

本文网址: <http://www.ship-research.com/cn/article/doi/10.19693/j.issn.1673-3185.03679>

期刊网址: www.ship-research.com

引用格式: 唐李军, 范云霞, 周星宇, 等. 基于改进麻雀搜索算法的 AUV 路径规划方法 [J]. 中国舰船研究, 2025, 20(3): 275–287.

TANG L J, FAN Y X, ZHOU X Y, et al. AUV path planning method based on improved sparrow search algorithm[J]. Chinese Journal of Ship Research, 2025, 20(3): 275–287 (in both Chinese and English).

基于改进麻雀搜索算法的 AUV 路径规划方法



扫码阅读全文

唐李军¹, 范云霞², 周星宇², 孙骞^{*2}

1 中国舰船研究设计中心, 湖北 武汉 430064

2 哈尔滨工程大学 信息与通信工程学院, 黑龙江 哈尔滨 150001

摘要: [目的] 针对复杂水下环境中自主水下航行器(AUV)三维路径规划算法存在的规划效果不理想、路径搜索不稳定等问题, 提出一种基于改进麻雀搜索算法的 AUV 路径规划方法。[方法] 推导评价区间响应的矢量分析方法公式, 引入分段学习和量子计算机制, 改进经典麻雀搜索算法的更新公式。通过汤普森采样策略动态更新种群数量。在复杂洋流环境中进行仿真测试, 验证改进算法的有效性。[结果] 测试结果表明, 改进算法的平均最长航行时间较改进前缩短 49.88%, 在极端突变的洋流环境下, 路径规划失败率降低 10.6%。[结论] 研究成果揭示了该方法具有较强的全局搜索能力和寻优性能、算法收敛性能较好, 具备高效的路径规划能力, 对 AUV 以及其他领域的路径规划问题有借鉴意义。

关键词: 自主水下航行器; 三维路径规划; 麻雀搜索算法; 区间优化; 矢量分析法; 运动规划

中图分类号: U675.73

文献标志码: A

DOI: 10.19693/j.issn.1673-3185.03679

0 引言

自主水下航行器(AUV)是一种能够自主执行水下任务的小型航行器^[1], 具有隐蔽性好、安全性高等优点, 在海洋勘探, 环境监测和水下救援等领域得到了广泛应用。

路径规划技术是 AUV 任务规划系统的关键技术之一^[2]。在 AUV 路径规划领域, 国内外学者提出了多种规划方法^[3], 主要可以分为传统经典算法和现代智能算法两大类。传统经典算法主要包括模拟退火算法、A*算法、人工势场法、遗传算法等。然而, 随着 AUV 作业环境的复杂性和不确定性不断增加, 传统经典算法在路径规划中容易陷入局部最优解。因此, 越来越多的学者将研究重点转向现代智能算法, 以提高 AUV 在复杂水下环境中的路径规划效率。具有代表性的现代智能算法包括粒子群优化算法(PSO)^[4]、差分进化算法(DE)^[5]、人工蜂群算法(ABC)^[6]、鲸鱼优化算法(WOA)^[7]、麻雀搜索算法(SSA)等^[8]。与其他

群智能算法相比, 麻雀搜索算法具有参数简单和收敛速度快等优点, 一经提出便受到了广泛关注。然而麻雀搜索算法也存在容易陷入局部最优、搜索精度较低、探索和开发能力不平衡等问题。为此, 众多研究者致力于对麻雀搜索算法进行改进。Lei 等^[9]在麻雀搜索算法的种群初始化中引入 Levy 飞行策略, 优化种群初始分布的不均匀性, 从而提高了种群的全局搜索能力。Song 等^[10]通过更新生产者麻雀在迭代过程中的位置, 引入非线性递减权重, 促进搜索空间的探索和利用, 有效提高了收敛速度和搜索精度。Liu 等^[11]提出一种麻雀搜索算法和仿生神经网络的融合算法, 结合两种方法的优点, 有效解决了复杂环境下的路径规划问题。Ouyang 等^[12]在麻雀搜索算法的跟随者阶段引入变螺旋搜索机制, 使算法的搜索范围更加细致, 提高了搜索精度。

尽管基于智能算法的 AUV 路径规划技术发展迅速, 但仍面临诸多挑战^[13]。首先, 在建模方面, 现有的路径规划方法通常将三维海洋环境简

收稿日期: 2023-12-08 修回日期: 2024-02-03 网络首发时间: 2024-07-02 11:04

基金项目: 国家自然科学基金面上项目资助(52271311)

作者简介: 唐李军, 男, 1984 年生, 博士

孙骞, 男, 1988 年生, 博士, 副教授。研究方向: 多源导航系统技术。E-mail: qsun@hrbeu.edu.cn

*通信作者: 孙骞

化为二维平面模型,这种简化在模型设计上存在一定局限性,无法充分满足 AUV 水下作业的实际需求;此外,现有的 AUV 三维路径规划算法在应对复杂水下环境时,存在计算复杂、路径规划效果不理想以及路径搜索不稳定等问题^[14]。针对上述问题,本文将从以下方面进行研究:

- 1) 改进适应度计算方法,采用矢量分析法以及基于可靠度的区间可能性度方法(RPDI)计算不确定洋流场下的适应度值,设计一种时间最短的路径规划算法,以提高计算效率。
- 2) 改进发现者更新过程,引入柯西-高斯机制平衡全局搜索和局部搜索的需求,使得算法在前期更新步长大,能够快速全局寻优,在后期更新步长小,有助于局部收敛。
- 3) 引入段学习机制和量子机制,产生新的个体,提高全局搜索能力。
- 4) 引入自适应选择机制,通过汤普森采样,更新种群数量,充分利用不同种群算子的优势,提高全局搜索和局部收敛的效果。

1 环境建模

1.1 水下地形环境

水下环境建模是 AUV 路径规划的前提,在进行水下地形建模时,以海底为零参考基准,分别以 ξ, η, ψ 作为纬度、经度和垂直高度,构建格栅化的三维地形建模。地形环境函数为

$$\psi(\xi, \eta) = \sum_a h_a \exp\left(-\left(\frac{\xi - \xi_a}{\xi_{sa}}\right)^2 - \left(\frac{\eta - \eta_a}{\eta_{sa}}\right)^2\right) \quad (1)$$

式中: A 为海山数量; (ξ_a, η_a) 为第 a 个海山坐标; h_a 为第 a 个海山高度; ξ_{sa} 为第 a 个海山的纬度衰减率; η_{sa} 为第 a 个海山的经度衰减率。

1.2 洋流环境

假设每个 AUV 均配备有水平声学多普勒水流剖面仪(H-ADCP),用于测量洋流速度。因此可以采用二维 Navier-Stokes 方程来近似描述在 t 时刻的洋流速度^[15]。由于垂直洋流相比于水平洋流对 AUV 的影响很小,在洋流环境建模中通常被忽略。因此,水平洋流速度的 x 轴分量 $V_{f,x}$ 和 y 轴分量 $V_{f,y}$ 如式(2)所示。

$$\begin{aligned} V_{f,x}(P_w(t)) &= -\frac{\Omega(y(t) - y_0)}{2\pi\|P_w(t) - P_0\|_2^2} \cdot \left(1 - e^{-\frac{\|P_w(t) - P_0\|_2^2}{r_f^2}}\right) \\ V_{f,y}(P_w(t)) &= \frac{\Omega(x(t) - x_0)}{2\pi\|P_w(t) - P_0\|_2^2} \cdot \left(1 - e^{-\frac{\|P_w(t) - P_0\|_2^2}{r_f^2}}\right) \end{aligned} \quad (2)$$

式中: Ω 为涡旋的强度; r_f 为涡旋的强半径; P_0 具体坐标为 (x_0, y_0) , 为涡旋的中心坐标; $P_w(t)$ 的具体坐标为 $(x(t), y(t))$, 是二维空间域。

1.3 约束条件

1.3.1 地形环境约束

AUV 航行路径应高于当前地形指定安全距离。根据不同地形栅格的缩放程度,安全距离可以按照相关规范进行调节。地形环境约束函数为

$$q_h = \begin{cases} 0, & h_b \geq \psi_b + d_{sf} \\ p_f \sum_{b=1}^{N_{ctrl}} (\psi_b + d_{sf} - h_b), & \text{其他} \end{cases} \quad (3)$$

式中: q_h 为地形高度适应度; N_{ctrl} 为控制点个数; h_b 为控制点航行高度; d_{sf} 为安全距离; p_f 为权重因子。

1.3.2 回转半径约束及俯仰角约束

在 AUV 进行航行工作时,其回转能力和俯仰行动能力受航行器限制。回转半径约束函数和俯仰角度约束函数分别如式(4)和式(5)所示。

$$q_r = \begin{cases} 0, & r_b \geq r_{sf} \\ p_f \sum_{b=1}^{N_{ctrl}-1} (r_{sf} - r_b), & \text{其他} \end{cases} \quad (4)$$

$$q_a = \begin{cases} p_f \sum_{b=1}^{N_{ctrl}} (\varphi_b - \varphi_{min}), & \varphi_b < \varphi_{min} \\ 0, & \varphi_{min} \leq \varphi_b \leq \varphi_{max} \\ p_f \sum_{b=1}^{N_{ctrl}} (\varphi_{max} - \varphi_b), & \varphi_b > \varphi_{max} \end{cases} \quad (5)$$

式中: q_r 为回转半径适应度; r_b 为对应控制点回转半径; r_{sf} 为最小回转半径; q_a 为俯仰角适应度; φ_b 为对应控制点的俯仰角度; $\varphi_{min}, \varphi_{max}$ 分别为角度限制的下界与上界。

式(3)~式(5)中的权重因子设为相同数值。这一决策能够确保在优化过程中,每个约束条件都被赋予相同的重要性,从而简化适应度函数的设计。同时,采用相同的权重因子也便于处理多目标约束问题,降低计算的复杂性。

2 区间优化

2.1 区间洋流建模

由于海洋是一个复杂的非线性系统,洋流行为受到多种因素的相互作用,包括风力、地球自转、水温、地形等,这些因素的复杂性使得洋流的准确预测十分困难,实际值与模拟值之间不可避免地存在差异。这些差异对非线性动力学的可预

测性造成了根本性的限制, 导致大多数洋流预测数据具有不确定性。这种不确定性使得从数据中估计正确的概率分布函数并计算最优路径或轨迹变得极为困难。因此, 洋流预测的不确定性是 AUV 在海洋环境下路径规划需要考虑的一个重要问题。区间优化是解决不确定性问题的一个重要解决方案, 通过对预测数据的上、下限进行建模和分析, 可以得到一个包含所有可能结果的区间, 然后通过优化这个区间来得到最优解。相比从数据中估计准确的概率分布函数, 估算海流矢量的两个边界比获取精确的海流或概率分布更容易, 计算上也更可行。通过构造区间洋流来表示不确定性, 区间洋流如式(6)和式(7)所示。

$$\tilde{v}_m(x, t) \in [\underline{v}_m(x, t), \bar{v}_m(x, t)] = [(1 - \sigma_m) \cdot v_m(x, t), (1 + \sigma_m) \cdot v_m(x, t)] \quad (6)$$

$$\tilde{v}_d(x, t) \in [\underline{v}_d(x, t), \bar{v}_d(x, t)] = [v_d(x, t) - \sigma_d, v_d(x, t) + \sigma_d] \quad (7)$$

式中: $\tilde{v}_m(x, t)$ 为预测的区间洋流速度; $\underline{v}_m(x, t)$ 为预测洋流速度的下限; $\bar{v}_m(x, t)$ 为预测洋流速度的上限; $v_m(x, t)$ 为确定性洋流速度; $\tilde{v}_d(x, t)$ 为预测的区间洋流方向; $\underline{v}_d(x, t)$ 为预测区间洋流方向的下限; $\bar{v}_d(x, t)$ 为预测区间洋流方向的上限; $v_d(x, t)$ 为确定性洋流方向; σ_m 为预测的洋流速度, 且 $\sigma_m < 1$; σ_d 为预测洋流方向不确定性的度量参数。

2.2 能量损失函数

当水下航行器尺寸小于流动结构尺寸时, 可以将水下航行器视为完整的运动模型, 即视为一个质点或一个刚体, 而忽略周围流体的细节。利用该模型, 可以使用数值方法进行计算和仿真, 得到的航行器对地速度 $V_g(x, t)$ 如式(8)所示。

$$V_g(x, t) = V_p(x, t) + V_f(x, t) \quad (8)$$

式中: $V_p(x, t)$ 为 AUV 的对水速度矢量; $V_f(x, t)$ 为洋流速度矢量。

AUV 的能量损失函数^[16] 如式(9)所示。

$$E(\gamma) = E_{\text{hotel}} + E_{\text{drag}} = \int_{t_s}^{t_g} [K_h + K_d |V_p(\gamma, t)|^3] dt \quad (9)$$

式中: $E(\gamma)$ 为 AUV 沿路径 γ 移动所消耗的能量; E_{hotel} 为 AUV 的基础运行能耗, 包括计算单元、传感器等非推进系统的固定能耗; E_{drag} 为克服流体施加的阻力所消耗的能量; t_s 为路径规划的开始时刻; t_g 为路径规划的结束时刻; K_h 和 K_d 是由航行器的设计决定的常数。

对于路径 γ 中的一小段路径 $[dx, dy, dz]$, 假设

它足够小, 可以近似看作一条直线, 且 AUV 的对水速度 V_p 在极小的路径段内可以视为一个定值, 则在时间 t 内遍历路径段 $[dx, dy, dz]$, 其能量损失函数 e 如式(10)所示。

$$e = (K_h + K_d |V_p|^3) \cdot t \quad (10)$$

2.3 区间响应

为计算 AUV 沿路径 γ 航行所消耗的总能量, 将路径 γ 视为由一组离散的控制点构成的连续样条曲线。相邻控制点之间的路径段构成路径序列 $K = [K_1 K_2, \dots, K_c K_{c+1}, \dots, K_\lambda K_{\lambda+1}]$, 其中 λ 表示路径段的数量。将每个路径段消耗的能量相加得到沿路径 γ 航行所消耗的总能量。在确定性洋流场中, 每个路径段消耗的能量都是确定的实数, 所以其消耗的总能量也是确定的实数。而在由区间洋流构成的不确定洋流场中, 洋流的大小和方向都是一个区间数, 由该洋流场计算得到的能量消耗 $e_c = [\underline{e}_c, \bar{e}_c]$ 也是一个区间数, 其中 $\underline{e}_c$ 和 $\bar{e}_c$ 分别表示路径段 c 在不确定流场中航行的最小能量损耗和最大能量损耗。因此, 在不确定流场中沿路径 γ 航行所消耗的总能量表示为

$$E = [E, \bar{E}] = \left[\sum_{c=1}^{\lambda} \underline{e}_c, \sum_{c=1}^{\lambda} \bar{e}_c \right] \quad (11)$$

式中, $[E, \bar{E}]$ 为区间响应。

2.4 矢量分析法

在求解能量损失函数时, 本文将航线路径 γ 细分成 λ 个路径段。当 λ 足够大时, 每个路径段都可以近似为一条直线, 且在该段路径中航行时间 t 极小。当时间 t 小于洋流变化率时, 相应的时变洋流速度 V_f 以及航行器对水速度 V_p 在极短时间之内可以视为常数, 且方向不变。由此作出 2 个假设:

假设 1: V_p 和 V_f 大小固定, θ 表示 V_g 和 V_f 之间的夹角, 对任意 θ_1 和 θ_2 , 如果 $\theta_1 < \theta_2$, 有 $|V_{g1}| > |V_{g2}|$ 。

假设 2: θ 和 V_p 大小固定。当 $0 \leq \theta \leq \pi/2$ 时, V_g 和 V_f 幅度之间的关系是一个凸函数, 当且仅当 V_f 垂直于 V_p 时, V_g 幅值最大; 当 $\pi/2 < \theta \leq \pi$ 时, V_g 与 V_f 幅度值之间的关系是单调递减。

根据该假设, 由式(10)得到的能量损耗 e_i 是一个与时间 t 呈线性关系的区间数。为求得能量损耗 e_i , 只需求得在路径段 $K_c K_{c+1}$ 内航行的最短时间以及最长时间。可以根据预航行方向和区间洋流边界来讨论区间洋流下路径段的区间响应^[17]。区间洋流下的最短及最长航行时间分类如图 1 所

示。其中, $K_c K_{c+1}$ 为预航行方向, $V_{f,\min}^*$ 和 $V_{f,\max}^*$ 分别为航行器航行时间最短和最长时对应的区间洋流速度。图 1(a)~图 1(b) 是预航行方向在区间洋流范围内的情况, 图 1(c)~图 1(f) 是预航行方向在区间洋流范围外的情况。

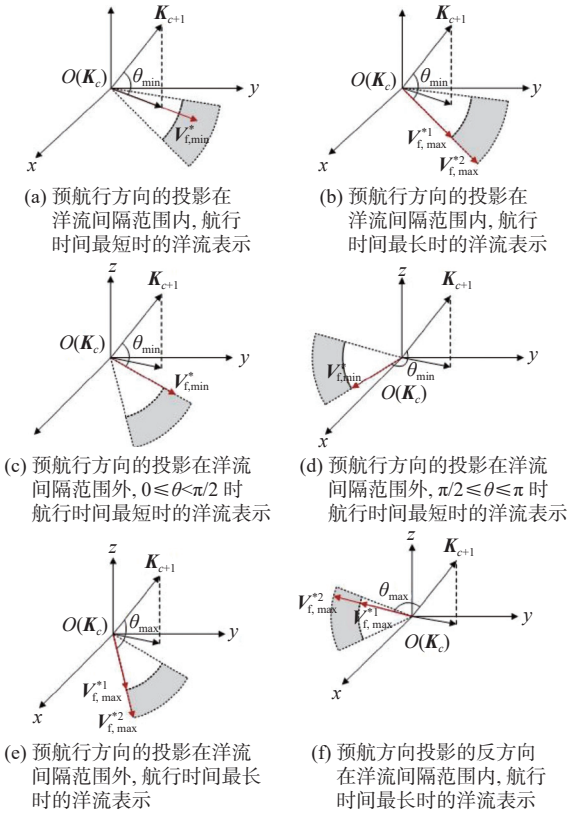


图 1 区间洋流下的最短及最长航行时间分类

Fig. 1 Classification of minimum and maximum navigation time under interval ocean currents

1) 预航行方向的投影在区间洋流范围内。

(1) 最短航行时间 t_c 。如图 1(a) 所示, 为使航行时间最短, 即让 V_g 幅值最大, 根据假设 1, 当 θ 最小时 V_g 幅值最大。由于预航行方向的投影在区间洋流范围内, 不难看出当洋流 V_f 的方向与预航行方向在水平面上的投影重合时, θ 最小, $\theta_{\min}$ 等于路径段 $K_c K_{c+1}$ 与其在水平面上投影之间的夹角。

由于 θ 等于路径段 $K_c K_{c+1}$ 与其在水平面上投影之间的夹角, 该夹角一定小于 $\pi/2$, 根据假设 2, $0 \leq \theta < \pi/2$ 时, V_g 和 V_f 之间幅度的关系是一个凸函数, 当且仅当 V_f 垂直于 V_p 时, $|V_g|$ 最大。当 V_f 垂直于 V_p 时, $V_f = |V_p| \cdot \cot\theta$, $|V_f|$ 值越接近 $|V_p| \cdot \cot\theta$ 时 $|V_g|$ 越大, t_c 越短, 因此, 当 $|V_p| \cdot \cot\theta$ 在区间 $[v_m(x, t), \bar{v}_m(x, t)]$ 内时, 时间最小时的洋流幅度 $v_{m,\min}^*$ 等于 $|V_p| \cdot \cot\theta$ 。当 $|V_p| \cdot \cot\theta$ 在 $[v_m(x, t), \bar{v}_m(x, t)]$ 外时, $v_{m,\min}^*$ 则选择最接近 $|V_p| \cdot \cot\theta$ 幅值边界。综上, $v_{m,\min}^*$ 如式(12)所示。

$$\begin{cases} v_{m,\min}^* = |V_p| \cdot \cot\theta, & v_m \leq |V_p| \cdot \cot\theta \leq \bar{v}_m \\ v_{m,\min}^* = v_m, & |V_p| \cdot \cot\theta < v_m \\ v_{m,\min}^* = \bar{v}_m, & |V_p| \cdot \cot\theta > \bar{v}_m \end{cases} \quad (12)$$

式中, $|V_p|$ 为固定常数。

(2) 最长航行时间 $\bar{t}_c$ 。如图 1(b) 所示, 为使时间最大, 即让 $|V_g|$ 最小, 根据假设 1, 当 θ 最大时 $|V_g|$ 最小。也就是在区间洋流范围内, 找到一个洋流方向, 使得该方向与预航行方向 $K_c K_{c+1}$ 之间的夹角最大。通过比较洋流方向的上界与预航行方向的夹角 ϕ_{ub} 和洋流方向的下界与预航行方向的夹角 ϕ_{lb} 两者的大小, 其中较大的夹角就是最大 θ 角, 则航行时间最大时洋流方向可以表示为 $v_{d,\max}^*$, 如式(13)所示。

$$\begin{cases} v_{d,\max}^* = \bar{v}_d, & \phi_{ub} \geq \phi_{lb} \\ v_{d,\max}^* = v_d, & \phi_{ub} < \phi_{lb} \end{cases} \quad (13)$$

根据假设 2, 当 $\pi/2 \leq \theta \leq \pi$ 时, V_g 与 V_f 之间的关系是单调递减的, 为使 V_g 最小, 即让 V_f 最大, 所以此时 $v_{m,\max}^*$ 应为洋流幅度的上界 $\bar{v}_m$; 而当 $0 \leq \theta < \pi/2$ 时, V_g 和 V_f 幅度之间的关系是一个凸函数, 所以, $|V_g|$ 的最小值一定出现在 $|V_d|$ 等于其区间上界或者下界时, 由此, 通过计算时间 $t(v_{d,\max}^*, v_m)$ 和时间 $t(v_{d,\max}^*, \bar{v}_m)$ 并比较, 选择使时间较大的幅值边界作为 $v_{m,\max}^*$ 。

2) 预航行方向的投影在区间洋流范围外。

(1) 最短航行时间 t_c 。如图 1(c)~图 1(d) 所示, 根据假设 1, 当 θ 最小时 $|V_g|$ 最大, 比较洋流方向的上下界与预航行方向的夹角 ϕ_{ub} 和 ϕ_{lb} 的大小, 选择较小的夹角作为最小 θ 角, 那么此时的角度边界即为时间最短时的洋流方向 $v_{d,\min}^*$, 如式(14)所示。

$$\begin{cases} v_{d,\min}^* = v_d, & \phi_{ub} \geq \phi_{lb} \\ v_{d,\min}^* = \bar{v}_d, & \phi_{ub} < \phi_{lb} \end{cases} \quad (14)$$

根据假设 2, 当 $0 \leq \theta < \pi/2$ 时, V_g 与 V_f 幅度之间的关系是一个凸函数, 且当且仅当 V_f 垂直于 V_p 时, $|V_g|$ 得到最大, 与情形 1) 类似, $v_{m,\min}^*$ 可由式(12)求得。另外, 当 $\pi/2 \leq \theta \leq \pi$ 时, V_g 与 V_f 幅度之间的关系是单调递减, 此时, $v_{m,\min}^*$ 应为洋流幅度的下界 v_m 。

(2) 最长航行时间 $\bar{t}_c$ 。如图 1(e) 所示, 根据假设 1, 角度 θ 越大, $|V_g|$ 越小, 则时间 t_i 最大。所以计算洋流方向的上下界与预航行方向的夹角 ϕ_{ub} 和 ϕ_{lb} 并比较, 选择较大的作为 θ 角, 根据式(13)求得其时间最大时的洋流方向 $v_{d,\max}^*$ 。角度确定后, 同样计算洋流幅度边界并比较, 在 $0 \leq \theta < \pi/2$ 时, 选

取幅度较小的夹角, 在 $\pi/2 \leq \theta \leq \pi$ 时, 选取幅度较大的夹角。

此外, 有一种特殊情况, 如图 1(f) 所示, 即预航行方向的投影的反方向在区间洋流范围内, 此时最大角度为航行方向与航行方向投影的反方向的夹角, 且该角度一定大于 $\pi/2$, 所以洋流幅度取上界即为时间最大。

通过上述矢量分析法, 可以确定时间最短和时间最大时的洋流方向以及幅值, 根据洋流矢量可以通过式 (15) 计算出 AUV 沿 $K_c K_{c+1}$ 航行的时间 t_c 。

$$\begin{cases} t_c = \frac{\sqrt{\Delta} - (\mathbf{V}_{fx} \cdot d_x + \mathbf{V}_{fy} \cdot d_y)}{|\mathbf{V}_p|^2 - |\mathbf{V}_f|^2}, & |\mathbf{V}_p| \neq |\mathbf{V}_f| \\ t_c = \frac{d_x^2 + d_y^2 + d_z^2}{2(\mathbf{V}_{fx} \cdot d_x + \mathbf{V}_{fy} \cdot d_y)}, & |\mathbf{V}_p| = |\mathbf{V}_f| \end{cases} \quad (15)$$

式中: $\Delta = (\mathbf{V}_{fx} \cdot d_x + \mathbf{V}_{fy} \cdot d_y)^2 - (|\mathbf{V}_f|^2 - |\mathbf{V}_p|^2) \cdot (d_x^2 + d_y^2 + d_z^2)$; $\mathbf{V}_{fx}$ 和 $\mathbf{V}_{fy}$ 分别为洋流 $\mathbf{V}_f$ 在水平方向 x 轴向和 y 轴向的分量; d_x , d_y , d_z 分别为路径段 $K_c K_{c+1}$ 在 x 轴、 y 轴和 z 轴方向上的投影。

由上述公式可以计算出航行过路径段 $K_c K_{c+1}$ 所需时间的区间数, 然后根据式 (10) 计算该路径段的区间响应, 最后根据式 (11) 计算沿路径 γ 航行的总能量区间数。

2.5 区间数比较

采用 RPDI^[18] 的区间比较关系对区间进行定量评价。与基于偏好的区间比较关系 (P-ICR) 和基于值的区间比较关系 (V-ICR) 相比, RPDI 的值域在整个实数域上, 更适用于工程可靠性分析和数值计算。RPDI 的核心公式为

$$P(E_A^L \leq E_B^L) = \frac{E_B^R - E_A^L}{2E_A^W + 2E_B^W} \quad (16)$$

式中: E_A 和 E_B 为 2 个区间数; E_A^L 和 E_A^R 分别为区间数 E_A 的左右边界; E_B^L 和 E_B^R 分别为区间数 E_B 的左右边界; E^W 为区间宽度; $P(E_A^L \leq E_B^L)$ 为 $E_A^L \leq E_B^L$ 的程度, 即当 $E_B^L \geq E_A^R$ 时, $P(E_A^L \leq E_B^L) \geq 1$, 上标 I 代表参考区间; 当 $E_B^R \leq E_A^L$ 时, $P(E_A^L \leq E_B^L) \leq 0$; 当 E_A^L 和 E_B^L 重合时, $0 < P(E_A^L \leq E_B^L) < 1$ 。

$$E^W = \frac{\bar{E} - E}{2} \quad (17)$$

在 RPDI 中, 必须引入一个参考区间 $V^I = [V^L, V^R]$, 然后利用式 (16) 化不确定区间数为一个确定的实值, 则基于区间分析的非线性优化如式 (18) 所示。

$$\begin{cases} \min_{\gamma} P(V^I \leq E^I(\gamma, \tilde{V}_f)) = \frac{E^R(\gamma, \tilde{V}_f) - V^L}{2E^W(\gamma, \tilde{V}_f) + 2V^W} \\ \text{s.t.} \quad \gamma \cap R_{\text{obstacle}} = \emptyset \\ \tilde{V}_f \in \tilde{V}_f^I = [\underline{V}_f, \overline{V}_f] \\ |\mathbf{V}_p| \leq V_{\max} \end{cases} \quad (18)$$

式中: R_{obstacle} 为障碍物的面积; $\tilde{V}_f$ 为由区间数表示的不确定洋流场; $V_{\max}$ 为 AUV 最大推进速度。

2.6 适应度函数

在路径规划和优化算法中, 适应度函数是关键的组成部分, 它评估了解在问题空间中的优劣。将地形环境的约束、俯仰角约束、回转半径约束、区间比较结果 P 以及 P 所对应的最小时间 $t_{\min}$ 进行加权, 得到适应度函数的计算公式。

对于地形环境的约束、俯仰角约束以及回转半径约束须保证其必要性, 所以利用数值较大的权重因子 p_f 使其满足必要性。而将区间比较结果 P 以及 P 所对应的最小时间 $t_{\min}$ 进行加权, 结合了路径的效率 (通过最小时间衡量) 和其实用性 (通过能量消耗区间比较结果 P 衡量)。这种综合评估使得路径规划不仅仅关注时间效率, 还考虑了能源效率, 对于能源有限的系统至关重要。整合所有约束条件, 得到适应度函数表达式, 如式 (19) 所示。

$$f = q_h + q_t + q_a + (1 - p_U)P + p_U \cdot t_{\min} \quad (19)$$

式中, p_U 的值为 0.6。

3 麻雀搜索算法优化

3.1 经典麻雀搜索算法

经典麻雀搜索算法是一种启发式算法, 其灵感来源于麻雀群体的觅食和反捕食行为。在麻雀的觅食过程中, 群体分为 2 部分: 生产者和加入者。生产者是整个麻雀群提供觅食区域和方向, 加入者利用生产者的位置获取食物。当麻雀群体检测到危险且警报值超过安全值时, 麻雀会采取行动对抗捕食者。

麻雀搜索算法流程如图 2 所示。

3.1.1 数学模型

麻雀种群初始位置可用 $(D \times O)$ 维矩阵描述, 如式 (20) 所示。

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,O} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,O} \\ \vdots & \vdots & \vdots & \vdots \\ x_{D,1} & x_{D,2} & \cdots & x_{D,O} \end{bmatrix} \quad (20)$$

式中: O 为种群个体数量; D 为解空间的维数。每个个体对应的适应度为

$$F_X = [f(X_1), f(X_2), \dots, f(X_O)] \quad (21)$$

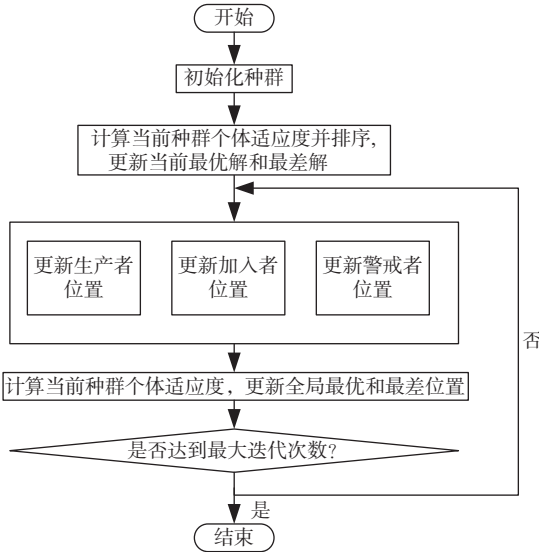


图2 麻雀搜索算法流程图

Fig. 2 Flowchart of the sparrow search algorithm

3.1.2 生产者的更新

生产者的主要任务是根据问题的特性和约束, 为整个算法提供初始的解决方案。生产者的更新公式为

$$X_{i,j}^{m+1} = \begin{cases} X_{i,j}^m \cdot \exp\left(\frac{-m}{\rho \cdot m_{\max}}\right), & R_2 < S_{\text{dange}} \\ X_{i,j}^m + p_g \cdot L, & R_2 \geq S_{\text{dange}} \end{cases} \quad (22)$$

式中: m 为当前迭代次数; $m_{\max}$ 为最大迭代次数; $X_{i,j}$ 为第 i 个麻雀个体在第 j 维中的位置信息; ρ 为在区间 $(0, 1]$ 中的一个随机数; $R_2 \in [0, 1]$, 为警戒值; S_{dange} 为安全阈值, 一般 $S_{\text{dange}} = 0.8$; p_g 为服从正态分布的随机数; L 为一个 $1 \times d$ 维度的矩阵, 且矩阵内每个元素均为 1。

3.1.3 加入者的更新

加入者根据生产者提供的解决方案进行优化和搜索, 旨在寻找更为优越的解决方案。加入者的更新公式为

$$X_{i,j}^{m+1} = \begin{cases} p_g \cdot \exp\left(\frac{X_{\text{worst}}^m - X_{i,j}^m}{i^2}\right), & i > N_d/2 \\ X_p^{m+1} + |X_{i,j}^m - X_p^{m+1}| \cdot Q^+ \cdot L, & \text{其他} \end{cases} \quad (23)$$

式中: X_p 为生产者中的最优解; X_{worst} 为全局历史最差解; N_d 为种群规模; Q 为 $1 \times d$ 的矩阵, 矩阵中的元素随机赋 1 或者 -1, 该矩阵指示搜索方向, 且 $Q^+ = Q^T(QQ^T)^{-1}$ 。

当 $i > N_d/2$ 时, 适应度较差的个体需要飞往其他方向进行解的搜寻。

3.1.4 警戒者的更新

警戒者负责监测环境并提供指导, 确保当遇到环境障碍时, 规划的路径不受局部最优解的限制, 能够调整策略以提高整体效果。警戒者的更新公式为

$$X_{i,j}^{m+1} = \begin{cases} X_{\text{best}}^m + \eta_r \cdot |X_{i,j}^m - X_{\text{best}}^m|, & f_i > f_g \\ X_{i,j}^m + p_s \cdot \left(\frac{|X_{i,j}^m - X_{\text{worst}}^m|}{(f_i - f_w) + p_o}\right), & f_i = f_g \end{cases} \quad (24)$$

式中: X_{best} 为当前全局最优解; η_r 为步长控制参数, 服从均值为 0、方差为 1 的正态分布的随机数; p_s 为在区间 $[-1, 1]$ 内的随机数; f_i 为当前麻雀个体的适应度值; f_g 和 f_w 分别为全局最优和最差适应度值, p_o 为很小的正数, 可以避免零除法误差。

3.2 麻雀搜索算法改进

3.2.1 分段学习机制

在复杂地形的实际场景中, 解空间往往存在大量局部最优解, 这给优化算法带来了挑战。为提升 SSA 算法的搜索能力, 引入分段学习机制。该机制通过分别构造边缘集合和精英集合, 使个体能够从不同来源(包括表现优异的个体或处于边缘位置的个体)学习有价值的信息, 从而产生新的个体, 指导种群的进化过程, 增强麻雀群体的多样性。

1) 边缘集合的构造。

边缘个体集合是通过计算当前个体 X_i^m 与其余个体 X_{rand}^m 之间的欧氏距离, 并通过排序选取距离最远的 20% 的个体构成的集合。

$$S_{\text{edge}}^m = \left\{ X_i^m \mid \text{rank}(|X_k^m - X_i^m|) < 0.2 \times O \right\} \quad (25)$$

式中: S_{edge}^m 为当前迭代中边缘个体集合; X_i^m 为当前个体的位置信息; X_k^m 为第 k 个个体的位置信息, 且 $1 \leq k \leq O$ 。

2) 精英集合的构造。

精英个体集合是通过利用适应度函数计算得到每个个体的适应度, 按升序或降序排列, 选出排名前 20% 的个体组成的集合。在算法迭代开始时, 精英个体之间的差距较为显著, 这在一定程度上可以保证种群的多样性。

$$S_{\text{elite}}^m = \left\{ X_i^m \mid \text{rank}(|F(X_k^m) - F(X_i^m)|) < 0.2 \times O \right\} \quad (26)$$

式中: S_{elite}^m 为当前迭代中精英个体集合; $F(X_i^m)$ 为当前个体的适应度; $F(X_k^m)$ 为第 k 个个体的适应度, 且 $1 \leq k \leq O$ 。

分段学习机制增加了个体从他人那里收集潜

在有用信息的概率。分段学习机制通过结合精英个体集和边缘个体集产生新的集合, 根据新的集合生成指导进化的虚拟个体。新集合 S_{obj} 的构造如式(27)所示, 虚拟个体的构造如式(28)所示。

$$S_{\text{obj}} = S_{\text{elite}} + S_{\text{edge}} \quad (27)$$

$$X^v(m) = X_{\text{obj}}(M_{\text{obj}}(m), m) \quad (28)$$

式中: $X^v(m)$ 为产生的虚拟个体的位置; $X_{\text{obj}}(m)$ 为属于 S_{obj} 个体的标签矩阵; $M_{\text{obj}}(m)$ 为由元素标签组成的列向量:

$$M_{\text{obj}}(m) = [r_{\text{obj},1}^m(m), \dots, r_{\text{obj},\text{seg}_o}^m(m), \dots, r_{\text{obj},\text{seg}_{S_N}}^m(m)] \quad (29)$$

式中: $M_{\text{obj}}(m)$ 为 S_N 个 $r_{\text{obj},\text{seg}_o}^m(t)$ 列向量的组合; S_N 为分段学习机制中的段数, $1 < o < S_N$; $r_{\text{obj},\text{seg}_m}^m(m)$ 为 $\text{seg}_{o+1} - \text{seg}_o + 1$ 维列向量, 且一定满足 $\text{seg}_{o+1} > \text{seg}_o$, seg_o 的构造是在 $[1, D]$ 区间内生成 S_N 个整数:

$$\text{seg}_o \in \text{Seg} = \{\text{seg}_o | \text{Randperm}(D, S_N)\} \quad (30)$$

式中: Seg 为抽象的段集合; seg 为集合中具体区间或段; Randperm 为随机排列函数, 用来生成一个包含整数的随机排列。

3.2.2 量子机制

量子计算机的最小信息单位是量子比特, 与经典计算机的二进制不同, 量子比特可以处于 0 和 1 的叠加态:

$$|p_\theta\rangle = p_\alpha|0\rangle + p_\beta|1\rangle \quad (31)$$

式中: $|\cdot\rangle$ 为狄拉克符号; p_α 和 p_β 为概率幅度, 且满足 $p_\alpha^2 + p_\beta^2 = 1$ 。

在经典坐标系中有三维坐标 $q_u = (q_{u1}, q_{u2}, q_{u3})$, 该坐标每个维度的范围均满足

$$q_{uv}^{\min} \leq q_{uv} \leq q_{uv}^{\max} \quad (32)$$

式中, $q_{uv}^{\min}$ 和 $q_{uv}^{\max}$ 分别为 q_{uv} 的下限和上限。

设与上述三维坐标对应的量子态为 $y_u = [y_{u1}, y_{u2}, y_{u3}]$, 且满足 $0 \leq y_{uv} \leq 1$, 则量子态与坐标的对应关系为

$$q_{uv} = y_{uv} \cdot (q_{uv}^{\max} - q_{uv}^{\min}) + q_{uv}^{\min} \quad (33)$$

SSA 搜索算法的收敛性能有待改进, 容易陷入局部最优解, 因此将量子计算理论与 SSA 搜索算法相结合, 将路径位置映射为量子比特, 通过量子旋转门的操作更新, 得到新的路径, 从而跳出局部最优解, 避免早熟, 加快算法的收敛速度。

麻雀种群个体数为 O , 最大迭代次数为 $m_{\max}$,

当迭代次数为 m 时, 第 i 个麻雀路径是 X_i^m :

$$X_i^m = [s_{i,1}^m, s_{i,2}^m, \dots, s_{i,j}^m, \dots, s_{i,D}^m] \quad (34)$$

式中: $s_{i,j}^m = [s_{i,j,u1}^m, s_{i,j,u2}^m, s_{i,j,u3}^m]$; 映射的量子位为 $w_{i,j}^m = [w_{i,j,u1}^m, w_{i,j,u2}^m, w_{i,j,u3}^m]^\circ$ 。

路径 X_i^m 映射的量子位可以表示为 $z_i^m = [w_{i,1}^m, w_{i,2}^m, \dots, w_{i,j}^m, \dots, w_{i,D}^m]$ 。第 i 个量子麻雀的第 j 维对应的量子旋转角如式(35)所示。

$$\cos(\varepsilon_{i,j,uv}^{m+1}) = \mu_s \times (w_{i,j,uv}^m - w_{\text{best},uv}^m) \quad (35)$$

式中: μ_s 为 $[0, 0.5]$ 区间的随机数。

经过量子旋转门更新, 得到的量子位为

$$w_{i,j,uv}^{m+1} = \text{abs}\left(w_{i,j,uv}^m \cos(\varepsilon_{i,j,uv}^{m+1}) + \sqrt{1 - (w_{i,j,uv}^m)^2} \sin(\varepsilon_{i,j,uv}^{m+1})\right) \quad (36)$$

$$w_{i,j,uv} = \frac{s_{i,j,uv} - s_{i,j,uv}^{\min}}{s_{i,j,uv}^{\max} - s_{i,j,uv}^{\min}} \quad (37)$$

最后, 将更新得到的量子位根据式(37), 映射回经典坐标系中, 获得新的路径, 计算得到的量子麻雀路径的适应度值, 更新种群中的全局最优路径。

3.2.3 柯西-高斯更新机制

在生产者的更新过程中需要平衡全局搜索和局部搜索的需求, 在更新前期, 步长大可以快速全局寻优, 在更新后期, 步长小有助于局部收敛。但在原本的 SSA 算法中, 将正态分布的随机数设置为更新步长。由于正态分布在概率中心的地方下降得相对较快, 从而可能跳过包含更优解的局部区域, 全局搜索能力受限。所以本文在生产者更新过程中引入柯西-高斯更新机制, 共同引导更新, 如式(38)所示。

$$X_{i,j}^{m+1} = \begin{cases} X_{i,j}^m \cdot \exp\left(\frac{-i}{\rho \cdot m_{\max}}\right), & R_2 < S_{\text{dange}} \\ X_{i,j}^m + (p_v \cdot p_a + (1 - p_v) \cdot p_g) L, & R_2 \geq S_{\text{dange}} \end{cases} \quad (38)$$

式中: p_a 为服从柯西分布的随机数; $p_v = e^{-5m/m_{\max}}$, 为随着迭代次数改变的加权因子。

柯西分布的概率密度函数在距离中心点较远的地方衰减速度较慢, 这一特性使其能够在搜索空间中进行更广泛的探索, 从而跳出局部最小值, 显著提升对全局最优解的搜索能力。在此基础上, 柯西-高斯更新机制通过平衡全局搜索和局部搜索的需求, 提高了算法的寻优性能。

3.2.4 自适应算子选择

在每一次迭代中, 传统 SSA 算法使用的种群数量及不同种群个体比例均保持固定。然而, 不

同种群算子在整个寻优过程中的作用存在差异,且在迭代的不同阶段,其重要性也有所不同。采用固定比例的种群分配方式可能导致计算资源浪费,同时影响全局搜索能力和局部收敛性。

为提高算法在前期搜索和局部收敛方面的性能,本文引入汤普森采样(TS)机制。这是一种基于贝叶斯推断的方法,根据每个动作的先验分布和观测数据来更新后验分布。然后,从后验分布中采样一个参数,再根据该参数选择最优的动作。这种机制可以使算法更灵活地分配资源,充分发挥不同种群算子的优势,从而提高全局搜索和局部收敛的效果。

自适应算子选择可以被视为一个多臂老虎机问题。在多臂老虎机问题中,伯努利分布的共轭先验为 Beta 分布,使得贝叶斯更新过程更为便捷。具体而言,每次选择推荐的种群(类似于选择一个老虎机)后,根据该种群是否成功进化(类似于老虎机是否吐出现)来更新 Beta 分布的参数。引入新参数后的 Beta 分布结合了此次经验,能够更准确地估计被选种群的成功率。

Beta 分布是(0,1)区间上的随机变量分布,由 2 个大于 0 的参数 α 和 β 决定,其概率密度函数如式(39)所示。

$$f_{Be}(\varpi;\alpha,\beta)=\frac{\varpi^{\alpha-1}(1-\varpi)^{\beta-1}}{B(\alpha,\beta)}\quad (39)$$

式中, $B(\alpha,\beta)$ 为归一化因子。

$$B(\alpha,\beta)=\int_0^1\varpi^{\alpha-1}(1-\varpi)^{\beta-1}d\varpi=\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}\quad (40)$$

式中: $\Gamma(\cdot)$ 为伽马函数; ϖ 为种群被选择的概率。

综上, Beta 分布接受 2 个参数进行伯努利试验,其试验成功的概率服从先验概率密度分布。这 2 个参数分别对应于重复试验之前已统计的成功次数和失败次数。根据试验结果,更新试验成功的后验概率分布,并进行归一化处理,采用赌轮盘的方式更新种群结构,提高全局收敛效果。

图 3 所示为引入多种改进机制得到的麻雀搜索算法(ISSA)流程。

4 仿真分析

4.1 环境参数配置

为验证改进的麻雀搜索算法在 AUV 路径规划中的性能优势,本文依据前文构建的模型环境,采用改进算法进行仿真实验。实验基本参数设置如表 1 所示。

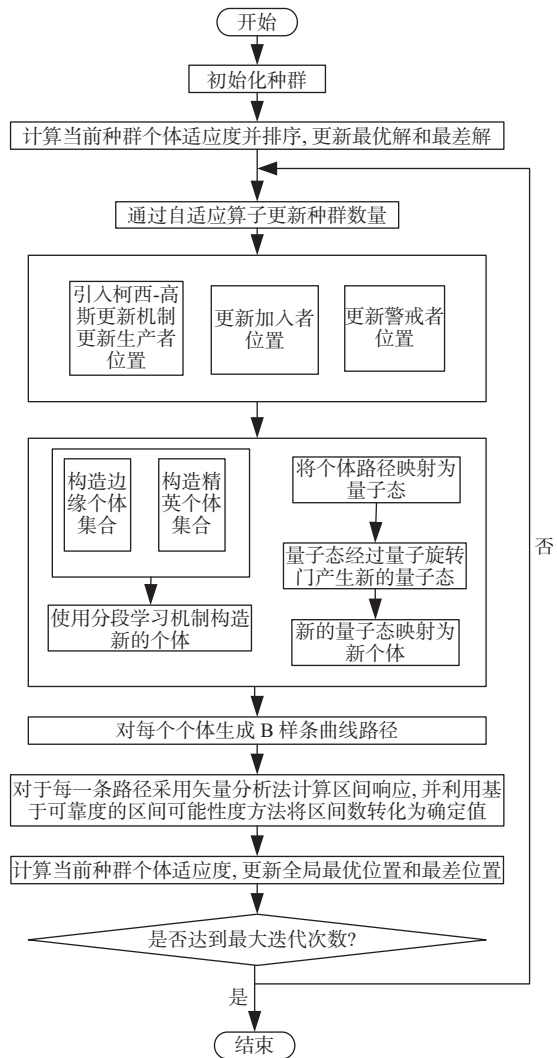


图 3 ISSA 算法流程图

Fig. 3 ISSA algorithm flowchart

表 1 仿真参数设置

Table 1 Simulation parameters

参数	数值
航行速度/(m·s ⁻¹)	1
最小回转半径/m	5
最小俯仰角/(°)	-20
最大俯仰角/(°)	20
安全距离/m	2
权重因子	10 ⁶
种群个数	30
控制点个数	10
最大迭代次数	100

4.2 仿真结果分析

为验证 ISSA 在 AUV 路径规划中的优越性,采用鲸鱼优化算法(WOA)、粒子群算法(POS)、差分优化算法(DE)、人工蜂群算法(ABC)与经典麻雀算法(SSA)与 ISSA 进行对比仿真。所有算

法均基于相同的环境条件和仿真参数进行测试。
图 4 和图 5 分别为采用上述 6 种算法在相同环境条件下进行路径规划得到的二维和三维最优规划路径。

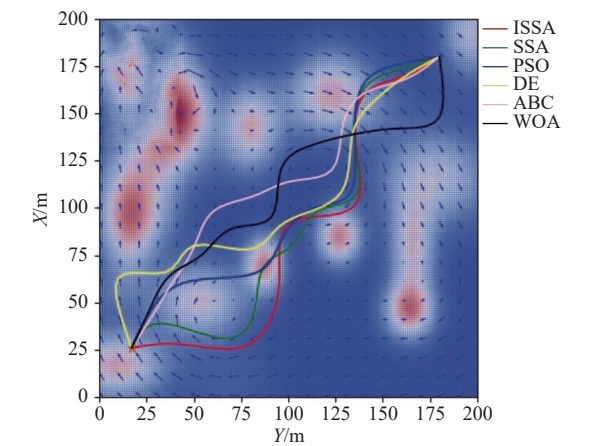


图 4 6 种搜索算法得到的二维最优路径
Fig. 4 Two-dimensional optimal paths for six algorithms

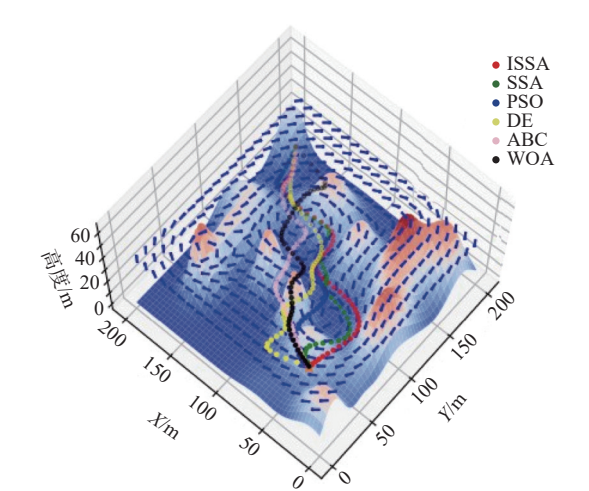


图 5 6 种搜索算法得到的三维最优路径
Fig. 5 Three-dimensional optimal paths for six algorithms

由图 4 和图 5 可见,在洋流场中,ISSA 和 SSA 算法能更好地利用洋流场的走势进行路径规划,而其他 4 种算法则更倾向于路径的短程特性,选择更近的路线。这表明麻雀搜索算法在路径规划中具有更好的智能性和鲁棒性,能够根据环境特征做出更有规划的决策。

图 6 为上述 6 种算法的适应度收敛速度对比结果。由图可见,在迭代前期,ISSA 算法的收敛速度最快,约迭代 20 次即可逼近 SSA 的最终结果;SSA 算法次之,约迭代 20 次后趋近于其他算法的最终结果;而 WOA,DE,PSO 和 ABC 算法的收敛速度较慢,需约迭代 40 次才能接近全局最优适应度。在迭代后期,所有算法均趋向全局最优适应度,但 ISSA 算法的适应度值最低,表明其性能最优。这说明 ISSA 算法不仅收敛速度快,且

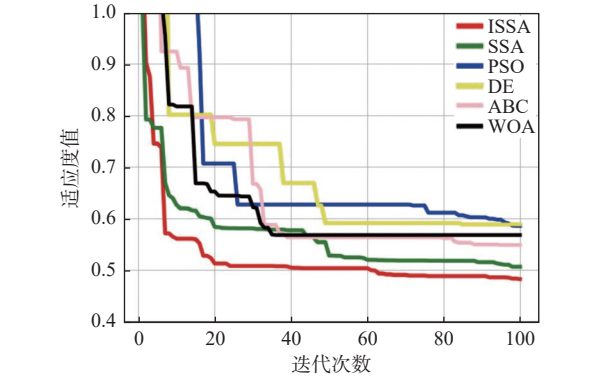


图 6 6 种算法的适应度收敛速度对比图
Fig. 6 Comparison chart of fitness convergence speed for six algorithms

全局搜索能力强,能够高效探索并找到潜在的全局最优解。

为验证分段学习机制和量子机制对全局搜索能力的影响,对不同群智能算法在多种地形环境中的路径规划结果进行评估,如表 2 和表 3 所示。由表可以看出,经过在 20 个随机环境中的 50 次重复实验,ISSA 算法在最短航行时间、最长航行时间和适应度方面均显著优于 PSO 和 SSA 算法。

表 2 不同环境下 AUV 最短/最长航行时间表
Table 2 AUV minimum/maximum voyage time in various environments

环境	最短航行时间/s			最长航行时间/s		
	PSO	SSA	ISSA	PSO	SSA	ISSA
1	532.0	504.3	414.9	1 401.1	849.6	637.5
2	494.9	344.1	345.8	1 165.2	599.9	546.6
3	577.1	310.0	286.4	1 927.1	560.0	636.4
4	506.5	318.9	303.9	1 522.3	743.1	645.5
5	504.9	393.8	293.9	2 119.7	632.4	628.4
6	748.2	524.4	461.3	3 915.8	1 018.8	967.5
7	522.4	350.5	353.8	1 324.6	661.5	609.5
8	409.9	252.2	173.1	1 946.5	776.7	715.0
9	524.0	280.8	320.2	2 833.1	725.8	675.8
10	329.4	285.1	271.9	763.7	609.0	537.2
11	727.4	521.2	451.2	2 509.7	1 082.2	1 015.2
12	482.5	332.4	322.6	1 469.2	607.9	560.8
13	586.0	297.6	235.4	3 566.5	752.7	705.1
14	686.8	386.0	351.5	3 496.2	676.5	970.5
15	534.5	435.9	295.2	2 715.2	785.9	604.5
16	327.7	322.9	203.0	1 977.5	798.6	603.4
17	505.8	330.7	347.3	1 120.2	665.3	602.2
18	395.9	305.6	294.6	1 224.6	615.8	631.6
19	788.3	566.8	474.1	3 775.7	1 238.3	771.8
20	386.9	320.5	328.4	1 224.1	648.0	643.1
均值	528.5	369.1	326.4	2 099.8	752.4	685.4

表 3 不同环境下 AUV 路径规划平均适应度表
Table 3 AUV path planning average fitness table in various environments

环境	适应度		
	PSO	SSA	ISSA
1	0.718	0.681	0.673
2	0.611	0.571	0.564
3	0.675	0.578	0.524
4	0.655	0.578	0.533
5	0.666	0.567	0.528
6	0.920	0.760	0.684
7	0.686	0.571	0.568
8	0.623	0.490	0.479
9	0.748	0.539	0.531
10	0.615	0.515	0.475
11	0.946	0.751	0.655
12	0.689	0.548	0.544
13	0.623	0.554	0.547
14	0.756	0.541	0.533
15	0.660	0.569	0.528
16	0.555	0.484	0.471
17	0.667	0.550	0.538
18	0.632	0.525	0.521
19	0.942	0.734	0.694
20	0.670	0.563	0.527
均值	0.703	0.583	0.556

具体而言,采用 ISSA 算法时,所需的最短航行时间相较于 PSO 和 SSA 算法分别减少了 38.2% 和 11.6%,最长航行时间则分别减少了 67.3% 和 8.9%。同时,其适应度也相对降低了 20.9% 和 4.7%。这些结果表明:ISSA 算法在路径规划任务中展现出更强的全局搜索能力,能够更有效地探索搜索空间并找到更优或接近于全局最优的解决方案。

为评估洋流的不确定性对 AUV 路径规划的影响,对多种不确定性参数条件下,不同群智能算法规划的路径进行系统评估。具体而言,采用蒙特卡洛方法进行重复实验,在洋流影响的区间内,分别对 50 次不同群智能算法的路径规划结果进行计算。在实验中,洋流方向服从 $N(\mu_d,\sigma_d)$,洋流幅值服从 $N(\mu_m,\sigma_m)$,随机生成 50 个洋流场。随后,对各算法生成的路径进行评估,相当于对一个算法生成的路径在一种不确定性参数下进行了 2 500 次评估。其中, μ_d 和 μ_m 分别为预测洋流的方向和速度, σ_d 和 σ_m 分别为洋流场方向和速度

的不确定值。不确定性参数与最短和最长航行时间的关系分别如表 4 和表 5 所示。

表 4 AUV 最短航行时间
Table 4 AUV minimum voyage time

不确定性参数		平均最短航行时间/s		
$\sigma_d/(^{\circ})$	$\sigma_m/(m\cdot s^{-1})$	PSO	SSA	ISSA
20	0.20	354.6	269.3	264.1
20	0.15	351.3	248.6	246.8
20	0.10	351.1	257.1	246.6
20	0.05	309.9	252.1	247.8
15	0.20	386.8	272.1	271.1
15	0.15	328.7	270.2	265.4
15	0.10	367.2	263.7	264.8
15	0.05	359.4	273.6	261.9
10	0.20	372.2	283.3	281.8
10	0.15	371.0	276.8	274.7
10	0.10	345.3	285.2	272.6
10	0.05	363.1	282.8	277.5
5	0.20	363.2	290.9	288.8
5	0.15	349.7	292.9	290.5
5	0.10	437.9	294.8	286.4
5	0.05	403.8	300.7	295.4
均值		363.4	275.88	271.01

表 5 AUV 最长航行时间
Table 5 AUV maximum voyage time table

不确定性参数		平均最长航行时间/s		
$\sigma_d/(^{\circ})$	$\sigma_m/(m\cdot s^{-1})$	PSO	SSA	ISSA
20	0.20	26 404.7	37 608.5	43 655.0
20	0.15	51 232.0	479 936.0	28 419.0
20	0.10	44 505.3	208 605.0	136 031.0
20	0.05	69 697 672.0	201 794.0	107 685.0
15	0.20	114 863 733.0	7 285.3	157 295.0
15	0.15	4 657.3	64 119.8	29 587.2
15	0.10	278 149.1	77 738.7	7 540.4
15	0.05	62 896.2	45 888.7	9 420.6
10	0.20	7 311.7	2 191.3	1 853.7
10	0.15	3 000.2	2 744.3	2 120.7
10	0.10	2 685.4	22 177.6	2 041.8
10	0.05	1 954.3	8 198.5	1 757.9
5	0.20	4 282.7	3 808.9	2 904.8
5	0.15	2 572.5	3 250.6	2 544.1
5	0.10	3 320.3	3 837.7	2 328.6
5	0.05	2 875.8	3 159.8	4 450.7
均值		11 566 078.4	73 271.6	3 3727.0

鉴于 RPDl 的比较结果是对区间的整体评估, 相较于区间下界(即最短航行时间)对评估结果的影响, 区间上界(即最长航行时间)的影响更为显著。因此, 在设置适应度函数时, 本文选择将区间响应结果与其最小时间进行加权处理, 赋予最短时间较大的权重, 为 0.6, 则最后生成的路径会优先考虑航行时间更短的路径。然而, 最长航行时间对路径选择的影响依然存在, 算法将倾向于选择航行时间较短且最长航行时间也较小的路径。

由表 4 和表 5 可以看出, ISSA 所选路径的平均最短时间比 SSA 少 4.87 s, 比 PSO 少 92.44 s, 平均最长时间相较于 SSA 优化了 53.9%。由此可见, 与其他算法相比, ISSA 的平均最短时间和平均最长时间均为最小值, 具有显著优势。

进一步分析表 6 和表 7, ISSA 的平均适应度比 SSA 低 2.1%, 比 PSO 低 23.5%。此外, ISSA 在极端环境下的失败次数比 SSA 低 10.6%, 比 PSO 低 69.2%。当不确定性参数较小时, ISSA 的规划路径甚至出现零失败的情况。这在现实场景中更具实际意义, 因为极端突变的洋流环境几乎不存在。然而, ISSA 的计算时间需要 71.28 s, 相比于 SSA 的 55.3 s 和 PSO 的 53.65 s, 计算时间明显上升。

表 6 AUV 路径规划平均适应度				
Table 6 Average fitness of AUV path planning				
不确定性参数		平均适应度		
$\sigma_d/(^{\circ})$	$\sigma_m/(m \cdot s^{-1})$	PSO	SSA	ISSA
20	0.20	0.9173	0.7248	0.7008
20	0.15	0.9079	0.6897	0.6838
20	0.10	0.9085	0.7087	0.6971
20	0.05	0.8151	0.6988	0.6941
15	0.20	0.9247	0.6805	0.6660
15	0.15	0.8100	0.6655	0.6540
15	0.10	0.8706	0.6537	0.6420
15	0.05	0.8163	0.6482	0.6296
10	0.20	0.9011	0.6697	0.6543
10	0.15	0.8585	0.6619	0.6481
10	0.10	0.7996	0.6519	0.6404
10	0.05	0.7890	0.6349	0.6207
5	0.20	0.8353	0.6751	0.6593
5	0.15	0.8002	0.6778	0.6624
5	0.10	0.9553	0.6528	0.6380
5	0.05	0.8384	0.6461	0.6275
均值		0.8592	0.6713	0.6574

表 7 AUV 路径规划失败次数分析结果				
Table 7 Count analysis result of AUV path planning failure				
不确定性参数		失败次数		
$\sigma_d/(^{\circ})$	$\sigma_m/(m \cdot s^{-1})$	PSO	SSA	ISSA
20	0.20	1 786	1 165	1 168
20	0.15	1 674	1 032	982
20	0.10	1 703	616	595
20	0.05	1 300	243	146
15	0.20	1 633	1 049	1 090
15	0.15	1 726	958	870
15	0.10	1 377	431	391
15	0.05	1 499	176	96
10	0.20	1 614	1 119	1 002
10	0.15	1 742	524	514
10	0.10	1 397	224	220
10	0.05	1 400	250	0
5	0.20	1 675	530	447
5	0.15	1 532	241	177
5	0.10	1 473	13	12
5	0.05	1 516	50	0
均值		1 566	539	482

5 结 语

针对不确定洋流场下的 AUV 路径规划问题, 本文提出了一种基于时间最短的路径规划算法。该方法通过区间洋流模型表示不确定洋流场, 利用矢量分析法计算区间响应, 并借助 RPDl 将区间响应转化为确定性数, 从而提高了算法的计算效率。此外, 本文引入了柯西-高斯机制、分段学习机制、量子机制以及自适应选择机制, 并通过汤普森采样动态更新种群数量。在不确定流场中的仿真测试表明, ISSA 相较于其他算法具有更强的寻优能力、更快的收敛速度和更高的安全性。该算法不仅改善了经典麻雀搜索算法容易陷入局部最优、搜索精度较低的缺点, 还有效解决了三维 AUV 路径规划在应对复杂水下环境时面临的计算复杂、路径规划效果不理想以及路径搜索不稳定等问题。

参考文献:

[1] 闫敬, 陈天明, 关新平, 等. 自主水下航行器协同控制研究现状与发展趋势 [J]. 水下无人系统学报, 2023, 31(1): 108–120.

YAN J, CHEN T M, GUAN X P, et al. Autonomous undersea vehicle cooperative control: current research status and development trends[J]. Journal of Unmanned

- [Undersea Systems](#), 2023, 31(1): 108–120 (in Chinese).
- [2] 刘祥, 叶晓明, 王泉斌, 等. 无人水面艇局部路径规划算法研究综述 [J]. [中国舰船研究](#), 2021, 16(增刊 1): 1–10.
LIU X, YE X M, WANG Q B, et al. Review on the research of local path planning algorithms for unmanned surface vehicles[J]. [Chinese Journal of Ship Research](#), 2021, 16(Supp 1): 1–10.
 - [3] 石晴晴, 牛文栋, 张润锋, 等. 水下滑翔机路径规划研究综述及展望 [J]. [中国舰船研究](#), 2023, 18(1): 29–42, 51.
SHI Q Q, NIU W D, ZHANG R F, et al. Review and prospects of underwater glider path planning[J]. [Chinese Journal of Ship Research](#), 2023, 18(1): 29–42, 51 (in Chinese).
 - [4] KENNEDY J, EBERHART R. Particle swarm optimization[C]//Proceedings of ICNN'95-International Conference on Neural Networks. Perth: IEEE, 1995, 4: 1942–1948.
 - [5] STORN R, PRICE K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces[J]. [Journal of Global Optimization](#), 1997, 11(4): 341–359.
 - [6] 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(3): 459–471.
 - [7] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. [Advances in Engineering Software](#), 2016, 95: 51–67.
 - [8] XUE J K, SHEN B. A novel swarm intelligence optimization approach: sparrow search algorithm[J]. [Systems Science & Control Engineering](#), 2020, 8(1): 22–34.
 - [9] LEI Y, DE G, FEI L. Improved sparrow search algorithm based DV-Hop localization in WSN[C]//2020 Chinese Automation Congress (CAC). Shanghai: IEEE, 2020: 2240–2244.
 - [10] SONG W, LIU S, WANG X C, et al. An improved sparrow search algorithm[C]//2020 IEEE Intl Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom). Elsevier: IEEE, 2020: 537–543.
 - [11] LIU Q L, ZHANG Y, LI M Q, et al. Multi-UAV path planning based on fusion of sparrow search algorithm and improved bioinspired neural network[J]. [IEEE Access](#), 2021, 9: 124670–124681.
 - [12] OUYANG C T, QIU Y X, ZHU D L. Adaptive spiral flying sparrow search algorithm[J]. [Scientific Programming](#), 2021, 2021: 6505253.
 - [13] 李文魁, 周铸, 宦爱奇, 等. 自主水下航行器自适应 S 面三维轨迹跟踪的仿真验证 [J]. [中国舰船研究](#), 2022, 17(4): 38–46, 91.
LI W K, ZHOU Z, HUAN A Q, et al. Simulation and verification of an adaptive S-plane three-dimensional trajectory tracking control for autonomous underwater vehicles[J]. [Chinese Journal of Ship Research](#), 2022, 17(4): 38–46, 91 (in Chinese).
 - [14] 姚金艺, 曾庆军, 赵强, 等. 全驱动型 AUV 三维路径跟踪控制系统设计及分析 [J]. [中国舰船研究](#), 2019, 14(6): 22–29.
YAO J Y, ZENG Q J, ZHAO Q, et al. Design and analysis of fully-actuated AUV's three-dimensional path tracking control system[J]. [Chinese Journal of Ship Research](#), 2019, 14(6): 22–29 (in Chinese).
 - [15] HU J N, CHEN C, CAI L, et al. UAV-assisted vehicular edge computing for the 6G internet of vehicles: Architecture, intelligence, and challenges[J]. [IEEE Communications Standards Magazine](#), 2021, 5(2): 12–18.
 - [16] KULARATNE D, BHATTACHARYA S, ANI HSIEH M. Going with the flow: a graph based approach to optimal path planning in general flows[J]. [Autonomous Robots](#), 2018, 42(7): 1369–1387.
 - [17] YAO X L, WANG F, YUAN C S, et al. Path planning for autonomous underwater vehicles based on interval optimization in uncertain flow fields[J]. [Ocean Engineering](#), 2021, 234: 108675.
 - [18] JIANG C, HAN X, LI D. A new interval comparison relation and application in interval number programming for uncertain problems[J]. [Computers, Materials & Continua](#), 2012, 27(3): 275–304.

AUV path planning method based on improved sparrow search algorithm

TANG Lijun¹, FAN Yunxia², ZHOU Xingyu², SUN Qian²

1 China Ship Development and Design Center, Wuhan 430064, China

2 College of Information and Communication Engineering, Harbin Engineering University,
Harbin 150001, China

Abstract: [**Objective**] To address the challenges of complex underwater environments, particularly the uncertainties in ocean currents, this study proposes an improved sparrow search algorithm (ISSA) for autonomous underwater vehicles (AUVs) path planning. The goal is to enhance the efficiency and robustness of path planning by minimizing navigation time and improving path stability in uncertain conditions. [**Method**] The proposed ISSA incorporates several key enhancements to the classic sparrow search algorithm (SSA). First, a vector analysis method is developed to evaluate interval responses, allowing the algorithm to effectively handle uncertainties in ocean currents. By modeling the uncertain ocean currents as intervals, the algorithm can accurately calculate the energy consumption and navigation times for different paths. Second, the ISSA introduces segmented learning and quantum mechanisms to improve global search capabilities. These mechanisms enable the algorithm to dynamically adjust its search strategy by learning from both elite and marginal individuals within the population, thereby enhancing diversity and preventing premature convergence. Third, a Cauchy-Gaussian mechanism is integrated into the update formula to balance global exploration and local exploitation during the search process. Finally, the population size is dynamically updated using Thompson sampling, allowing the algorithm to adaptively allocate computational resources based on the complexity of the environment. [**Results**] Simulation results demonstrate that the ISSA significantly outperforms the original SSA and other state-of-the-art algorithms, such as particle swarm optimization (PSO), differential evolution (DE), artificial bee colony (ABC), and whale optimization algorithm (WOA). Specifically, the ISSA reduces the average maximum navigation time by 49.88% compared to the original SSA and decreases the failure rate in extreme ocean current conditions by 10.6%. The ISSA also exhibits superior convergence speed, achieving near-optimal paths in approximately 20 iterations, while other algorithms require around 40 iterations to approach the global optimum. Moreover, the ISSA shows a lower average fitness value, indicating better optimization performance and path planning efficiency. [**Conclusion**] The ISSA demonstrates strong global search capabilities and robustness in dynamic and uncertain underwater environments, making it a promising solution for AUV path planning. The improvements in convergence characteristics and the ability to handle complex ocean currents highlight the algorithm's potential for practical applications. Future work will focus on further optimizing the computational efficiency of ISSA and exploring its application in more diverse and challenging underwater scenarios.

Key words: autonomous underwater vehicles; three-dimensional path planning; sparrow search algorithm; interval optimization; vector analysis method; motion planning