

基于蜚螳优化算法的水下机器人集群控制研究

张训源^{1,2,3} 李慧敏^{1,2,3} 王 鹏^{1,2}
ZHANG Xunyu LI Huimin WANG Peng

摘要

针对水下机器人集群在高动态、强干扰及通信受限环境中协同控制的挑战,文章提出了一种基于蜚螳优化算法(DBO)的分布式控制方法。通过模拟蜚螳群体的滚球、觅食、繁殖等生物行为,设计了融合水流补偿、信号丢失鲁棒性策略的改进算法,解决了编队形成与路径规划中的协同难题。此外,通过构建三维水流模型量化环境干扰,引入局部记忆缓存与自主探索策略应对通信不确定性,重构多模态避障成本函数提升复杂场景适应性。实验结果表明,相较于蜘蛛黄蜂算法(SWO)和中华穿山甲算法(CPO),DBO算法在收敛速度、全局搜索能力及综合成本优化上优势显著,为水下机器人集群在复杂海洋环境中的高效协同提供了仿生学解决方案,对提升其自主性与鲁棒性具有重要工程应用价值。

关键词

蜚螳优化算法; 集群控制; 路径规划

doi: 10.3969/j.issn.1672-9528.2025.10.046

0 引言

随着海洋资源勘探、环境监测以及国防安全等领域需求的不断增长,水下机器人集群技术已成为当前海洋工程与智能系统研究的热点方向。水下机器人集群通过多个体的协同作业,能够高效完成单一机器人难以胜任的复杂任务,如大面积区域搜索、目标围捕、协同勘探等。然而,水下环境具有高动态性、强干扰性和通信受限性等特点,海水的复杂流场、低能见度以及水声通信的时延和带宽限制,对集群的自主协同控制提出了严峻挑战。传统的集中式控制方法依赖全局信息,在通信中断或节点失效时鲁棒性不足;而基于经典智能算法的分布式控制策略,如粒子群优化、遗传算法等,虽在一定程度上提升了灵活性,但在动态环境中的适应性和实时性仍有待改进。自然界中,蜚螳群体展现出卓越的自组织协作能力,通过简单个体行为规则实现复杂任务的分工与协同,这为解决水下机器人集群控制难题提供了新的仿生学思路。借鉴蜚螳群体的协作机制,研究高效的集群控制算法,

对于提升水下机器人在复杂环境中的协同效率、鲁棒性和自主性具有重要的理论意义和工程应用价值。

诸多国内外学者针对水下机器人集群控制开展了大量研究,在协同围捕、路径规划、编队控制等关键技术取得了阶段性成果。胡桥等人^[1]在《AUV智能集群协同任务研究进展》中提出一种改进的人工势场法(HAP-IAPF),该方法通过设计目标引力函数与避障偏好策略,解决了有障碍水下弱信息环境中多AUV协同围捕问题,有效提升了集群避障路径的平滑性与最终队形的稳定性。李爱国等人^[2]提出一种并行双层结构的MAUV协同路径规划算法,该算法在子层采用粒子群优化(PSO)算法规划个体路径,主层结合差分进化(DE)算法进行路径组合优化,有效解决了多机器人路径规划中的碰撞问题与系统消耗最小化问题。苗润龙^[3]在《分布式无人艇集群协同区域搜索与目标定位研究》中提出基于Nash-BPSO的分布式全局协同搜索策略,通过引入纳什均衡概念与二进制粒子群优化算法,实现了USV集群在有限时间域内的同步并行迭代优化,提升了分区分配的均衡性与全局协同效率,仿真验证其在复杂岛礁环境中能有效降低巡航路径重复率并提高搜索效能。此外,施宇等人^[4]结合Glasius启发式神经网络(GBNN)与置信函数,提出非均匀多AUV动态围捕算法,该方法针对具有不同运动能力的AUV,在三维动态环境中实现了对逃逸目标的有效追捕,但受限于水下复杂环境的信息获取困难,神经网络模型的通用性仍需优化。这些研究虽在不同场景提升了集群性能,但针对水下通信受限、流场干扰等极端环境,仍缺乏对生物群体智能行为的深度仿生,尤其在自组织协作与实时决策机制上存在改进空间。

1. 威海海洋职业学院 山东威海 264300

2. 山东省船舶控制工程与智能系统工程技术研究中心
山东威海 264300

3. 威海市海洋大数据智能应用工程技术研究中心
山东威海 264300

[基金项目] 山东省船舶控制工程与智能系统工程技术研究中心科研开放专项资金项目支持(SSCC20240001);威海市海洋大数据智能应用工程技术研究中心科研开放专项资金项目支持(HYSJ20240003)

在此背景下,本文聚焦蜣螂群体协作行为的仿生建模,提出一种基于蜣螂算法的水下机器人集群控制框架。通过分析蜣螂个体间的信息交互与任务分工机制,设计分布式自组织算法,实现集群在编队形成、动态避障、路径规划等任务中的高效协同,为水下智能系统的协同控制提供新的仿生解决方案。

1 蜣螂算法的基本原理

蜣螂算法(dung beetle optimizer, DBO)是一种受蜣螂自然行为启发而提出的元启发式优化算法。在自然界中,蜣螂具有多种独特的行为,如滚球、跳舞、觅食、偷窃和繁殖等。这些行为可以帮助蜣螂在复杂的环境中生存和繁衍,同时也为算法设计提供了丰富的灵感。该算法通过模拟蜣螂的这些行为,在搜索空间中不断更新蜣螂个体的位置,以寻找最优解。每个蜣螂个体代表一个潜在的解,其适应度值反映了该解的优劣程度。算法通过迭代更新,不断优化蜣螂个体的位置,使得整个种群逐渐向最优解收敛。

1.1 滚球行为

蜣螂滚球是为了将食物(粪便球)搬运到安全的地方储存或繁殖。在算法中,滚球行为用于探索搜索空间,帮助蜣螂寻找更优的位置。位置更新公式为:

$$x_i(t+1) = x_i(t) + \alpha \times k \times x_i(t-1) + b \times \Delta x$$

$$\Delta x = |x_i(t) - X^o| \quad (1)$$

式中: t 表示当前迭代次数; $x_i(t)$ 表示第 i 只蜣螂在 t 时刻的位置; α 表示方向概率系数,值为 1 或 -1; $k \in (0,0.2]$, 表示扰动系数,控制步长; $b \in (0,1)$, 表示均匀分布随机数; X^o 表示全局最差位置; Δx 表示模拟光强变化对方向的影响。

1.2 跳舞行为

当遇到障碍物时,蜣螂会通过“跳舞”重新确定方向,位置更新公式为:

$$x_i(t+1) = x_i(t) + \tan(\theta) \times (x_i(t) - x_i(t-1)) \quad (2)$$

当蜣螂遇到障碍物无法前进时,需要通过跳舞来重新调整自己,以获得新的路线。 $\theta \in [0, \pi]$, 表示偏转角。该公式通过切线函数调整方向,模拟蜣螂重新规划路径的行为。

1.3 繁殖行为

在自然界中,雌性蜣螂将粪球运到适合产卵的安全地方并将其隐藏起来,以此为后代提供一个安全的环境。雌性蜣螂一旦确定了产卵区,就会选择在该区域育雏产卵。

$$L_b^* = \max(X^* \times (1-R), L_b) \quad (3)$$

$$U_b^* = \min(X^* \times (1+R), U_b) \quad (4)$$

式中: X^* 表示当前最优位置; L_b^* 、 U_b^* 表示动态产卵区上下

限; $T = 1-t/T_{\max}$ 表示随迭代次数递减的系数,其中 $T_{\max}$ 表示最大迭代次数; L_b 、 U_b 表示动态产卵区上下限。

1.4 觅食行为

模拟成熟小蜣螂寻找食物的行为,最佳觅食区居于全局最优位置。

觅食区边界定义:

$$L_b^b = \max(X^b \times (1-R), L_b) \quad (5)$$

$$U_b^b = \min(X^b \times (1+R), U_b) \quad (6)$$

式中: X^b 为全局最优位置; L_b^b 、 U_b^b 为动态觅食区上下限。

小蜣螂位置更新:

$$x_i(t+1) = x_i(t) + C_1 \times (x_i(t) - L_b^b) + C_2 \times (x_i(t) - U_b^b) \quad (7)$$

式中: C_1 表示服从正态分布的随机数; $C_2 \in (0,1)$, 表示均匀分布随机变量。通过向觅食区域边界靠拢,引导蜣螂探索高潜力区域。

1.5 偷窃行为

模拟盗贼蜣螂窃取其他蜣螂粪球的行为,位置更新基于全局最优解:

$$x_i(t+1) = X^b + S \times g \times (|x_i(t) - X^b| + |x_i(t) - X^b|) \quad (8)$$

式中: $x_i(t)$ 为盗贼蜣螂当前位置; S 为偷窃强度系数。通过结合自身与最优解的距离差,快速向优质区域移动,增强局部开发能力。

2 蜣螂算法在水下环境的优化改进

水下环境的强动态性、通信受限对蜣螂优化算法构成挑战。针对水下机器人集群控制需求,围绕水流干扰、信号丢失等核心问题展开算法改进。通过构建三维水流模型量化水流影响,设计融合水流补偿的位置更新公式;针对水下通信不确定性,提出局部记忆缓存与自主探索相结合的鲁棒性策略。改进策略旨在提升算法在复杂水下环境中的适应性与可靠性,为集群协同控制提供理论支撑。

2.1 水流建模

水下水流是影响水下机器人运动的核心环境因素,其速度场和方向场具有时空异质性。为准确描述水流对机器人的作用,构建三维水流模型用公式表示为:

$$V_c(x,y,z,t) = (v_{cx}, v_{cy}, v_{cz}) \quad (9)$$

该公式为水流速度场。其中, v_{cx} 、 v_{cy} 、 v_{cz} 分别为笛卡尔坐标系下 x 、 y 、 z 方向的水流速度分量。

水下机器人运动时受水流阻力作用,其合力可表示为:

$$F_d = -\frac{1}{2} \rho C_d A (\mathbf{v}_r - \mathbf{v}_c) \|\mathbf{v}_r - \mathbf{v}_c\| \quad (10)$$

式中： ρ 为海水密度； C_d 为阻力系数； A 为迎流面积； $\mathbf{v}_r$ 为机器人自身运动速度矢量。

该模型用于修正机器人实际运动速度，为位置更新提供动力学依据。

2.2 水流影响下的位置更新

在蜚螂算法的核心行为（如滚球、觅食、繁殖）中引入水流影响，通过修正位置更新公式实现对水流干扰的补偿：

（1）滚球行为修正：

生产者蜚螂的位置更新公式（原滚球行为）调整为：

$$X_i^{t+1} = pX_i^t + 0.3\|pX_i^t - \text{worse}^t\| \cdot \mathbf{d}_i + \alpha \cdot 0.1 \cdot X_i^t + \mathbf{v}_c^t \cdot \Delta t \quad (11)$$

式中： $\mathbf{d}_i$ 为当前位置到最差位置的单位方向矢量； $\mathbf{v}_c^t$ 为 t 时刻水流速度矢量； Δt 为时间步长； $\mathbf{v}_c^t \cdot \Delta t$ 直接反映水流对机器人位置的漂移作用。

（2）觅食行为修正：

跟随者蜚螂的位置更新引入水流速度补偿：

$$X_i^{t+1} = \text{best}X_i^t + \text{rand}(1, d) \cdot (pX_i^t - X_{\text{new}1}^{t+1}) + \text{rand}(1, d) \cdot (pX_i^t - X_{\text{new}2}^{t+1}) + \mathbf{v}_c^t \cdot \Delta t \quad (12)$$

式中： $X_{\text{new}1}^{t+1}$ 和 $X_{\text{new}2}^{t+1}$ 为跳舞行为生成的候选位置，通过叠加水流位移项确保位置更新的物理真实性。

2.3 信号丢失下的位置更新策略

水下声呐通信存在高延迟、低带宽和间歇性信号丢失问题，需设计鲁棒性位置更新策略以应对通信中断场景。

（1）局部记忆缓存机制

为每个机器人维护最近 T 步的历史位置和最优解缓存：

$$H_i = \{X_i^{t-k}, pX_i^{t-k} | k = 0, 1, \dots, T-1\} \quad (13)$$

当检测到信号丢失（即无法获取全局最优解 $\text{best}X^t$ 或邻居位置）时，切换至局部记忆驱动模式，利用历史最优解进行位置更新：

$$X_i^{t+1} = pX_i^t + \text{randn}(1, d) \cdot \frac{\|pX_i^t - \text{best}H_i^t\|}{2} \quad (14)$$

式中： $\text{best}H_i^t$ 为缓存中的局部历史最高位置。

（2）自主探索模式

在长时间信号丢失时，激活自主探索行为，模拟蜚螂的随机觅食机制：

$$X_i^{t+1} = X_i^t + \text{rand}(1, d) \cdot \sigma \cdot \mathbf{B} \quad (15)$$

式中： σ 为探索步长因子，随迭代次数递减以平衡全局探索与局部开发； $\mathbf{B}$ 为边界约束矩阵，确保位置处于可行域内。

3 实验

3.1 算法实验流程

在多水下机器人协同集群避障路径规划中，首先要构建全面且准确的模型。此模型需涵盖地形地貌、机器人的起始

点与目标点，以及区域内的障碍物分布等信息。地形信息借助数字高程模型（DEM）来表示，以网格形式呈现海底的起伏变化，为后续规划提供基础地理数据。障碍物信息则用几何图形（如圆形、矩形）来描述其位置、大小和范围。这些信息共同构成水下机器人的空间环境，为路径规划设定边界和约束条件

DBO 算法运行：

（1）种群初始化

按照区域的边界条件随机生成种群，每个个体的位置用向量 $\mathbf{X}_i = \{x_{i1}, x_{i2}, \dots, x_{id}\}$ 表示。其中 $i=1, 2, \dots, N$ ， N 是种群大小； d 为位置向量的维度。表示在初始阶段，算法在可行的搜索空间内随机撒点，这些点代表了机器人可能的初始路径状态。

（2）适应度函数设定

以路径、深度、威胁和转角成本作为适应度函数，综合评估每条路径的优劣。适应度函数值越低，表明该路径越优。通过不断优化适应度函数值，算法能够逐步找到最优路径。

（3）位置更新

按照蜚螂算法的规则，在每次迭代过程中更新种群中每个个体的位置。蜚螂算法模拟了蜚螂的觅食和繁殖行为，通过多种策略更新位置，包括探索和开发阶段。在探索阶段，算法在较大的搜索空间内寻找可能的最优解；在开发阶段，算法则聚焦于当前找到的较优解附近进行精细搜索。设第 t 次迭代时个体 X_i 的位置更新公式为 $X_i^{t+1} = f(X_i^t, X_{\text{best}}^t)$ ，其中 X_{best}^t 为第 t 次迭代时的最优个体位置。

（3）终止条件判断

当达到预设的最大迭代次数时，算法停止迭代，输出当前找到的最优解，即最优路径。

3.2 成本计算方法

路径规划的成本综合考虑了路径长度、深度、威胁以及转角等因素，具体计算方法如下：

（1）路径长度成本（ J_1 ）

设路径由 n 个离散点 $P_i = (x_i, y_i, z_i)$ ， $i=1, 2, \dots, n$ 组成。路径长度成本 J_1 计算路径中相邻点之间的欧几里得距离之和，公式为：

$$J_1 = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \quad (16)$$

路径长度成本反映了机器人的实际路程，较短的路径长度通常意味着更低的能耗和更短的时间。

（2）威胁成本（ J_2 ）

水下区域内存在多个威胁区域（障碍物），设第 j 个威胁区域的中心位置为：

$$C_j = (c_{j1}, c_{j2}, c_{j3}) \quad (17)$$

半径为 r_j ，对于路径上的每一段线段 $P_i P_{i+1}$ ，计算其与每个威胁区域的最小距离 d_{ij} 。威胁成本 J_2 的计算规则如下：

当 $d_{ij} > r_j + \delta$ (δ 为安全距离阈值) 时，该线段与威胁区域无碰撞，对 J_2 的贡献为 0；

当 $d_{ij} \leq r_j$ 时，发生碰撞，对 J_2 的贡献为一个很大的惩罚值 M ；

当 $r_j < d_{ij} \leq r_j + \delta$ 时，处于危险距离内，对 J_2 的贡献为 $k(r_j + \delta - d_{ij})$ ，其中 k 为系数。则 J_2 的计算公式为：

$$J_2 = \sum_{i=1}^{n-1} \sum_{j=1}^m \omega_{ij} \quad (18)$$

式中： m 为威胁区域的数量； ω_{ij} 为根据上述规则确定的该线段与第 j 个威胁区域的成本贡献。

威胁成本的引入确保了机器人能够有效避开障碍物和威胁区域，提高安全性。

(3) 深度成本

理想下潜深度设为 h_{ideal} 。对于路径上的每个点 P_i ，其深度为 z_i 。深度成本 J_3 计算机器人下潜深度与理想深度的偏差，公式为：

$$J_3 = \sum_{i=1}^n \begin{cases} M & z_i < 0 \\ |z_i - h_{ideal}| & z_i \geq 0 \end{cases} \quad (19)$$

如果机器人深度低于海底平面 ($z_i < 0$)，深度成本设为很大的惩罚值 M ；否则，计算深度与理想深度的绝对值偏差并累加。深度成本的设置有助于引导机器人保持在合适的下潜深度，避免过低或过高带来的风险。

(4) 转角成本

对于路径上相邻的 3 个点 P_{i-1} 、 P_i 和 P_{i+1} ，计算相邻线段 $P_{i-1}P_i$ 和 P_iP_{i+1} 之间的转角 θ_i 。设最大允许转角为 θ_{max} 。转角成本 J_4 的计算公式为：

$$J_4 = \sum_{i=2}^{n-1} \begin{cases} k_1(\theta_i - \theta_{max}), & \theta_i > \theta_{max} \\ 0, & \theta_i \leq \theta_{max} \end{cases} \quad (20)$$

式中： k_1 为系数。转角成本的考虑使得路径更加平滑，减少不必要的转弯，提高效率和舒适性。

综合各项成本，总成本 J 的计算公式为：

$$J = \omega_1 J_1 + \omega_2 J_2 + \omega_3 J_3 + \omega_4 J_4 \quad (21)$$

式中： ω_1 、 ω_2 、 ω_3 和 ω_4 分别为各项成本的权重系数，且 $\sum_{i=1}^4 \omega_i = 1$ 。通过调整这些权重系数，可以根据实际需求对不同成本因素进行权衡和优化。

3.3 实验结果与分析

为了验证 DBO 优化算法在水下机器人集群控制中的有效性，本研究选择了蜘蛛黄蜂优化算法 (spider wasp optimizer, SWO) 与中华穿山甲优化算法 (Chinese pangolin optimizer, CPO) 进行对比实验，结果如图 1~2 和表 1 所示。

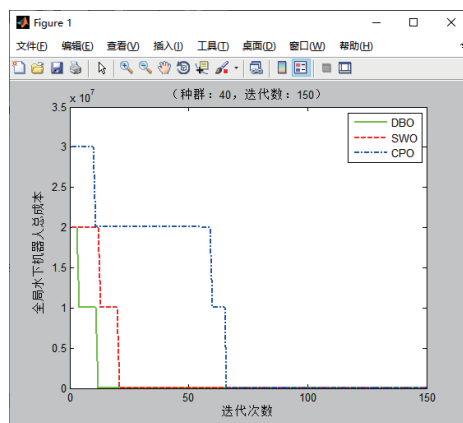


图 1 收敛曲线

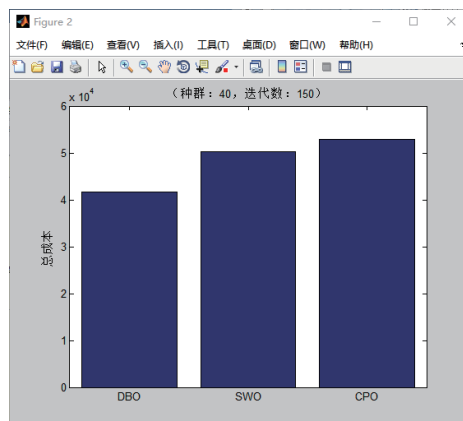


图 2 惩罚成本

表 1 DBO、CPO、SWO 结果对比

算法	收敛性	惩罚成本
DBO	模拟蜣螂滚球、觅食等行为，迭代初期迅速定位并深入较优解区域，高效探索使全局水下机器人总成本快速下降，展现出极快的收敛速度。	凭借高效且平衡的全局与局部搜索机制，有效规避陷入局部最优困境。在种群和迭代数既定情况下，精准规划出综合成本最低的水下机器人路径，优化效果显著。
CPO	起始阶段具备一定收敛能力，但在迭代过程中，因算法搜索机制不够稳定，易在不同解空间徘徊振荡，导致总成本起伏明显，稳定性方面存在欠缺。	受自身稳定性不足的影响，在趋近最优解的过程中难以精准收敛。
SWO	算法在搜索前期，对潜在优质解区域的探索效率较低，致使收敛曲线下降态势平缓。需历经更多迭代步骤，才能逐步靠近最优解，整体收敛进程相对迟缓。	搜索策略存在局限性，未能全面且深入地挖掘最优解空间，累积的成本相对较高。

实验结果表明，在多水下机器人协同集群避障路径规划研究中，DBO、CPO 和 SWO 算法展现出不同特性。DBO 算法模拟蜣螂行为，收敛速度极快，在迭代初期就能快速定位

基于 GQPSO 算法的光伏系统 MPPT 控制研究

齐云瑞¹ 周庆才¹
QI Yunrui ZHOU Qingcai

摘要

针对光伏组件在局部遮阴工况下输出功率-电压特性曲线存在多个极值点,传统最大功率点跟踪技术因全局搜索能力不足导致追踪失效的问题,文章提出了一种基于高斯概率分布的突变算子的量子粒子群算法用于光伏 MPPT 控制。该算法使用高斯变异算子代替随机序列,有效提升了量子粒子群算法的性能,防止其过早收敛到局部最优。实验对比分析表明,所提算法在追踪速度和准确度上有着显著优势。

关键词

光伏阵列;最大功率点追踪;局部遮阴;粒子群算法

doi: 10.3969/j.issn.1672-9528.2025.10.047

0 引言

作为可再生能源体系的重要组成部分,光伏发电技术近年来在能源结构转型中占据关键地位^[1]。研究表明,光伏阵列运行特性易受辐照强度、组件温度等环境参数波动影响,导致最大功率点呈现动态漂移特性^[2]。更为复杂的是,光伏单元固有的强非线性输出特性决定了需采用最大功率点跟踪技术实现有限光照条件下的能量捕获最优化^[3]。然而,经典控制策略如扰动观测法、电导增量法在遮蔽工况下的多峰寻优场景中易发生局部收敛现象^[4],这严重制约了复杂环境下

的发电效率提升。许多研究者针对此问题提出了多种改进算法,如改进人工鱼群算法^[5]、鲸鱼算法等控制方法^[6],但 these 方法计算资源消耗大、参数过多、调整复杂且较敏感,基于此,本文提出一种粒子群优化算法,该算法可以在局部遮阴的条件下快速而精准地跟踪到最大功率点,且收敛性好,鲁棒性强。

1 光伏阵列输出特性

1.1 光伏电池数学模型

基于工程应用需求,光伏发电系统普遍采用模块化设计理念,通过串并联拓扑架构将单体光伏单元集成为阵列化发电单元。该系统中各独立光伏元件的等效电路模型如图 1 所示。

1. 吉林建筑大学 吉林长春 130119

[基金项目] 吉林省科技计划项目(20240304183SF)

并深入较优解区域,有效平衡全局与局部搜索,优化效果显著,能精准规划出低成本路径。CPO 算法起始有一定收敛能力,但搜索机制不稳定,迭代中总成本波动大,稳定性欠佳,难以精准收敛到最优解。SWO 算法前期搜索优质解区域效率低,收敛曲线下降平缓,需更多迭代趋近最优,且因搜索策略局限,累积成本较高。综合来看,DBO 算法在收敛性和优化成本方面优势明显。

4 结语

本文针对水下机器人集群在复杂环境中的协同控制难题,提出一种基于蜣螂优化算法(DBO)的分布式控制框架。通过模拟蜣螂群体的滚球、觅食、繁殖等行为,结合水下环境特性改进水流补偿、信号丢失鲁棒性策略,有效提升了集群在动态环境中的路径规划效率与协同可靠性。实验结果表明,DBO 算法在收敛速度、全局搜索能力及综合成本优化上显著优于对比算法,为解决水下通信受限、流场干扰等挑战提供了新的仿生解决方案。

参考文献:

- [1] 胡桥,赵振轶,冯豪博,等. AUV 智能集群协同任务研究进展[J]. 水下无人系统学报,2023,31(2):189-200.
- [2] 李爱国,覃征,鲍复民,等. 粒子群优化算法[J]. 计算机工程与应用,2002,38(21):1-3.
- [3] 苗润龙. 分布式无人艇集群协同区域搜索与目标定位研究[D]. 哈尔滨:哈尔滨工程大学,2021.
- [4] 施宇. 复杂环境 AUV 集群自组织编队协同及避障控制[D]. 长春:吉林大学,2022.

【作者简介】

张训源(1989—),男,山东滨州人,硕士,讲师,研究方向:控制工程。

(收稿日期:2025-04-19 修回日期:2025-10-11)