$$\boldsymbol{X}_{i,j}^{t+1} = \frac{1}{2}(\boldsymbol{X}_{i,j}^{t} + \boldsymbol{X}_{i-1,j}^{t}) \tag{18}$$

式(18)中：$\boldsymbol{X}_{i,j}^{t+1}$、$\boldsymbol{X}_{i,j}^{t}$ 分别为个体 $i(i \geqslant 2)$ 在 j 维空间的更新位置和原位置；$\boldsymbol{X}_{i-1,j}^{t}$ 为前一个个体的 j 维空间位置。

樽海鞘个体在迭代过程中可能会越界移动到搜索空间之外，用式(19)将其代回搜索空间。

$$\boldsymbol{X}_{i,j} = \begin{cases} \boldsymbol{lb}_j, & \boldsymbol{X}_{i,j} \leqslant \boldsymbol{lb}_j \\ \boldsymbol{ub}_j, & \boldsymbol{X}_{i,j} \leqslant \boldsymbol{ub}_j \\ \boldsymbol{X}_{i,j}, & \text{其他} \end{cases} \tag{19}$$

3　动态多子群樽海鞘群算法

已有诸多学者采用多子群策略来解决单子群元启发式群智能算法导致的种群多样性差、收敛效率低等问题，如陈忠云等[18]提出多子群共生策略，根据适应度值将种群均匀划分为多个子群，分别执行不同的位置更新策略来获得更好的寻优精度和收敛速度。李大海等[19]引入新式小生境机制将种群自动划分为若干个子群，以保持种群多样性并提升算法的收敛速度。在 SSA 算法的基础上，保持种群个体总量不变的条件下，结合 *K*-means + + 聚类算法将种群动态的划分为领导者、追随者和链尾者 3 个子群。首先在领导者位置更新公式的参数 c_1 中引入随机游走因子来控制领导者的收敛特性，以更好地平衡算法的搜索和开发能力；然后采用趋优追随机制来更新追随者位置，充分发挥精英个体的引导作用，增强全局搜索能力；最后使用精密消除策略来更新链尾者的位置，使其在更优的位置进行精细搜索，增强局部开发能力。

3.1　基于 *K*-means + + 聚类优化的动态多子群划分策略

聚类算法的目的是根据特定标准将数据集分成不同的簇，将相似度高的数据归为同一簇，而将差异较大的数据分到不同的簇中。*K*-means + + 聚类算法是 Arthur 等[20]在 2007 年提出的一种聚类算法，相较于 *K*-means 聚类算法随机选择的初始质心，*K*-means + + 聚类算法在一定程度上解决了对初始质心选择的敏感性问题。*K*-means + + 聚类算法是一种分区聚类算法，将给定的数据集划分为簇并找到数据集中各个数据点与任一簇质心之间的最小平方误差，然后将每个数据点分配给距离它最近的簇，具体步骤如下。

步骤 1　确定簇的个数 k，从数据集 X 中均匀随机的选择一个初始簇质心 $\boldsymbol{\mu}_1$。

步骤 2　$D(x)$ 表示数据点 x 到任一质心的最短欧氏距离，以最大比例 $\frac{D(x')^2}{\sum_{x \in X} D(x)^2}$ 选择下一个质心 $\boldsymbol{\mu}_j = \boldsymbol{x}' \in X(j = 2,3,\cdots,k)$，重复此步骤直至总共取到 k 个质心。

步骤 3　将簇 N_j 设置为 X 中所有距离 $\boldsymbol{\mu}_j$ 比距离 $\boldsymbol{\mu}_i(i \neq j)$ 更近数据点的集合，距离统一用欧氏距离度量。

步骤 4　重新计算簇 N_j 的质心：$\boldsymbol{\mu}_j = \frac{1}{|N_j|}\sum_{\boldsymbol{x} \in N_j} \boldsymbol{x}$。

步骤 5　重复步骤 3 和步骤 4，直至质心 $\boldsymbol{\mu}$ 不再变化。

K-means + + 聚类算法的目标是将所有数据点与簇质心之间的平方误差之和最小化，其表达式为

$$\varphi(X) = \sum_{j=1}^{k} \sum_{\boldsymbol{x}_i \in N_j} \left\| \boldsymbol{x}_i - \boldsymbol{\mu}_j \right\|^2 \tag{20}$$

式(21)中：$\boldsymbol{x}_i$ 为簇 N_j 内的数据点；$\boldsymbol{\mu}_j$ 为簇内质心。

结合 *K*-means + + 聚类算法提出动态多子群划分策略如下。

首先确定子群个数，选择种群适应度值最优个体的位置当作第一子群的初始质心。

其次计算所有樽海鞘个体与该质心的最短欧氏距离 $D(x)$，根据最大欧氏距离平方比 $\frac{D(x')^2}{\sum_{x \in X} D(x)^2}$ 依次确定其他子群初始质心位置，再将所有樽海鞘个体分配到距离最近质心所在的子群。

最后各子群重新计算质心的位置，直至子群质心的位置始终不变时，即可确定子群质心的最终位置以及子群个体分布及数量 N_i，如图 4(a)所示。将第一质心所在的子群划分为领导者子群，使其能够在食物源周围进行搜索，负责平衡算法的搜索与开发。通过计算第一子群的质心 $\boldsymbol{\mu}_1$ 与其他两个质心 $\boldsymbol{\mu}_j$ 之间的欧式距离 L_j 来划分追随者子群和链尾者子群的位置[式(21)]，最终种群划分如图 4(b)所示。为了使樽海鞘能通过快速协调的变化和觅食来实现更好的搜索，将沿用基础樽海鞘群算法的链式建模思想，按照子群内所有个体与子群质心的欧氏距离从小到大顺序把所有个体链接在一起，最后把领导者子群位于链首，随后是追随者子群，最后是链尾者子群。

$$L_j^t = \sqrt{(\mu_j^t - \mu_1^t)^2}, j \in (2,3,\cdots,k) \tag{21}$$

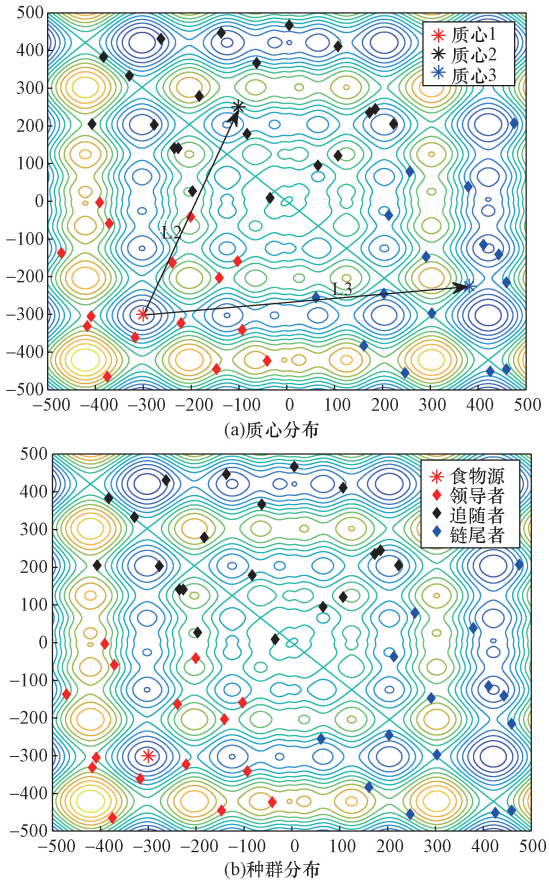


图 4 种群划分示意图

Fig. 4 Population division schematic diagram

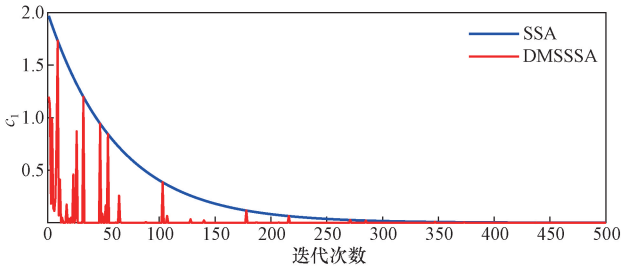


图 5 c_1 在 SSA 和 DMSSSA 之间的差异

Fig. 5 Differences of c_1 between SSA and DMSSSA

3.2 多策略协同改进

3.2.1 领导者位置更新

在领导者位置更新公式[式(16)]中,参数 c_1 是 SSA 中最为关键且敏感的参数,起着平衡算法全局搜索与局部开发能力的作用。在式(22)中,将 m 的值设为 2.5,参数 c_1 能使算法的搜索能力和开发能力处于更好的平衡状态[18],其次引入随机游走因子 R,让参数 c_1 的取值随迭代进程的变化而变化,以达到改变领导者收敛特性的效果。如图 5 所示,在迭代前期,DMSSSA 中参数 c_1 的取值始终控制在 0 和原 SSA 的参数 c_1 之间,且在较长的时间取值为 0,这让领导者个体在迭代前期向着食物源的位置游走,且只负责对食物源周围进行搜索,能较好地维持算法的全局搜索能力;在迭代后期,参数 c_1 的取值趋近于 0,领导者个体将紧跟食物源的位置游走,此时领导者负责锁定食物源的位置,引导追随者和链尾者进行局部位置更新,以增强算法的局部开发能力。在 DMSSA 的领导者位置更新策略中,参数 c_1 起着至关重要的作用,随机因子 R 的引入能增强领导者在子群中的协同能力。将式(17)替换为式(22)来更新参数 c_1,再根据式(16)对领导者个体位置进行更新,公式为

$$c_1 = 2e^{-\left(\frac{4Rt}{T_{\max}}\right)^m} \tag{22}$$

式(22)中:随机游走因子 R 是[0,50]的随机整数。参数 c_1 随 R 的增大而减小。

3.2.2 趋优追随

在标准 SSA 中,追随者新个体通过链中相邻两个体位置产生,如式(18)所示,此种方法存在以下不足:没有考虑新个体与原有个体的适应度优劣而盲目替换原有个体;追随者个体仅根据链中相邻个体位置进行更新,使新生成个体对前一个个体有较强的依赖性,缺乏与其他个体之间的学习。因此文献[21]引入线性递减惯性权重来决定前一个个体对当前个体的影响程度,在一定程度上增强了算法的局部搜索能力,但可能使得非精英个体在迭代过程中发挥作用;文献[22]提出了一种自适应惯性权重的方法引入种群的成功率,进一步增强了局部搜索能力,但仍存在非精英个体在引导过程中发挥作用的问题。故提出并引入趋优追随机制更新追随者子群,使其根据链中相邻个体适应度关系采用不同的更新机制,加快追随者个体趋向适应度更优的位置,减弱其对前一个个体的依赖性,加强与其他个体之间的信息交流提高算法的收敛速度,提出基于趋优追随机制改进的追随者个体位置更新策略具体如下。

(1)加速追随机制。若前一个个体的适应度值比当前个体更优时,引入加速追随机制更新追随者的位置,即

$$\boldsymbol{X}_{i,j}^{t+1} = \frac{1}{2}r_1\boldsymbol{X}_{i,j}^{t} + \left(1 - \frac{1}{2}r_1\right)\boldsymbol{X}_{i-1,j}^{t} \tag{23}$$

(2)反向游走机制。若当前个体适应度值优于前一个个体时,采用反向游走机制使当前个体不再追随前一个个体,朝反方向游动,其位置更新方式为

$$\boldsymbol{X}_{i,j}^{t+1} = \boldsymbol{X}_{i,j}^{t} + r_1(\boldsymbol{X}_{i-1,j}^{t} - \boldsymbol{X}_{i,j}^{t}) \tag{24}$$

(3)随机跳出机制。若前一个个体的适应度值与当前个体相当不足以引导当前个体趋向更优位置时,算法容易陷入局部最优,此时在随机选择

的一个个体位置上采用与领导者相似的位置更新方式来帮助当前个体跳出,位置更新公式如下。

$$\boldsymbol{X}_{i,j}^{t+1}=\begin{cases}\boldsymbol{X}_{r,j}^{t}+r_1[(\boldsymbol{ub}_j-\boldsymbol{lb}_j)r_2+\boldsymbol{lb}_j], & r_3\geqslant 0.5\\ \boldsymbol{X}_{r,j}^{t}-r_1[(\boldsymbol{ub}_j-\boldsymbol{lb}_j)r_2+\boldsymbol{lb}_j], & r_3<0.5\end{cases} \tag{25}$$

式(25)中:$N_1\leqslant r<N_2$;$\boldsymbol{X}_{r,j}^{t}$ 为第 t 次迭代第 j 维空间的追随者子群中随机选择的一个樽海鞘个体;参数 r_1、r_2 和 r_3 为区间[0,1]生成的随机数。

3.2.3　精密消除策略

为了获得更多高质量的解向量,许多学者将对立学习[14]引入算法中,尽管这种策略是有效的,但它也存在不足:若最优解恰好位于对立学习解和原始解的对称平面上,这两个解对应的适应度值将完全相同,在这种情况下,对立学习策略就会失效。受到进化理论“适者生存”的启发,结合自适应 T 分布提出精密消除策略来淘汰原链尾者子群个体并让具有更优适应度的个体替代,具体公式为

$$\boldsymbol{X}_{i,j}^{t+1}=\boldsymbol{F}_j+(\boldsymbol{ub}-\boldsymbol{lb})R_{\mathrm{r}}T\left[\frac{1}{m}\exp\left(\frac{\ln t_{\mathrm{c}}}{t_{\mathrm{c}}}\right)^{t}\right] \tag{26}$$

式(26)中:R_{r} 为动态半径,$R_{\mathrm{r}}=1-t/T_{\max}$,$t$ 为当前迭代次数,$T_{\max}$为最大迭代次数。

T 分布为统计学中使用的抽样分布,由缩放因子 m 和迭代临界点 t_{c} 控制自适应 T 分布的自由度,使其能够根据迭代次数实现自适应调节,在迭代前期较小的自由度使链尾者个体有更大的突变概率,以获得更优解,在迭代后期较大的自由度使链尾者在食物源位置附近进行突变,来保持种群多样性。在迭代后期往往会出现种群个体聚集的情况,精密消除策略将自适应 T 分布、动态半径和搜索上下界相乘以构建弹性半径,使新生成个体的位置在迭代后期仍有一定的向外扩展概率,以提高算法的精细搜索能力,避免算法陷入局部最优。将缩放因子 m 设置为 10,将迭代临界点 t_{c} 设置为 $T_{\max}/5$,t 为当前迭代次数,$T_{\max}$ 为最大迭代次数。精密消除策略的示意图如图 6 所示。

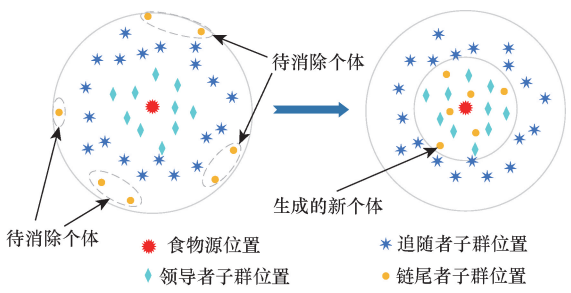


图 6　精密消除策略

Fig. 6　Precision elimination strategy

3.3　DMSSSA 算法设计与时间复杂度分析

综合 3.1 节和 3.2 节,DMSSSA 算法的伪代码如下。

输入:最大迭代数 $T_{\max}$、种群个体数量 N、搜索上下边界 ub 和 lb、问题维度 D;

输出:全局最优位置 F_j 以及其适应度值 $f(F_j)$。

[1]种群初始化:$X_i=(x_{i,1},x_{i,2},\cdots,x_{i,j},\cdots,x_{i,D})$,设置相关参数;

[2] 计算种群适应度并排序,确定食物源 F_j;

[3]while $t\leqslant T_{\max}$ then

[4]利用 *K*-means + + 聚类算法对种群进行分群,确定各子群个体数量 N_1、N_2、N_3;

[5] for $i=1:N$

[6] if $i\leqslant N_1$ then

[7] 按式(22)计算 c_1,并根据式(16)更新领导者位置;

[8] else if $N_1<i\leqslant N_1+N_2$

[9] 按式(23)~式(25)更新追随者位置;

[10] else if $N_1+N_2<i\leqslant N$

[11] 按式(26)更新链尾者位置;

[12] end if

[13] end for

[14] 再次计算种群适应度值,更新食物源位置;

[15]end while

[16]输出最优位置 F_j 及其适应度值 $f(F_j)$。

DMSSSA 算法流程图如图 7 所示。

传统樽海鞘群算法的时间复杂度为 $O[T(N\times D+N\times f)]$[12],其中 D 为问题维度,T 为算法的最大迭代数,N 为种群个体数量,f 为待解决问题适应度函数的计算量。根据算法的伪代码和图 7,可以计算出 DMSSSA 算法的时间复杂度,在迭代之前初始化阶段需要对 N 个个体的每个维度进行初始化设置,所以该步骤的时间复杂度表示为 $O_1(N\times D)$。在初始化之后需要对全部个体进行适应度评估并且使用快速排序从小到大排序,然后确定食物源的位置,所以该步骤的时间复杂度为 $O_2(N\times f+N\times\log_2 N)$。进入迭代之后,需利用 *K*-means + + 聚类算法进行划分子群,根据文献[20]可以将该步骤的时间复杂度表示为 $O_3[T\log_2 k)]$,其中 k 为质心数量。领导者、追随者和链尾者子群个体进行相应的位置更新,由于樽海鞘个体总量不变,且迭代次数都为 T,所以该步骤的时间复杂度可以表示为 $O_4[T(N\times D)]$。各子群在每次迭代更新之后,需要对全部个体再次计算适应度值并更新食物源的位置,该步骤时间复杂度为 $O_5[T(N\times f)+T\times N]=O_5[T(N\times f)]$。综上所述,合并同类项之后 DMSSSA 的时间复杂度为 $O=O_1+O_2+O_3+O_4+O_5=O[T(N\times D+N\times f)]$,DMSSSA 的时间复杂度与 SSA 为同一数量级,证明 DMSSSA 在提升寻优性能的情况下并未增加算法复杂性。

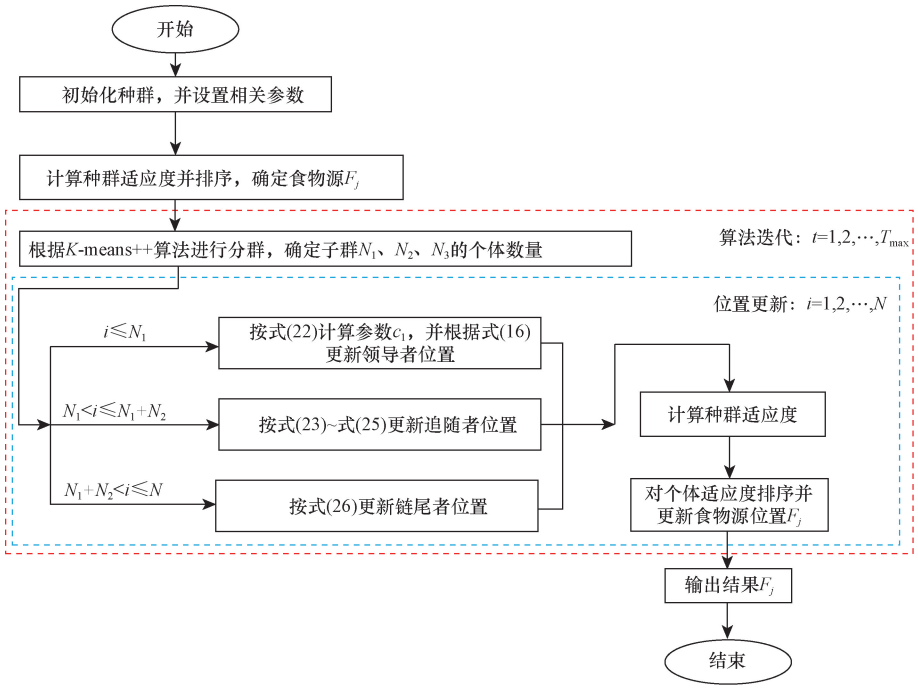


图 7 DMSSSA 的执行流程

Fig. 7 Flowchart of DMSSSA

4 仿真实验与结果分析

4.1 DMSSSA 算法性能检验

4.1.1 实验设计

所有仿真实验均在相同的实验环境中进行，实验仿真软件为 MATLAB R2022b，运行在 Intel(R) i7-8750H @ 2.20GHz、16.0GB 内存和 Windows11(64 位)操作系统上。实验对 DMSSSA 以及其他 5 种元启发式算法：ISSA[23]、MSNSSA[18]、IBSO[24]、MBFPA[25]、SSA[12]进行性能测试。选取了 CEC2017 测试集中的 12 个具有代表性的测试函数，其中F_1 ~ F_4 为多峰函数、F_5 ~ F_8 为混合函数、F_9 ~ F_{12}为复合函数。所选取的这些测试函数复杂度较高且含有大量的局部最优解，测试集函数如表 1 所示。

将 DMSSSA 与其他 5 个对比算法在维度为 100 的测试函数上进行测试，对比算法的参数设置皆取自原文献，具体设置如表 2 所示，每个算法迭代 500 次，并在各个测试函数上独立运行 30 次以排除实验数据的随机性。评估各算法稳定性和优化能力的指标包括：30 次运行后获得的最优解的平均值、方差和排名。排名评估标准优先考虑平均值，其次是方差，最优解平均值反映出算法的寻优性能，方差反映出算法的稳定性，如表 4 所示，排名最优者用加粗字体表示。

表 1 测试函数

Table 1 Benchmark function

函数	函数名称	类型	F_i^*
F_1	Shifted and Rotated Rosenbrock's Function	Simple Multimodal Functions	400
F_2	Shifted and Rotated Rastrigin's Function		500
F_3	Shifted and Rotated Expanded Scaffer's F6 Function		600
F_4	Shifted and Rotated Non-Continuous Rastrigin's Function		800
F_5	Hybrid Function 2 (N=3)	Hybrid Functions	1 200
F_6	Hybrid Function 4 (N=4)		1 400
F_7	Hybrid Function 6 (N=4)		1 600
F_8	Hybrid Function 6 (N=6)		2 000
F_9	Composition Function 2 (N=3)	Composition Functions	2 200
F_{10}	Composition Function 3 (N=4)		2 300
F_{11}	Composition Function 5 (N=5)		2 500
F_{12}	Composition Function 8 (N=6)		2 800

注：F_i^* 为函数的最优收敛值。

表 2 算法的参数设置

Table 2 Parameter settings of algorithms

算法	参数设置
DMSSSA	$D=100, N=30, T_{max}=500, m=2.5, R=\text{rand}(0,50)$
ISSA	$D=100, N_{init}=30, N_{tot}=240, T_{max}=500, m=2$
MSNSSA	$D=100, N=30, T_{max}=500, m=2.5, b=2$
IBSO	$D=100, N=30, T_{max}=500, c_1=0.5, c_2=0.05, c_3=2$
MBFPA	$D=100, N=30, T_{max}=500, c=0.01, a=0.1$
SSA	$D=100, N=30, T_{max}=500, m=2$

4.1.2 实验结果与收敛曲线对比分析

如表 5 所示，DMSSSA 在 12 个测试函数中皆有

较好的表现,最优平均值均是第一,在总排名也取得了第一的成绩。DMSSSA 的方差除了在测试函数 F_6 和 F_{11} 中排名略落后于 MBFPA 和 MSNSSA 外,在其他 10 个测试函数中均取得第一,总的方差排名也是第一,说明动态多子群策略相较于其他算法的改进策略在优化能力、稳定性和鲁棒性方面具有较大的优势。DMSSSA 在所有测试函数上的收敛精度都优于其他算法,在测试函数 F_5、F_6、F_7、F_{10}、F_{12} 中的收敛精度比其他对比算法高一个数量级。

在图 8 中给出了各算法在 100 维下运行 30 次的平均收敛曲线图,为了更直观地对比实验结果,在部分收敛曲线上做了放大处理,如图 8(a)和图 8(e)所示。在图 8(a)~图 8(d)多峰函数中,DMSSSA均以较快的收敛速度领先于其他算法,在 F_2、F_3 和 F_4 中,大约迭代到 300 次时的收敛精度已经超过其他对比算法的最终收敛精度,随着迭代次数的增加,算法的最终收敛精度要优于其他改进算法;图 8(e)~图 8(h)为混合函数 F_5、F_6、F_7、F_8 的收敛曲线图,算法在混合函数上也有优秀的表现,收敛速度和收敛精度都明显优于其他改进算法,尤其是在 F_5 和 F_7 中,其收敛精度比其他算法低 1~2 个数量级。图 8(i)~图 8(l)为复合函数 F_9、F_{10}、F_{11}、F_{12} 的收敛曲线图,尽管测试函数含有大量的局部最优解,例如,在 F_9 函数中,其他一些对比算法很早就陷入了局部最优,并在迭代到 350 次左右停止收敛,而 DMSSSA 算法在迭代到 100~200 次时短暂地陷入局部最优,迭代至 200 次后迅速搜索到更优解,由此可以反映出 DMSSSA 跳出局部最优的能力,函数 F_6 和 F_8 也可以反映出算法跳出局部最优的能力。

4.1.3　Friedman 检验

为进一步证明 DMSSSA 算法与其他 5 种对比算法存在显著性差异,采用 Friedman 检验对各算法 30 次运行结果的平均值进行分析。Friedman 检验[26]是用于检测各数据集之间是否存在显著性差异的非参数双向方差分析方法。结果如表 3 所示,其中的 P 表示渐进显著性,用来判断各算法之间是否存在显著性差异,若 $P<0.01$ 则表明算法的各项数据之间存在显著性差异。对于 30 维、50 维、100 维,P 分别为 $2.312\,5\times10^{-9}$、$5.342\,3\times10^{-10}$ 和 $3.776\,4\times10^{-9}$,都远远小于 0.01,说明 DMSSSA 算法在 3 个维度上都与其他对比算法存在显著性的差异。表 2 中其他值表示算法在各维度下秩的平均值,在 3 个不同维度中,DMSSSA 的秩均值都是最小的,进一步证明了 DMSSSA 的寻优能力。综上所述,DMSSSA 总体上明显优于其他对比算法。

表 3　Friedman 统计结果

Table 3　Friedman statistical results

维度	30 维	50 维	100 维
P	$2.312\,5\times10^{-9}$	$5.342\,3\times10^{-10}$	$3.776\,4\times10^{-9}$
ISSA	2.15	2.33	1.95
MSNSSA	3.77	3.65	4.08
IBSO	3.32	4.45	5.16
MBFPA	3.77	3.97	3.21
SSA	5.24	5.05	4.93
DMSSSA	1.16	1.04	1.34

4.1.4　完整性消融实验

为了验证 DMSSSA 各个策略的有效性,对 DMSSSA 中改进策略进行完整性消融实验。设将标准 SSA 按动态多子群策略划分的算法为 DMSSSA1,在 DMSSSA1 的基础上只对领导者个体位置更新进行改进的算法为 DMSSSA2,在 DMSSSA2 的基础上引入追随者个体趋优追随策略的算法为 DMSSSA3,最后在 DMSSSA3 的基础上增加精密消除策略的算法为 DMSSSA。将标准 SSA、DMSSSA1、DMSSSA2、DMSSSA3 和 DMSSSA 基于表 1 的基准函数进行测试,各算法迭代次数为 500,并独立运行 30 次,算法参数设置均保持一致,表 4 展现了 DMSSSA 不同改进策略收敛精度情况。

表 4　各策略测试结果

Table 4　Test results of each strategies

函数	平均值/方差				
	SSA	DMSSSA1	DMSSSA2	DMSSSA3	DMSSSA
F_1	$3.04\times10^4/6.51\times10^3$	$1.97\times10^4/3.66\times10^3$	$1.34\times10^4/2.80\times10^3$	$1.25\times10^4/2.76\times10^3$	$1.31\times10^3/7.63\times10^1$
F_2	$1.62\times10^3/1.13\times10^2$	$1.53\times10^3/7.72\times10^1$	$1.50\times10^3/6.12\times10^1$	$1.47\times10^3/7.48\times10^1$	$1.33\times10^3/4.27\times10^1$
F_3	$7.28\times10^2/7.32\times10^0$	$7.03\times10^2/4.98\times10^0$	$6.80\times10^2/3.76\times10^0$	$6.85\times10^2/4.04\times10^0$	$6.79\times10^2/3.12\times10^0$
F_4	$2.21\times10^3/9.58\times10^1$	$2.20\times10^3/1.06\times10^2$	$2.09\times10^3/1.21\times10^2$	$2.11\times10^3/8.67\times10^1$	$1.92\times10^3/7.50\times10^1$
F_5	$7.46\times10^8/2.61\times10^8$	$6.21\times10^8/2.25\times10^9$	$6.15\times10^8/1.33\times10^{10}$	$6.20\times10^8/8.20\times10^8$	$5.96\times10^8/1.78\times10^8$
F_6	$2.43\times10^4/4.07\times10^3$	$2.31\times10^4/5.17\times10^3$	$1.43\times10^4/4.37\times10^3$	$3.34\times10^4/3.50\times10^3$	$1.21\times10^4/1.42\times10^3$
F_7	$8.43\times10^3/9.25\times10^2$	$8.27\times10^3/9.62\times10^2$	$8.17\times10^3/8.64\times10^2$	$8.22\times10^3/1.07\times10^3$	$7.39\times10^3/9.48\times10^2$
F_8	$6.63\times10^3/4.54\times10^2$	$6.30\times10^3/4.33\times10^2$	$6.19\times10^3/3.42\times10^2$	$6.11\times10^3/2.45\times10^2$	$5.75\times10^3/6.33\times10^2$
F_9	$2.56\times10^4/1.71\times10^3$	$2.38\times10^4/1.57\times10^3$	$2.29\times10^4/1.36\times10^3$	$2.33\times10^4/1.73\times10^3$	$2.22\times10^4/1.84\times10^3$
F_{10}	$4.79\times10^3/2.30\times10^2$	$4.61\times10^3/2.19\times10^2$	$4.58\times10^3/2.21\times10^2$	$4.64\times10^3/2.97\times10^2$	$4.57\times10^3/1.09\times10^2$
F_{11}	$8.13\times10^3/6.93\times10^2$	$8.01\times10^3/9.74\times10^2$	$6.96\times10^3/1.11\times10^3$	$6.72\times10^3/2.24\times10^3$	$3.78\times10^3/9.55\times10^1$
F_{12}	$7.20\times10^3/1.03\times10^3$	$7.02\times10^3/7.95\times10^2$	$6.97\times10^3/1.45\times10^3$	$6.89\times10^3/1.60\times10^3$	$4.14\times10^3/1.18\times10^2$

从表4中可以看出,DMSSSA1在12个测试函数中的收敛精度都优于标准SSA,说明动态多子群分群策略的优越性。在分群策略的基础上对3个子群的位置更新方式进行改进的算法DMSSSA2、DMSSSA3和DMSSSA的测试结果都优于DMSSSA1,说明各子群位置更新策略的改进对算法性能有显著提升,并且DMSSSA的测试结果比DMSSSA2和DMSSSA3都要好,说明多策略协同改进进一步提升了算法的性能。实验结果表明每个策略在DMSSSA上都是有效的,各个策略都是不可或缺的,共同提升了DMSSSA的综合寻优性能。

4.2 基于DMSSSA的无人机三维路径规划测试及分析

无人机路径规划问题本质上是一个非确定性多项式求解问题,采用DMSSSA来解决这一问题,则是通过对目标函数进行求解寻优。樽海鞘链的每一只樽海鞘代表着一条可能的路径,其维数对应着路径点的个数,而樽海鞘个体的各维数值则代表着路径点的坐标。在路径规划中,樽海鞘个体的适应度则对应着目标函数的值,该目标函数由飞行高度成本函数、路径长度成本函数、飞行转角成本函数和碰撞成本函数构成。在樽海鞘群中,领导者子

表5 100维时各算法30次运行结果

Table 5 Results from 30 runs of each algorithm in 100 dimensions

函数	指标	ISSA	MSNSSA	IBSO	MBFPA	SSA	DMSSSA
	平均值	4.28×10^{3}	2.83×10^{3}	4.87×10^{3}	3.58×10^{3}	4.57×10^{4}	1.22×10^{3}
F_1	方差	1.05×10^{3}	5.07×10^{2}	1.37×10^{3}	8.83×10^{2}	1.17×10^{4}	1.05×10^{2}
	排名	4	2	5	3	6	1
	平均值	1.62×10^{3}	1.48×10^{3}	1.71×10^{3}	1.44×10^{3}	1.95×10^{3}	1.38×10^{3}
F_2	方差	1.09×10^{2}	8.29×10^{1}	1.32×10^{2}	6.58×10^{2}	1.05×10^{2}	6.36×10^{1}
	排名	4	3	5	2	6	1
	平均值	6.87×10^{2}	6.74×10^{2}	6.92×10^{2}	6.76×10^{2}	6.99×10^{2}	6.69×10^{2}
F_3	方差	6.01×10^{0}	4.44×10^{0}	7.94×10^{0}	7.52×10^{0}	6.58×10^{0}	4.26×10^{0}
	排名	4	2	5	3	6	1
	平均值	1.95×10^{3}	1.86×10^{3}	2.08×10^{3}	1.85×10^{3}	2.40×10^{3}	1.82×10^{3}
F_4	方差	1.12×10^{2}	1.11×10^{2}	1.39×10^{2}	8.49×10^{1}	1.14×10^{2}	8.43×10^{1}
	排名	4	3	5	2	6	1
	平均值	2.79×10^{9}	1.58×10^{9}	4.94×10^{9}	4.99×10^{9}	5.38×10^{9}	6.39×10^{8}
F_5	方差	1.11×10^{9}	6.94×10^{8}	1.81×10^{9}	8.73×10^{9}	1.61×10^{9}	1.34×10^{8}
	排名	3	2	4	5	6	1
	平均值	8.79×10^{3}	8.36×10^{3}	9.97×10^{3}	8.98×10^{3}	1.34×10^{4}	7.76×10^{3}
F_6	方差	1.10×10^{3}	9.05×10^{2}	1.33×10^{3}	3.30×10^{2}	1.64×10^{4}	7.45×10^{2}
	排名	3	2	5	4	6	1
	平均值	9.64×10^{3}	1.22×10^{4}	1.26×10^{4}	1.04×10^{4}	1.68×10^{4}	7.84×10^{3}
F_7	方差	1.13×10^{3}	1.55×10^{3}	1.73×10^{3}	2.43×10^{3}	2.56×10^{3}	8.34×10^{2}
	排名	2	4	5	3	6	1
	平均值	5.77×10^{3}	6.58×10^{3}	7.80×10^{3}	5.93×10^{3}	7.18×10^{3}	5.63×10^{3}
F_8	方差	6.64×10^{2}	6.00×10^{2}	5.96×10^{2}	6.31×10^{2}	4.61×10^{2}	4.49×10^{2}
	排名	2	4	6	3	5	1
	平均值	2.74×10^{4}	2.91×10^{4}	2.69×10^{4}	2.69×10^{4}	3.39×10^{4}	2.12×10^{4}
F_9	方差	1.82×10^{3}	1.91×10^{3}	2.19×10^{3}	2.97×10^{3}	1.75×10^{3}	1.55×10^{3}
	排名	4	5	2	3	6	1
	平均值	8.28×10^{3}	1.15×10^{4}	6.38×10^{3}	5.66×10^{3}	2.69×10^{4}	3.85×10^{3}
F10	方差	9.20×10^{2}	1.03×10^{3}	7.20×10^{2}	7.38×10^{2}	4.76×10^{3}	7.31×10^{1}
	排名	4	5	3	2	6	1
	平均值	4.48×10^{3}	4.47×10^{3}	4.63×10^{3}	5.98×10^{3}	6.52×10^{3}	4.12×10^{3}
F_{11}	方差	3.16×10^{2}	3.20×10^{2}	3.72×10^{2}	1.97×10^{3}	7.86×10^{2}	4.49×10^{2}
	排名	2	3	4	5	6	1
	平均值	7.53×10^{3}	6.80×10^{3}	7.60×10^{3}	8.04×10^{3}	2.48×10^{4}	4.00×10^{3}
F_{12}	方差	1.33×10^{3}	9.81×10^{2}	7.76×10^{2}	2.00×10^{3}	3.26×10^{3}	1.34×10^{2}
	排名	3	2	4	5	6	1
排名第一次数		0	0	0	0	0	12
平均排名		3.25	3.08	4.41	3.33	5.91	1
总排名		3	2	5	4	6	1

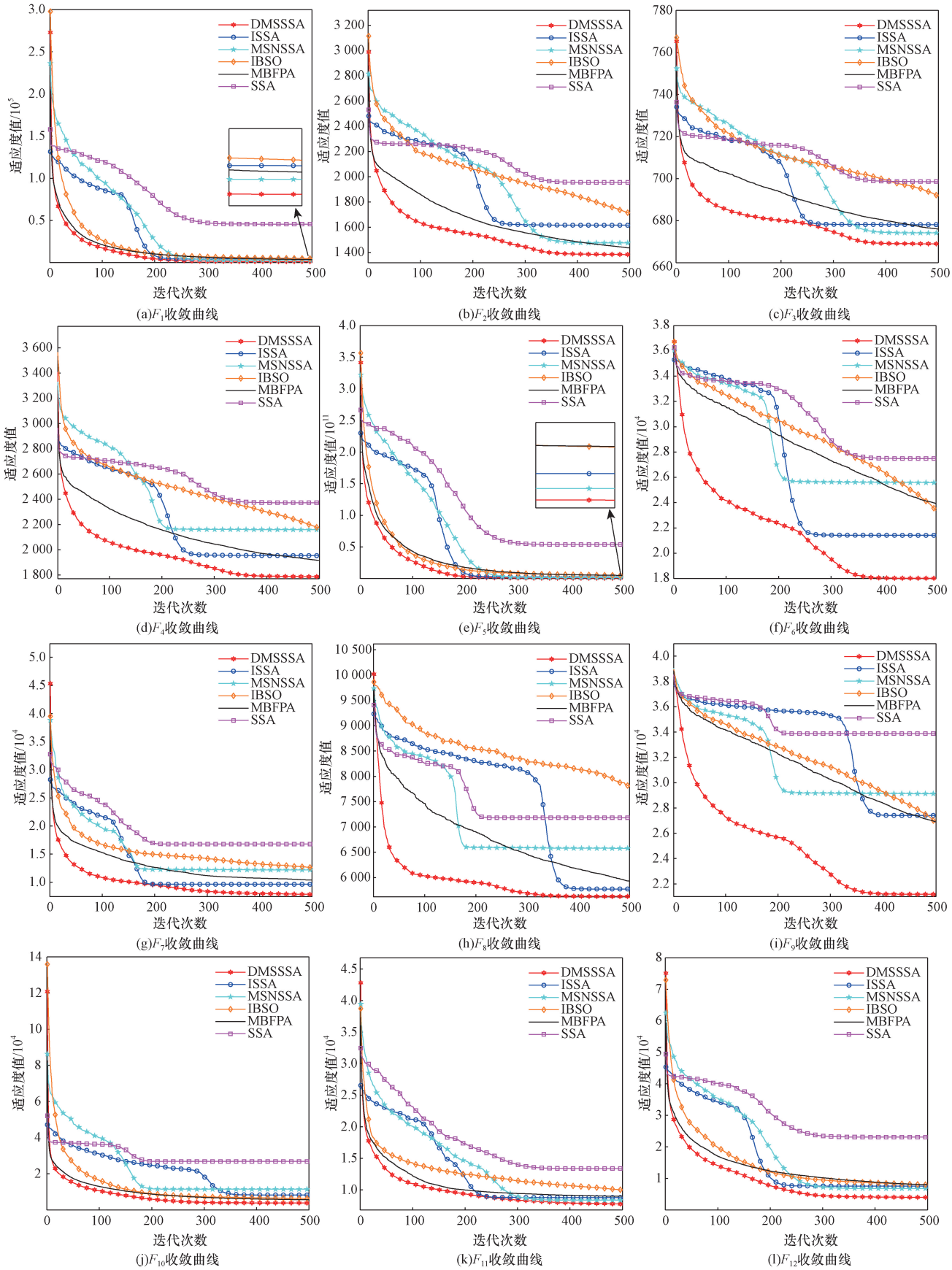


图 8　各算法平均收敛曲线图

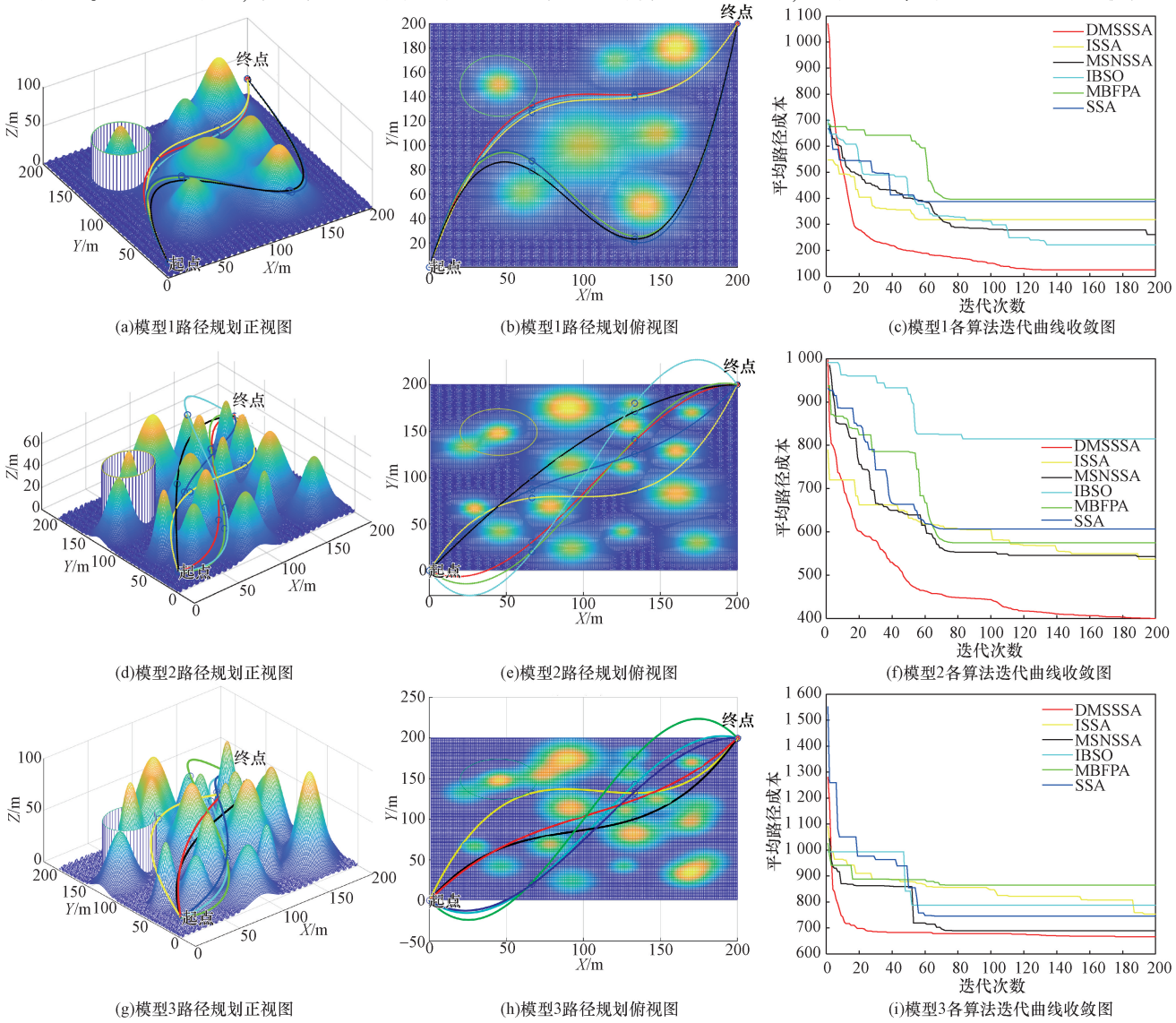
Fig. 8　Average convergence curve plots of each algorithm

群代表当前目标函数值较优的个体,链尾者子群则代表目标函数值较差的个体。通过不断更新领导者、追随者和链尾者的位置,樽海鞘群逐步趋向最优位置,经过多次迭代,最终得到的最优解即为最佳路径。

三维路径规划采用与上述基准函数测试相同的软硬件实验环境,各维度范围在[0,200]的空间中,建立了3个不同的环境模型,路径规划难度依次递增。使用上述算法性能测试时所用的5种算法作为对比算法,为保证实验的公平性,各算法均采用原文参数设置,统一设置起点坐标为 $S(0,0,20)$,终点坐标为 $T(200,200,20)$,种群规模为50,迭代次数为200。为减少实验的偶然性让各算法分别运行30次,分别统计了路径长度最优值、平均值、方差、平均用时和有效路径率等指标用来体现算法性能的优劣。这些指标中,平均值越小代表路径规划平均成本越低,方差体现出在面对复杂三维环境时算法的稳定性和鲁棒性,有效路径率则反映算法在多次独立实验中找到满足约束路径的比例。

环境模型1如图9(a)所示,在该模型中仅设置了少量的山体障碍物和威胁区域,同时各座山体分布地在较为分散,这样算法在路径规划时具有更高的概率找到最优路径。实验结果如表6所示,所有算法均在30次独立运行时能够规划出有效路径,如图9(c)所示,从收敛精度上看,DMSSSA都明显要低于其他算法,从收敛曲线上看,DMSSSA的曲线更为平滑,拐点较少,证明该算法在多次独立运行中具有较强的稳定性。

环境模型2和模型3属于高难度模型,用于进一步评估DMSSSA在三维路径规划的性能,如图9(d)~图9(g)所示,相较于模型1其增加了山体障碍物的数量,山体沿 X、Y 轴的坡度均减小使其



(a)模型1路径规划正视图 (b)模型1路径规划俯视图 (c)模型1各算法迭代曲线收敛图

(d)模型2路径规划正视图 (e)模型2路径规划俯视图 (f)模型2各算法迭代曲线收敛图

(g)模型3路径规划正视图 (h)模型3路径规划俯视图 (i)模型3各算法迭代曲线收敛图

图9 各算法在不同模型中的性能分析

Fig. 9 Performance analysis of various algorithms in different model

表 6　各算法路径规划性能指标

Table 6　Path planning performance metrics of each algorithm

环境模型	指标	ISSA	MSNSSA	IBSO	MBFPA	SSA	DMSSSA
模型 1	最优值	162.19	177.21	200.13	260.61	170.80	125.26
	平均值	318.51	260.39	220.72	395.98	387.74	130.85
	方差	46.82	55.02	55.55	79.12	73.85	1.17
	平均用时/s	20.18	18.32	24.35	23.11	22.07	19.21
	有效路径率/%	100	100	100	100	100	100
模型 2	最优值	438.91	467.76	567.58	461.24	512.23	351.71
	平均值	537.18	543.25	814.23	574.71	606.85	400.11
	方差	46.50	66.45	77.26	72.26	78.22	39.23
	平均用时/s	25.31	26.01	31.22	28.65	29.18	26.12
	有效路径率/%	96	92	79	88	83	100
模型 3	最优值	570.45	463.70	511.56	497.64	578.13	500.34
	平均值	821.19	688.78	787.12	864.27	744.46	665.51
	方差	134.12	85.67	100.37	177.13	901.85	44.17
	平均用时/s	31.45	29.45	34.01	31.35	34.12	30.12
	有效路径率/%	73	86	75	71	81	100

变得更加陡峭,明显增加了路径规划的难度。在独立运行次数为 30 和迭代次数为 200 的条件下,仅有 DMSSSA 在 30 次测试中都规划出有效路径。在模型 2 中,DMSSSA 的平均路径长度相较于 ISSA、MSNSSA、IBSO、MBFPA 和 SSA 分别降低了 25.51%、26.34%、50.86%、33.72% 和 34.06%,当然,这种优越的性能是以更高的计算时间为代价,在计算时间上,DMSSSA 的平均用时位于 ISSA 和 MSNSSA 之后。在模型 3 中,所提算法的平均路径长度分别降低了 18.91%、3.38%、15.45%、23.01% 和 10.61%,有效路径率分别提高了 27%、14%、25%、29% 和 19%,在计算时间上,DMSSSA 位于 MSNSSA 之后,领先于 ISSA、IBSO、MBFPA 和 SSA。

综上所述,基于 *K*-means + + 聚类优化的动态多子群樽海鞘群算法在不同的环境模型下均有良好的鲁棒性,收敛速度快,寻优能力强,都能够以 100% 的效率规划出有效路径,面对复杂地形有更显著的优势。

5　结论

提出一种基于 *K*-means + + 聚类优化的动态多子群樽海鞘群算法用于求解无人机三维路径规划问题。首先结合 *K*-means + + 聚类算法设计动态多子群机制以平衡算法的全局搜索与局部开发,各子群结合多策略协同改进提高算法全局寻优能力同时避免算法陷入局部最优。其次,将 DMSSSA 放在 12 个 CEC2017 测试函数上测试其性能,实验证明了 DMSSSA 的寻优能力优于其他对比算法;最后将该算法应用在 3 个不同复杂性的三维环境模型中,对最优路径规划问题进行求解。仿真实验结果表明,在 3 种不同的环境模型下,DMSSSA 都能高效地规划出有效路径,并且平均有效路径率相比于 ISSA、MSNSSA、IBSO、MBFPA 和 SSA 提高了 15.5%、11%、23%、20.5% 和 18%。在 3 种环境模型中的平均规划路径长度最小,在复杂模型规划的路径长度分别比其他算法降低了 22.21%、14.86%、33.15%、28.36% 和 22.33%,从而证明了 DMSSSA 在无人机三维路径规划问题中的有效性。尽管 DMSSSA 在算法设计和路径规划上有较好的性能,但是也存在一些局限,当问题的复杂性和维度增加时,DMSSSA 会增加算法的计算复杂性和计算时间。因此在未来的工作中将尝试使用分布式并行计算方法来减少路径规划问题的计算时间,并引入更多的约束条件,比如动态障碍物约束和飞行速度约束,以更好地适应和满足无人机的特殊作业需求。

参　考　文　献

[1] Yahia H S, Mohammed A S. Path planning optimization in unmanned aerial vehicles using meta-heuristic algorithms: a systematic review[J]. Environmental Monitoring and Assessment, 2023, 195(1): 30.

[2] Khan M T R, Muhammad S M, Ru Y, et al. Aspects of unmanned aerial vehicles path planning: overview and applications[J]. International Journal of Communication Systems, 2021, 34(10): e4827.

[3] Poudel S, Arafat M Y, Moh S. Bio-inspired optimization-based path planning algorithms in unmanned aerial vehicles: a survey[J]. Sensors, 2023, 23(6): 3051.

[4] Baek J, Han S I, Han Y. Energy-efficient UAV routing for wireless sensor networks[J]. IEEE Trans on Vehicular Technology, 2019, 69(2): 1741-1750.

[5] Kiani F, Seyyedabbasi A, Aliyev R, et al. Adapted-RRT: novel hybrid method to solve three-dimensional path planning problem using sampling and metaheuristic-based algorithms[J]. Neural

Computing and Applications, 2021, 33(22): 15569-15599.
[6] Woods A C, La H M. A novel potential field controller for use on aerial robots[J]. IEEE Trans on Systems Man and Cybernetics: Systems, 2017, 49(4): 665-676.
[7] Wang H W, Qi X Y, Lou S J, et al. An efficient and robust improved A* algorithm for path planning[J]. Symmetry, 2021, 13(11): 2213.
[8] Silva A J, Silva A M, Motta T C F, et al. Heuristic and genetic algorithm approaches for UAV path planning under critical situation [J]. International Journal on Artificial Intelligence Tools, 2017, 26(1): 1760008.
[9] Wang D S, Tan D P, Liu L. Particle swarm optimization algorithm: an overview[J]. Soft Computing, 2018, 22: 387-408.
[10] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69: 46-61.
[11] Arora S, Singh S. Butterfly optimization algorithm: a novel approach for global optimization[J]. Soft Computing, 2019, 23: 715-734.
[12] Mirjalili S, Gandomi A H, Mirjalili S Z, et al. Salp swarm algorithm: a bio-inspired optimizer for engineering design problems [J]. Advances in Engineering Software, 2017, 114: 163-191.
[13] Zhang J, Wang J S. Improved salp swarm algorithm based on Levy flight and sine cosine operator[J]. IEEE Access, 2020, 8: 99740-99771.
[14] Tubishat M, Idris N, Shuib L, et al. Improved salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection[J]. Expert Systems with Applications, 2020, 145: 113122.
[15] Ahmed M, Kamel S H, Abbasy N H, et al. A gaussian random walk salp swarm algorithm for optimal dynamic charging of electric vehicles[J]. Applied Soft Computing, 2023, 147: 110838.
[16] 邢致恺, 贾鹤鸣, 宋文龙. 基于莱维飞行樽海鞘群优化算法的多阈值图像分割[J]. 自动化学报, 2021, 47(2): 363-377.
Xing Zhikai, Jia Heming, Song Wenlong. Levy flight trajectory-based salp swarm algorithm for multilevel thresholding image segmentation[J]. Acta Automatica Sinica, 2021, 47(2): 363-377.
[17] Gupta H, Verma O P. A novel hybrid coyote-particle swarm optimization algorithm for three-dimensional constrained trajectory planning of unmanned serial vehicle[J]. Applied Soft Computing, 2023, 147: 110776.
[18] 陈忠云, 张达敏, 辛梓芸. 多子群的共生非均匀高斯变异樽海鞘群算法[J]. 自动化学报, 2022, 48(5): 1307-1317.
Chen Zhongyun, Zhang Damin, Xin Ziyun. Multi-subpopulation based symbiosis and non-uniform Gaussian mutation salp swarm algorithm[J]. Acta Automatica Sinica, 2022, 48(5): 1307-1317.
[19] 李大海,李鑫, 王振东. 融合小生境机制的增强麻雀搜索算法及其应用[J]. 计算机应用研究 2024, 41(4): 1077-1085.
Li Dahai, Li Xin, Wang Zhendong. Enhanced sparrow search algorithm by integrating niche mechanism and its application[J]. Application Research of Computers, 2024, 41(4): 1077-1085.
[20] Arthur D, Vassilvitskii S. *K*-means++: the advantages of careful seeding[C]//In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. Philadelphia: SIAM, 2007, 7: 1027-1035.
[21] 张达敏, 陈忠云, 辛梓芸, 等. 基于疯狂自适应的樽海鞘群算法[J]. 控制与决策, 2020, 35(9): 2112-2120.
Zhang Damin, Chen Zhongyun, Xin Ziyun, et al. Salp swarm algorithm based on craziness and adaptive[J]. Control and Decision, 2020, 35(9): 2112-2120.
[22] 白钰, 彭珍瑞. 基于自适应惯性权重的樽海鞘群算法[J]. 控制与决策, 2022, 37(1): 237-246.
Bai Yu, Peng Zhenrui. Salp swarm algorithm based on adaptive inertia weight[J]. Control and Decision, 2022, 37(1): 237-246.
[23] Tudose A M, Picioroaga I I, Sidea D O, et al. Solving single-and multi-objective optimal reactive power dispatch problems using an improved salp swarm algorithm [J]. Energies, 2021, 14(5): 1222.
[24] 黎观锋, 梁志坚, 杨武. 基于改进二进制蛇优化算法的配电网故障定位[J]. 科学技术与工程, 2024, 24(18): 7710-7718.
Li Guanfeng, Liang Zhijian, Yang Wu. Fault location in distribution networks based on an improved binary snake optimization algorithm [J]. Science Technology and Engineering, 2024, 24(18): 7710-7718.
[25] Wang Z, Luo Q, Zhou Y. Hybrid metaheuristic algorithm using butterfly and flower pollination base on mutualism mechanism for global optimization problems[J]. Engineering with Computers, 2021, 37: 3665-3698.
[26] 张新明, 姜云, 刘尚旺, 等. 灰狼与郊狼混合优化算法及其聚类优[J]. 自动化学报, 2022, 48(11): 2757-2776.
Zhang Xinming, Jiang Yun, Liu Shangwang, et al. Hybrid coyote optimization algorithm with grey wolf optimizer and its application to clustering optimization [J]. Acta Automatica Sinica, 2022, 48(11): 2757-2776.