

混合多策略改进蜣螂算法的避障路径规划^{*}

万怡华 张雪梅

(宁夏师范学院物理与电子信息工程学院 宁夏 756000)

摘要:为了实现移动机器人在复杂环境中路径规划的高效搜索能力,提出了一种混合多策略的改进蜣螂算法。首先,引入改进ISPM混沌策略用于初始化蜣螂的初始种群,使初始总体分布更均匀,并降低算法落入局部最优解的可能性。然后,将贪婪选择策略与改进透镜成像反向学习策略相结合,改进蜣螂觅食行为的位置更新,平衡算法的局部开发和全局搜索能力,提高算法的收敛能力;最后,利用莱维飞行策略并加入改进动态权重更新方式,改进蜣螂偷窃行为的位置更新,改变最优全局解,防止算法陷入局部最优。为了验证改进算法的性能,采用基本测试函数和路径优化方面仿真,将改进后算法与其他四种群体智能算法进行了比较。实验结果表明,改进的蜣螂优化算法显著提高了收敛速度和优化精度,具有良好的鲁棒性。

关键词: 蜣螂算法;路径规划;混沌映射;莱维飞行

中图分类号: TP18;TP242 文献标识码: A 国家标准学科分类代码: 520.2099

Obstacle avoidance path planning of hybrid multi-strategy improved dung beetle optimizer

Wan Yihua Zhang Xuemei

(School of Physical and Electronic Information Engineering, Ningxia Normal University, Ningxia 756000, China)

Abstract: In order to achieve efficient search capability for path planning of mobile robots in complex environments, a hybrid multi-strategy improved dung beetle optimizer has been proposed. Firstly, the ISPM chaos strategy is introduced to initialize the initial population of fireflies. This ensures a more uniform distribution of the initial population and reduces the likelihood of the algorithm getting stuck in local optima. Then, the greedy selection strategy is combined with the improved lens imaging reverse learning strategy to update the positions of the fireflies during their foraging behavior. This balances the algorithm's local exploration and global search capabilities, thereby enhancing its convergence ability. Finally, the Levy flight strategy and an improved dynamic weight update mechanism are employed to update the positions of the fireflies during their stealing behavior. This helps to change the optimal global solution and prevent the algorithm from getting trapped in local optima. To evaluate the performance of the improved algorithm, comparative experiments are conducted with four other swarm intelligence algorithms using benchmark test functions and simulation of path optimization. The experimental results demonstrate that the improved dung beetle optimizer significantly improves convergence speed and optimization accuracy, while maintaining good robustness.

Keywords: dung beetle optimizer; path planning; chaotic mapping; Levy flight

0 引言

人工智能是解决各个领域问题的最有前途的技术之一。机器人是人工智能技术中一种灵活、低成本、高效率的有力辅助工具,在各种场景中具有广泛的应用^[1]。其中,路径搜索作为机器人运行的支撑技术,具有重要的研究意义。许多研究人员一直在尝试开发路径规划问题的解决方案,

除了传统方法,如人工势场算法(artificial potential field, APF)^[2]、模糊逻辑算法^[3]、快速探索随机树(rapid-exploration random tree, RRT)^[4]等方法之外,研究者提出了基于自然现象或规律的元启发式算法来求解路径规划问题,如蚁群优化算法(ant colony optimization, ACO)^[5]、粒子群优化算法(particle swarm optimization, PSO)^[6]、麻雀算法(sparrow search algorithm, SSA)^[7]、狮群优化算法

(loin swarm optimization, LSO)^[8] 等。然而, 目前还没有一种方法可以完全解决机器人路径规划问题。因此, 近年来, 研究人员一直在不断寻求新的更有效和高效的解决方案或改进现有的方法。

蜣螂算法(dung beetle optimizer, DBO)是一种新型的智能优化算法^[9], 广泛应用于许多重要领域, 它主要通过模仿蜣螂的生物学行为实现优化。与 PSO、ACO 等智能优化算法相比, DBO 具有参数少、原理简单、实现方便等优点。但 DBO 存在收敛速度慢、易受局部最优影响、精度差和稳定性有限等缺点。为此, 许多学者做出了许多改进。周亚中等^[10]通过在蜣螂优化算法中引入周期突变机制以提升算法的寻优能力, 并利用改进后的蜣螂优化算法对差分整合移动平均自回归模型参数 p 和 q 进行定阶, 实现对变压器振动信号的预测; 甄然等^[11]提出了一种改进蜣螂算法的航迹规划方法, 引入 Levy 策略与一种具有策略自适应的横向交叉, 获得更可行、更高效的路径; Alamgeer 等^[12]提出了一种新的蜣螂优化, 改进肺癌检测与分类深度特征融合模型技术; Zilong 等^[13]提出一种利用当前种群定向偏移的最优值和基于对立的学习多策略 DBO 算法; Guo 等^[14]提出蜣螂优化卷积神经网络, 以提高说话人识别的正确率及其抗噪能力; Xue 等^[15]提出蜣螂优化器-反向传播神经网络与改进粒子群优化算法相结合的多目标优化方法; Ji 等^[16]提出了一种 Tent 混沌映射和蜣螂混合算法相结合, 应用于水下伸缩臂机器人轨迹避障规划方法; Yuan 等^[17]提出了一种围绕蜣螂优化器-卷积神经网络模型, 并辅以特征融合和决策级融合的多样化优化策略, 用以增强 SAR 图像中的水体检测性能。

尽管 DBO 算法已广泛应用于各种工程问题, 如数值模拟和稳定性域、数据集分类、特征获取选择等, 但在移动机器人路径规划中的应用较少。因此, 本文提出一种新的改进蜣螂算法用于解决移动机器人规避障碍物的路径规划。在所提算法中, 首先, 使用 ISPM 混沌映射初始化蜣螂位置。二是应用贪婪选择与改进透镜成像策略对蜣螂觅食阶段的路径信息进行更新, 提高全局搜索能力。最后, 引入莱维飞行与改进动态权重策略及时跳出局部最优解。对几种基准函数进行了仿真, 并与其他智能优化算法进行了对比, 验证了改进算法的有效性。改进后的 DBO 在移动机器人路径规划上进行了测试, 验证了该算法的实用性。

1 路径规划地图建模

为了便于离散表示以及导航, 本研究采用了基于网格的方法, 将环境分割成一组相同大小的网格。在给定的二维离线路径规划问题中, 将环境映射到大小为 $n \times n$ 的网格上, 其中网格大小 n 取决于路径规划问题所需的精度。在网格中取机器人的尺寸小于单元格的尺寸, 从而不考虑自身的旋转动力学。为简单起见, 将起始位置作为左下角, 表示网格位置(1,1)。在笛卡尔坐标系下, 目的地被固定为

(n, n)(或网格的右上角), 如图 1(a)所示。这个假设是合理的, 因为大多数其他路径规划场景都可以通过简单的线性转换(如平移、旋转和缩放)映射到这样的网格上。在图 1(b)中将机器人环境转换为二维矩阵, 网格中每个位置 (x, y) 有两个可能值: 空闲单元为 0, 被占用单元为 1。合适的解决方案是从起点到目标点的路径(一组位置), 穿过一组自由位置。

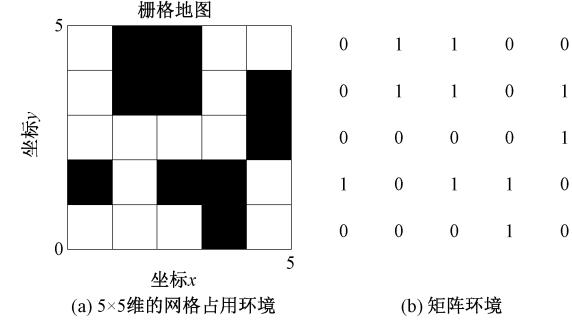


图 1 地图环境

2 蜣螂算法

蜣螂优化算法基于自然界中蜣螂的滚动、跳舞、繁殖、觅食和偷窃行为。总体优化算法, 即优化算法的数学模型, 如下所示:

1) 蜣螂滚球行为的数学模型

蜣螂是自然界中常见的昆虫, 以动物粪便为食。当蜣螂将粪球滚动时, 它可以根据月光或阳光导航, 使粪球直线运动。当没有光源时, 它的轨迹是弯曲的。假设光源的强度会影响蜣螂的运动路径, 其在搜索空间中的位置更新可以表示如下:

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

$$\Delta x = |x_i(t) - X^w|$$

式中: t 表示当前迭代次数, $x_i(t)$ 表示第 i 只蜣螂在第 t 次迭代时的位置信息, $k \in (0, 0.2]$ 为表示偏转系数的常数, b 为 $(0, 1)$ 范围内的不变量, α 为取 1 或 -1 的自然系数, Δx 表示模拟光强的变化, X^w 为全局最差位置。

2) 蜣螂跳舞行为的数学模型

当蜣螂遇到障碍物无法移动时, 它们会跳到粪球的顶部, 重新定位自己并获得新的路线。假设蜣螂在确定新方向后会立即继续滚动粪球, 该过程可以表示如下:

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

其中, $\theta \in [0, \pi]$ 为偏转角。

3) 蜣螂繁殖行为的数学模型

在蜣螂将粪球运送到安全位置并将其隐藏后, 雌性蜣螂会在粪球中产卵。选择合适的产卵地点对蜣螂来说非常重要。模拟蜣螂产卵的区域边界选择策略定义如下:

$$Lb^* = \max(X^* \cdot (1 - R), Lb) \quad (3)$$

$$Ub^* = \min(X^* \cdot (1 + R), Ub)$$

其中, X^* 表示当前局部最优位置; Lb^* 和 Ub^* 分别代表产卵区的下界和上界; $R = 1 - t/T_{\max}$; T_{\max} 表示最大迭代次数; Lb 和 Ub 分别是搜索空间的下界和上界。

一旦确定了产卵区域, 雌性蜣螂就会选择该区域进行产卵。对于 DBO 算法, 假设每只雌性蜣螂在每次迭代中只产下一个卵。卵的位置可以表示如下:

$$B_i(t+1) = x^* + \mathbf{b}_1 \cdot (B_i(t) - Lb^*) + \mathbf{b}_2 \cdot (B_i(t) - Ub^*) \quad (4)$$

其中, $B_i(t)$ 为第 i 个卵球在第 t 次迭代时的位置; \mathbf{b}_1 和 \mathbf{b}_2 是两个独立的随机向量, 大小为 $1 \times D$; D 是维数。

4) 蜣螂觅食行为的数学模型

一些成熟的小蜣螂会从地下出来寻找食物, 需要建立最优的觅食区, 以引导蜣螂觅食。最佳觅食区的边界定义如下:

$$\begin{aligned} Lb^b &= \max(X^b \cdot (1 - R), Lb) \\ Ub^b &= \min(X^b \cdot (1 + R), Ub) \end{aligned} \quad (5)$$

其中, X^b 表示全局最优位置; Lb^b 和 Ub^b 分别为最优觅食区域的下界和上界。因此, 小蜣螂的位置更新如下:

$$x_i(t+1) = x_i(t) + c_1 \cdot (x_i(t) - Lb^b) + c_2 \cdot (x_i(t) - Ub^b) \quad (6)$$

其中, c_1 为服从正态分布的随机数; c_2 是 $(0, 1)$ 范围内的随机向量。

5) 蜣螂偷窃行为的数学模型

还有一些蜣螂称为小偷蜣螂, 会从其他蜣螂那里偷粪球。DBO 算法假设偷窃行为发生在最佳觅食位置 X^b , 并且偷窃蜣螂的位置更新如下:

$$x_i(t+1) = x^b + s \cdot \mathbf{g} \cdot (|x_i(t) - X^*|) + |x_i(t) - X^b| \quad (7)$$

其中, \mathbf{g} 是大小为 $1 \times D$ 的正态分布随机向量, s 是常量值。

3 改进蜣螂算法

3.1 策略 1: 基于 ISPM 混沌的种群初始化

混沌映射是指一类非线性动力学系统, 由一组非线性微分方程描述。在优化算法中, 利用混沌序列进行种群初始化的操作会对整个算法的效果产生显著影响, 通常比使用伪随机数生成的种群效果更好。本文提出了一种将正弦映射和分段线性混沌映射相结合而设计的改进混沌映射, 该映射具有优越的混沌性和遍历性。

1) Sine 映射可以用式(8)表示:

$$a(t+1) = \mu \sin(\pi a(t)) \quad (8)$$

其中, $\mu \in [0, 1]$, $a(t) \in [0, 1]$ 。而 Sine 映射也存在一些缺点, 如具有有限的混沌映射范围和分布不均匀性。

2) 分段线性混沌映射(piece wise linear chaotic map, PWLCM)^[18]可以用式(9)表示:

$$a(t+1) = f(a(t), \eta) = \begin{cases} \frac{a(t)}{\eta}, & 0 \leq a(t) < \eta \\ \frac{a(t) - \eta}{0.5 - \eta}, & \eta \leq a(t) < 0.5 \\ 0, & a(t) = 0.5 \\ f(1 - a(t), \eta), & 0.5 \leq a(t) < 1 \end{cases} \quad (9)$$

其中, 当控制参数 $\eta \in (0, 0.5)$, $a(t) \in [0, 1]$, 系统处于混沌状态。PWLCM 映射具备一致的变量分布和较好的遍历性, 但由于其存在零点, 即当 $a(t) = 0.5$ 时, 系统会失去混沌性质, 降低了 PWLCM 在使用中的安全有效性, 因此需要对其进行改进。

3) 本节结合 Sine 与 PWLCM 映射设计一个新的混沌系统 (ISPM), 混沌映射方程 ISPM 定义如式(10)所示。

$$a(t+1) = \begin{cases} \text{bmod}\left(\frac{a(t)}{\eta} + \mu \sin(\pi a(t)) + r, 1\right), & 0 \leq a(t) < \eta \\ \text{bmod}\left(\frac{a(t) - \eta}{0.5/\eta} + \mu \sin(\pi a(t)) + r, 1\right), & \eta \leq a(t) < 0.5 \\ \text{bmod}\left(\frac{(1+a(t))/\eta}{0.5/\eta} + \mu \sin(\pi(1+a(t))) + r, 1\right), & 0.5 \leq a(t) < 1 - \eta \\ \text{bmod}\left(\frac{(1+a(t))}{\eta} + \mu \sin(\pi(1+a(t))) + r, 1\right), & 1 - \eta \leq a(t) < 1 \end{cases} \quad (10)$$

其中, $\eta \in (0, 1)$, $\mu \in (0, 1)$ 时, 系统处于完全混沌状态; r 是混沌系统的扰动参数。本次实验取 $\eta = 0.4$, $\mu = 0.3$, r 和 $a(1)$ 均取 $0 \sim 1$ 之间的随机数。

不同的混沌映射对于算法初始种群效果也不尽相同, 本文对 Cubic、Logistic、Fuch 和改进 SPM 映射在相同迭代次数 2 000 和相同初始值的条件下进行比较, Cubic 混沌映射分布图如图 2(a) 所示, Logistic 混沌映射分布图如图 2(b) 所示, Fuch 混沌映射分布图如图 2(c) 所示, 改进 SPM 混沌映射分布图如图 2(d) 所示。改进 SPM 映射更均匀地分布, 而 Logistic、Cubic 和 Fuch 映射的个体更多地分布在边界周围, 导致种群多样性的丧失。综上, 本文最终选择改进 SPM 映射作为 DBO 算法的初始化方式, 以进一步提高算法的搜索性能和解决能力。

3.2 策略 2: 混合贪婪选择策略与改进透镜成像反向学习策略

通过利用凸透镜的折射现象和光线的直线传播, 可以实现透镜成像原理。将这一原理应用在蜣螂觅食行为的搜索阶段, 可以帮助蜣螂更好地跳出局部最优解并提高收敛精度和速度。

通过变换得出 D 维空间中的反向点位置为:

$$x_j^* = \frac{Lb + Ub}{2} + \frac{Lb + Ub}{2n} - \frac{x_j}{n} \quad (11)$$

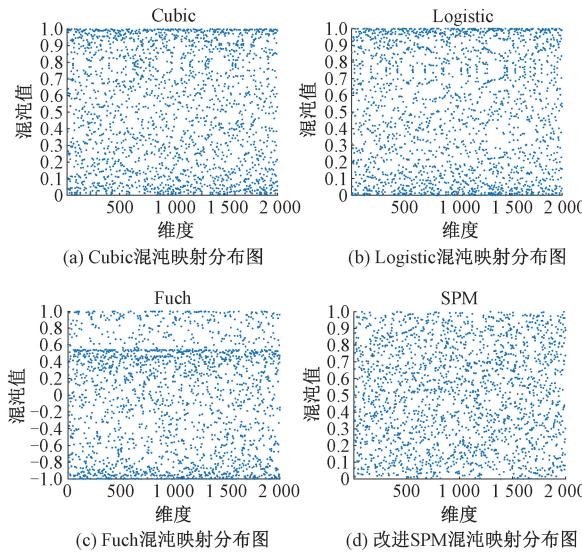


图 2 4 种映射在 2 000 次迭代下生成的分布图

其中, n 为缩放因子。

缩放因子 n 是影响透镜成像学习性能的一个重要参数。考虑到较小的 n 值生成的反向解范围更大, 而较大的 n 值能产生小范围内反向解。因此, 本文提出一种随迭代次数与搜索范围一起变化的调节因子, 这种调节方式增强了算法迭代后期局部位置的精细搜索, 表达式如下:

$$n = (Ub - Lb) \cdot (1 + \sqrt{t/T})^10 \quad (12)$$

改进后小蜣螂的位置更新如下:

$$x_{new} = x_j^* + c_1 \cdot (x_j^* - Lb^b) + c_2 \cdot (x_j^* - Ub^b) \quad (13)$$

通过透镜成像反向学习产生的反向解, 不一定优于小蜣螂位置更新式(6)的原始解。因此, 引入贪婪选择策略, 选择是否将原始解用反向解替代; 即只有当反向解的适应度值更优时, 才进行替换。公式如下:

$$x(t)' = \begin{cases} x(t), & f(x(t)) < f(x_{new}) \\ x_{new}, & f(x(t)) > f(x_{new}) \end{cases} \quad (14)$$

$x(t)'$ 为贪婪算法选择后小蜣螂的位置。

3.3 策略 3: 引入 Levy flight 并加入改进动态权重更新方式

Levy 飞行是一类随机行走, 其步数由步长决定^[19]。与高斯分布相比, Levy 分布在长距离上不会迅速下降。对于布朗运动, 每次跳跃通常很小, 分布的方差是有限的。然而, 对于 Levy 运动, 小的跳跃穿插着较长的跳跃或“飞行”, 导致分布的方差发散。因此, Levy 跳变没有特征长度尺度。

综上, 通过将 Levy 飞行与 DBO 结合, 可以有效提高算法的全局搜索能力和多样性, 防止种群陷入局部最优状态。本文使用 Mantegna 算法生成对称 Levy 稳定分布的随机步长, 即步长可能是正数也可能是负数。值得注意的是, 这里的“对称”指的是步长的正负性相等概率, 从而更好地探索解空间。根据 Mantegna 方法, Levy 分布的步长公式如式(15)所示。

$$Levy(\alpha) = X/Y^{1/\alpha} \quad (15)$$

$$\begin{cases} X: N(0, \delta_X) \\ Y: N(0, \delta_Y) \\ \delta_X = \left[\frac{\Gamma(1+\alpha) \sin\left(\frac{\pi\alpha}{2}\right)}{\Gamma\left(\frac{1+\alpha}{2}\right) \cdot \alpha \cdot 2^{(\alpha-1)/2}} \right]^{1/\alpha} \\ \delta_Y = 1 \end{cases} \quad (16)$$

式中: α 为莱维分布的参数, 取值区间通常为 $1 \leq \alpha \leq 3$, 一般取 $\alpha = 1.5$; X, Y 是服从正态分布的随机数; Γ 为伽玛函数, 计算如下:

$$\Gamma(1+\alpha) = \int_0^\infty t^\alpha e^{-t} dt \quad (17)$$

引入莱维飞行与动态权重后, 其 DBO 算法偷窃行为位置更新公式为:

$$x_i(t+1) = Levy \cdot x^b + s \cdot g \cdot (|x_i(t) - X^*|) + |x_i(t) - \omega \cdot X^b| \quad (18)$$

在基本蜣螂优化算法的偷窃行为更新阶段, 可能存在搜索范围不足、陷入局部最优解以及收敛失败等问题。为了克服这些弊端, 可以通过在位置更新公式中引入改进的动态权重系数 ω 。

改进动态权重系数如下:

$$\omega = \frac{e^{3 \cdot (1-t/T)} - e^{(-3 \cdot (1-t/T))}}{e^{3 \cdot (1-t/T)} + e^{(-3 \cdot (1-t/T))}} \quad (19)$$

在迭代初期, 使该权重系数取较大值, 从而促进全局搜索; 而在迭代后期, 该权重系数会自适应地变小, 促进局部搜索并加快算法的收敛速度。

3.4 改进蜣螂优化算法流程

本文提出了一种基于改进蜣螂优化算法的方法, 主要包括以下步骤:

步骤 1) 利用式(8)进行混沌初始化, 生成初始种群。设定蜣螂种群大小(N), 包括滚球、繁殖、觅食和盗窃在内的 4 种蜣螂比例, 变量参数的维度(D), 最大迭代次数(T)和下限(Lb)和上限(Ub)。

步骤 2) 计算所有蜣螂的适应度值, 记录全局最佳位置。

步骤 3) 更新所有蜣螂的位置。如果是滚球的蜣螂, 在无障碍模式下根据式(1)滚动动作或在障碍物模式下根据式(2)跳舞动作来更新其位置; 如果是繁殖蜣螂, 则根据式(3)和(4)通过繁殖作用更新其位置; 如果是觅食的蜣螂, 则根据式(5)和改进式(13)、(14)通过觅食动作更新其位置; 如果是盗窃蜣螂, 则根据改进式(18)通过盗窃行动更新其位置。

步骤 4) 更新后, 确定每只蜣螂的位置是否超出边界 Lb 和 Ub 。如果是, 请返回到步骤 2)。否则, 请继续执行。

步骤 5) 更新目前的最优解及其适应度值。

步骤 6) 重复上述步骤 2)~5), 直到达到最大迭代次数, 并将最优参数输出, 即所求问题的最优解, 算法终止。

4 实验结果与分析

实验环境:操作系统 Windows 11(64bit),处理器 12th Gen Intel(R) Core(TM) i7-12700H 2.30 GHz,电脑内存 16 G,仿真平台 MATLAB 2021a。

4.1 性能验证与分析

为了评估 IDBO 算法的性能,可选择表 1 中列出的

9 个基准测试函数进行测试。F1、F2 与 F3 是高维单峰函数,其中只有一个全局最优解存在,这类函数通常被选用来测试算法在局部搜索能力方面的表现;F4、F5 与 F6 为高维多峰函数,具有多个局部最优解,可用于测试算法跳出局部最优的性能;F7、F8 与 F9 为低维多峰函数,具有多个局部最优解,可以从低维角度分析算法跳出局部最优的性能。

表 1 测试函数

函数	维数	范围	理论最值
$F1 = \sum_{i=1}^n x_i^2$	30	[-100,100]	0
$F2 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10,10]	0
$F3 = \sum_{i=1}^n ix_i^4 + \text{random}[0,1]$	30	[-1.28,1.28]	0
$F4 = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	[-500,500]	-12 569.5
$F5 = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left[\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right] + 20 + e$	30	[-32,32]	0
$F6 = \frac{1}{4000} \sum_{i=1}^n x_i^2 + \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600,600]	0
$F7 = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[-5,5]	0.000 307 5
$F8 = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	2	[-5,5]	0.397 887
$F9 = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2\right)$	6	[0,1]	-3.321 995

本文选取了基本的蜣螂算法(DBO)、鲸鱼算法(whale optimization algorithm, WOA)、灰狼算法(grey wolf optimizer, GWO)和麻雀算法(SSA)与改进的蜣螂算法(IDBO)对函数进行优化结果对比,所有算法的共有参数统一设置,种群规模统一设置为 50,迭代次数设置为 500 次,5 种优化算法具体参数设置见表 2,其保证了实验的公平性与合理性。各种算法各独立运行 20 次,记录结果的最优值 best、平均值 ave 和标准差 std 用来衡量其优化效果。实验结果如表 3。

表 2 5 种优化算法的关键参数设置

算法	参数值设置
IDBO	$K=0.1, b=0.3, S=0.5$
DBO	$K=0.1, b=0.3, S=0.5$
WOA	$b=1$
GWO	$a_{\max}=2, a_{\min}=0$
SSA	$Dis=0.7 \text{ N}, Flo=0.1 \text{ N}, Vig=0.2$

由表 3 可以看出:对于高维单峰函数 F1、F2 只有本文提出的 IDBO 算法能找到理论最优值,均值、标准差均为 0;对于高维单峰函数 F3,只有 IDBO 算法能找到相对最优值,且均值比其他算法好,标准差最小。对于高维多峰函数 F4, IDBO 算法与 DBO 算法搜索结果较好,其余算法搜索结果误差较大。对于高维多峰函数 F5, IDBO 算法与 SSA 算法和 DBO 算法搜索结果一样,其性能优于 GWO 算法和 WOA 算法。对于高维多峰函数 F6,5 种算法实现了 100% 的最优搜索效果,值得注意的是, IDBO 算法在搜索最优值时保留了 DBO 算法的性能,两项指标的均值和标准差均为 0,同时保留了原算法的性能。其中,对于低维多峰函数 F7,只有本文提出的 IDBO 算法能找到理论最优值,且 3 个评价指标优于其他 4 种算法。对于低维多峰函数 F8,除 GWO 算法其他算法都能找到理论最优值,但 IDBO 算法与 SSA 算法搜索精度更高。对于低维多峰函数 F9,除 WOA 算法外其他算法都能找到理论最优值,但 IDBO 算法整体评价指标比其他算法好。总体而言,对于高维单峰和高维多峰测试函数, IDBO 算法的性能明显优

表 3 测试结果对比

函数	IDBO	DBO	WOA	GWO	SSA
F1	f_{best}	0	7.9667×10^{-160}	9.1949×10^{-86}	1.0907×10^{-28}
	ave	0	4.829×10^{-125}	1.2272×10^{-73}	2.8527×10^{-27}
	std	0	1.5255×10^{-124}	3.8805×10^{-73}	5.7412×10^{-27}
F2	f_{best}	0	9.714×10^{-74}	9.9341×10^{-56}	2.8425×10^{-17}
	ave	0	3.9593×10^{-60}	1.9508×10^{-51}	7.4769×10^{-17}
	std	0	1.252×10^{-59}	3.6473×10^{-51}	6.4426×10^{-17}
F3	f_{best}	4.3233×10^{-6}	3.2918×10^{-5}	0.000 462 82	4.7294×10^{-5}
	ave	0.000 114 41	0.000 826 6	0.002 988	0.002 037 6
	std	9.0928×10^{-5}	0.000 797 43	0.003 338 2	0.000 973 93
F4	f_{best}	-12 569.464 8	-12 569.486 6	-12 351.452 7	-7 970.907
	ave	-12 569.060 7	-12 567.768 5	-10 776.479 3	-6 199.448 8
	std	0.353 45	3.790 7	1 699.525 3	1 042.375 6
F5	f_{best}	8.8818×10^{-16}	8.8818×10^{-16}	4.4409×10^{-15}	7.5495×10^{-14}
	ave	8.8818×10^{-16}	8.8818×10^{-16}	5.862×10^{-15}	1.0321×10^{-13}
	std	0	0	1.8346×10^{-15}	1.8709×10^{-14}
F6	f_{best}	0	0	0	0
	ave	0	0	0	0
	std	0	0	0	0
F7	f_{best}	0.000 307 5	0.000 468 28	0.000 309 75	0.000 307 83
	ave	0.000 369 19	0.000 750 66	0.000 643 84	0.000 529 44
	std	6.5889×10^{-5}	0.000 203 12	0.000 382 14	0.000 260 06
F8	f_{best}	0.397 89	0.397 89	0.397 89	0.397 89
	ave	0.397 89	0.397 89	0.397 89	0.397 89
	std	0	6.0089×10^{-6}	4.2221×10^{-6}	0.000 165 94
F9	f_{best}	-3.322	-3.322	-3.3214	-3.322
	ave	-3.312 9	-3.247 8	-3.254 1	-3.216 7
	std	0.023 572	0.082 321	0.079 237	0.316 13

于其他算法;对于低维多峰函数, IDBO 算法的性能相比其他算法有较大的提升。

为了更直观地比较算法的收敛精度和收敛速度,本文根据迭代次数和适应度值绘制了被测函数的收敛曲线,如图 3 所示为部分函数收敛图。首先,通过观察高维单峰函数 F1、F2 的平均收敛曲线如图 3 中(a)与(b)所示,可以看出 IDBO 算法的收敛速度和收敛精度优于其他算法。其次,分析高维多峰函数的平均收敛曲线如图 3 中(c)与(d)所示, IDBO 算法在高维多峰函数 F4、F5 上的收敛速度和收敛精度,远远超过其他算法。最后,通过观察低维多峰函数 F7、F8 的平均收敛曲线如图 3 中(e)与(f)所示,可以看出 IDBO 算法的收敛速度虽有所下降,但仍能保证算法的收敛精度高于其他算法。总体而言, IDBO 算法的收敛速度和收敛精度普遍优于其他算法,说明所提算法具有较好的全局搜索能力和跳出局部最优的能力。

4.2 算法应用与对比

通过在不同大小和复杂度的栅格地图上进行实验,验

证改进蜣螂算法 (IDBO) 在路径规划方面的可行性。同时,与新型智能优化算法蜣螂算法 (DBO)、鲸鱼算法 (WOA)、灰狼算法 (GWO) 和麻雀算法 (SSA) 进行对比。在接下来的实验中,使用相同的参数运行相关算法:群大小为 50,最大迭代次数 100,各种算法各独立运行 20 次,机器人从左下角圆形标记的初始位置开始并接近右上角标记为矩形的目的地。算法重要参数设置同表 2。

1) 20×20 栅格地图实验

对于本节第一个实验,将采用改进蜣螂算法的路径规划相关性能与其他 4 种算法的工作结果进行比较。相同的地图环境也在这些算法中分别使用,它由 15% 的障碍物占比组成。改进的算法与其他算法的性能比较包括:对改进的 DBO 算法和其他算法在相同环境下的最短路径长度、运行时间和到达可行路径的迭代次数进行了比较。实验结果如图 4 和表 4 所示。

图 4(a) 中 SSA 与 WOA 算法收敛的相对最快,但容易陷入局部最优解, IDBO 收敛效果更好。图 4 中(b)~(f)

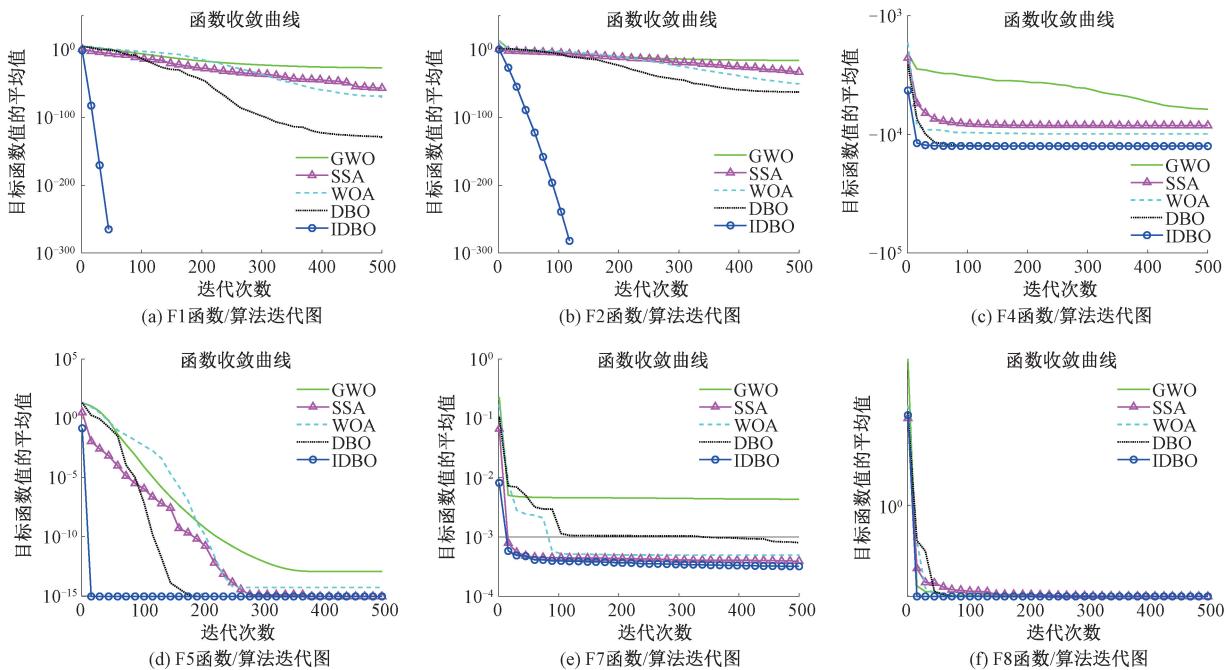


图3 部分函数收敛图

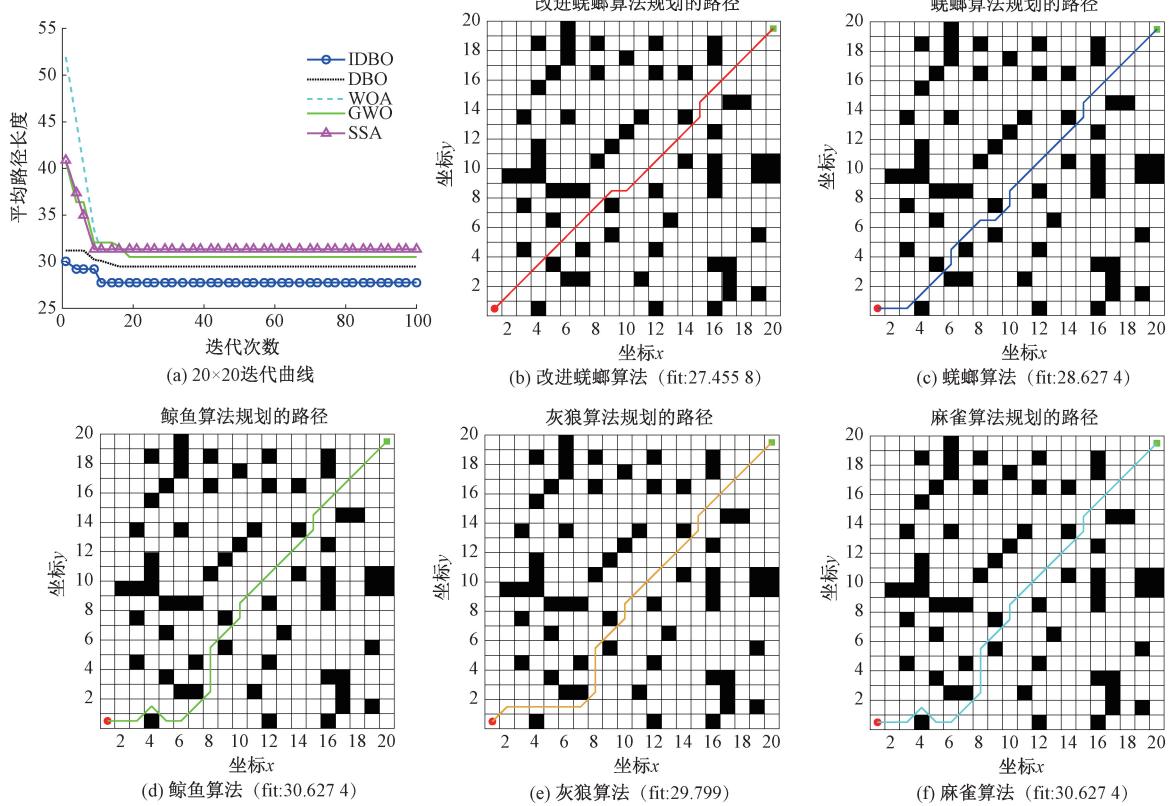


图4 20×20地图实验结果

表示5种算法各自运行20次找到的最优路径,从中可以看出,所有算法都可以跳过陷阱障碍物,并能高效地找到机器人行进路径,但IDBO找到了接近最优的路径并表现

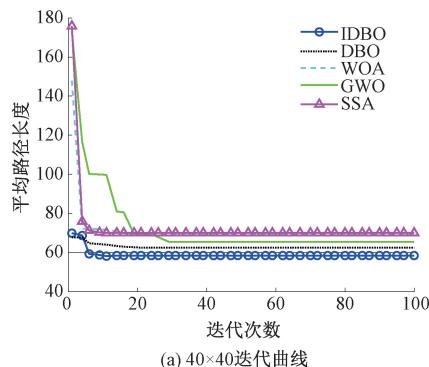
更好。

表4统计了每种算法解的最优值、平均值、平均迭代次数、平均运行时间、拐点数。将改进蜣螂算法和其他

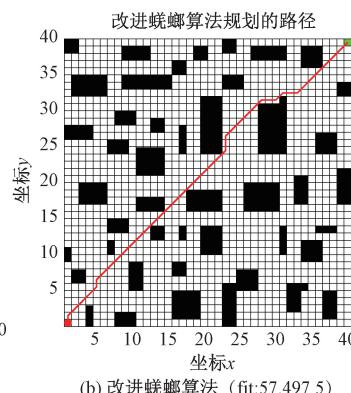
表 4 20×20 地图实验结果对比

算法	最优路径 / m	平均路径 / m	平均迭代次数 / 次	平均运行时间 / s	拐点数
IDBO	27.455 8	27.748 7	11	1.313	4
DBO	28.627 4	29.489 3	16	1.627	9
WOA	30.627 4	31.091 8	10	2.261	10
GWO	29.799 0	30.506 1	19	2.541	8
SSA	30.627 4	31.334 5	9	3.152	10

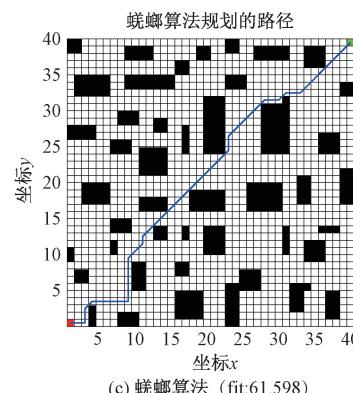
4 种算法路径最优值的结果比较, 比其他算法路径平均值



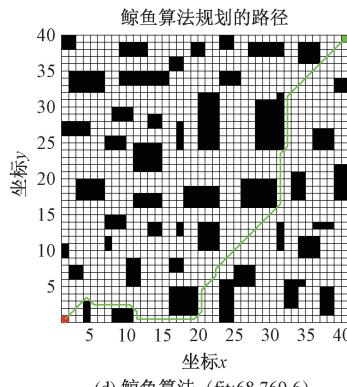
(a) 40×40 迭代曲线



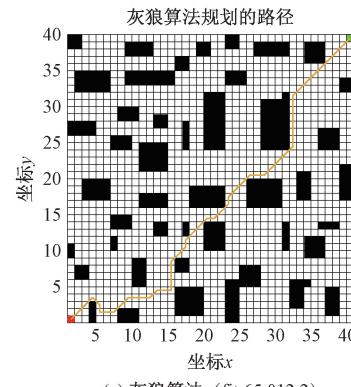
(b) 改进蜣螂算法 (fit: 57.497 5)



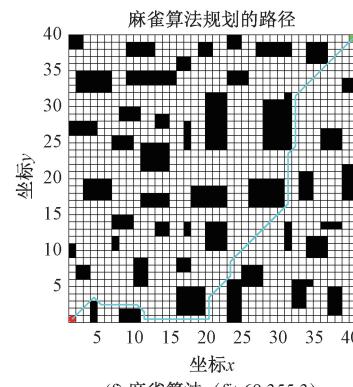
(c) 蜣螂算法 (fit: 61.598)



(d) 鲸鱼算法 (fit: 68.769 6)



(e) 灰狼算法 (fit: 65.012 2)



(f) 麻雀算法 (fit: 69.355 3)

图 5 40×40 地图实验结果

表 5 40×40 地图实验结果对比

算法	最优路径 / m	平均路径 / m	平均迭代次数	平均运行时间 / s	拐点数
IDBO	57.495 7	58.376 2	10	2.952	9
DBO	61.598	62.426 4	21	2.777	13
WOA	68.769 6	69.355 3	17	8.927	14
GWO	65.012 2	65.436 2	29	9.823	19
SSA	69.355 3	69.941 1	11	9.95	13

从图 5(a)~(f) 中可以显示, 实验环境 2 中的障碍物覆盖率比环境 1 中提高了 10% 以后, 增加障碍物覆盖范围会使得搜索全局最优解变得更加困难, 但各类算法从初始种群开始, 逐代进化, 最终都得到了接近理想的路径, 即最短安全无碰撞

缩短约 9.28%。在弯曲次数上, 改进后的算法减少了路径规划中的冗余拐点和冗余节点, 路径更平坦、更短。综合最优路径、平均路径、平均迭代次数和平均运行时间等参数进行评比, 这些参数值也略有优势。基于上述结果, 可以明显看出改进蜣螂算法优于其他优化技术。所提出的 IDBO 算法提供了最佳路径。

2) 40×40 棚格地图实验

在大小为 40×40 的正方形区域内, 将 25% 障碍物占比的地图环境用于第 2 个实验来进行研究比较, 实验结果如图 5 和表 5。

路径。相比之下, 从收敛曲线可以看出 IDBO 的收敛精度高于 DBO、WOA、GWO 和 SSA, 而且与其他算法相比具有较强的探索能力, 在效率和收敛性方面具有最佳性能。

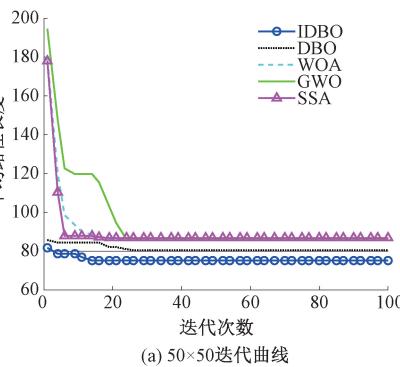
表 5 中数据与表 4 中数据相比, 表 5 中路径规划的模拟时间和距离都有所增加。与蜣螂算法相比, IDBO 在此实验中仿真耗时方面略显不足。但是, 从路径长度和平滑度的角度来看, IDBO 在复杂环境中仍然有效, 比其他算法路径平均值缩短约 12.93%, 拐角数最少, 平滑度和路径安全性大大提高。此外, IDBO 的平均迭代次数最少, 进一步证明了 IDBO 的快速收敛性。总而言之, IDBO 的性能优于其他四种测试算法。

3) 50×50 棚格地图实验

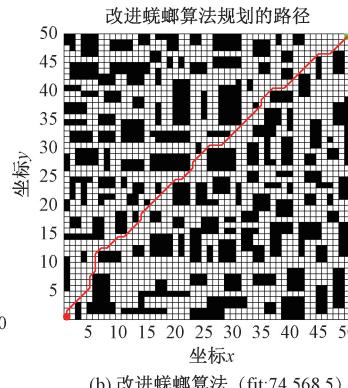
在大小为 50×50 的正方形区域内, 将 40% 障碍物占

比的地图环境用于第3个实验来进行研究比较,实验结果

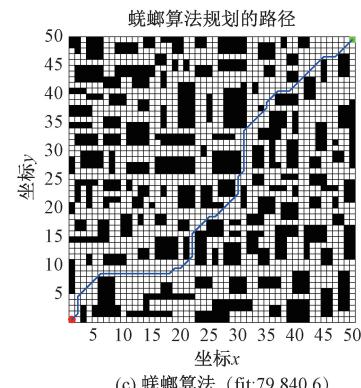
如图6和表6所示。



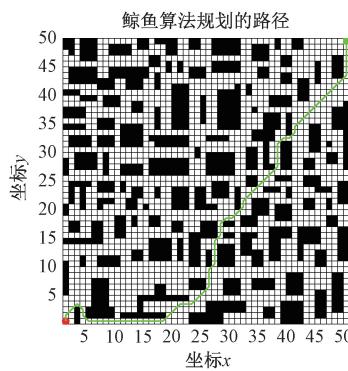
(a) 50×50迭代曲线



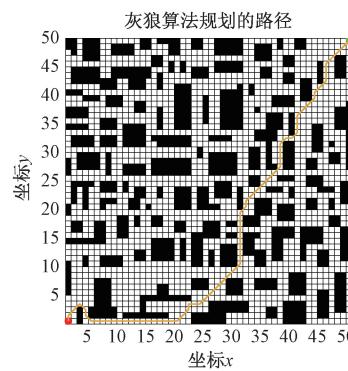
(b) 改进蜣螂算法 (fit: 74.568 5)



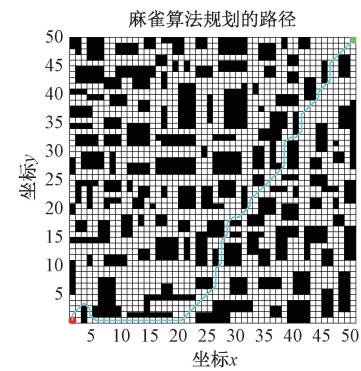
(c) 蜣螂算法 (fit: 79.840 6)



(d) 鲸鱼算法 (fit: 85.254 8)



(e) 灰狼算法 (fit: 85.840 6)



(f) 麻雀算法 (fit: 85.254 8)

图6 50×50地图实验结果

表6 50×50地图实验结果对比

算法	最优路径/ 平均路径/ 平均迭代		平均运行时间/s	拐点数	
	m	m			
IDBO	74.568 5	75.154 3	13	5.098	22
DBO	79.840 6	80.497 4	25	6.048	23
WOA	85.254 8	86.312 0	18	11.567	25
GWO	85.840 6	86.126 5	23	12.883	23
SSA	85.254 8	86.816 9	17	13.348	26

从图6(a)~(f)中可以显示,WOA、GWO 和 SSA 在优化过程中的最终收敛趋势很相似,并且随着障碍物覆盖率继续增大,这3种算法收敛到更好的路径变得困难,稳定性上相对较差。但IDBO可以在每个实验中总是迅速收敛到最优值,收敛速度非常快,这是因为IDBO在迭代开始时会迅速收敛到一个稳定的值。经过上述分析,可以看出IDBO与其他先进的算法相比稳定性略好,能够在全局勘探和局部开发之间取得一定的平衡。

从表6列出的不同算法在路径规划中的数据可得,改进算法比其他算法路径平均值缩短约11.201%。随着障碍物占比增大,迭代次数也会不断增大,最优解的搜索时间也会增加,但改进蜣螂算法实际上具有更高的效率,表明算法具有较强的搜索能力。综合以上数据和分析说明,

无论在简单场景还是复杂场景中, IDBO 算法的收敛精度和收敛速度等性能均得到了提升。

5 结论

移动机器人的路径规划是指为机器人找到一条合适的路径,同时避免机器人和障碍物之间的碰撞。本文在传统蜣螂算法的基础上,提出了一种多混合策略的改进蜣螂算法。从仿真结果可以看出,将改进蜣螂算法用在求解网格地图中机器人路径规划问题时的性能与其他算法进行比较,本文提出的IDBO算法在最短距离(即具有更高的适应度值)方面优于其他优化技术。该算法能够有效地引导机器人以接近最优的路径从起始位置移动到结束位置,而不会与环境中的障碍物发生碰撞。

参考文献

- [1] 周敬淞,张军,肖毅,等.基于仿蟹刚柔耦合机构的搜救机器人设计[J].仪器仪表学报,2023,44(6):11-20.
- [2] 林俊志,席万强,周俊,等.基于改进PRM和APF的移动机器人路径规划[J].国外电子测量技术,2022,41(12):1-6.
- [3] 王文韬,胡立夫,苏安琪,等.基于模糊算法的双轮机器人垃圾分类设计[J].国外电子测量技术,2022,41(12):133-139.
- [4] 张德慷,马桂金,李坤,等.面向ESM任务的机械臂控

- 制策略与优化算法[J]. 电子测量与仪器学报, 2022, 36(5):30-38.
- [5] 童枭军, 陈洋, 路浩, 等. 面向表计读数的变电站巡检机器人路径规划[J]. 电子测量与仪器学报, 2022, 36(8): 167-177.
- [6] 郝琨, 邓晁硕, 赵璐, 等. 基于区域搜索粒子群算法的机器人路径规划[J]. 电子测量与仪器学报, 2022, 36(12):126-135.
- [7] 戈一航, 杨光永, 徐天奇, 等. 基于 SSA 优化 PID 在移动机器人路径跟踪中的研究[J]. 国外电子测量技术, 2021, 40(9):64-69.
- [8] 黄志锋, 刘媛华. 基于四阶贝塞尔曲线和改进狮群优化算法求解路径规划问题[J]. 信息与控制, 2023, 52(2): 176-189.
- [9] 潘志远, 卜凡亮. 基于蜣螂算法优化的 DV-Hop 定位算法[J]. 电子测量与仪器学报, 2023, 37(7):33-41.
- [10] 周亚中, 何怡刚, 邢致恺, 等. 基于 IDBO-ARIMA 的电力变压器振动信号预测[J]. 电子测量与仪器学报, 2023, 37(8):11-20.
- [11] 甄然, 袁明月, 武晓晶, 等. 基于改进蜣螂算法的无人机航迹规划[J/OL]. 无线电工程:1-13[2023-11-25].
- [12] ALAMGEER M, ALRUWAIS N, ALSHAHRANI H M, et al. Dung beetle optimization with deep feature fusion model for lung cancer detection and classification[J]. Cancers, 2023, 15(15): 3982.
- [13] ZILONG W, PENG S. A multi-strategy dung beetle optimization algorithm for optimizing constrained engineering problems[C]. IEEE Access, 2023.
- [14] GUO X, QIN X, ZHANG Q, et al. Speaker recognition based on dung beetle optimized CNN[J]. Applied Sciences, 2023, 13(17): 9787.
- [15] XUE H, LI T, LI J, et al. Multi-objective optimization for turning process of 304 stainless steel based on dung beetle optimizer-back propagation neural network and improved particle swarm optimization[J]. Journal of Materials Engineering and Performance, 2023: 1-14.
- [16] JI H, YAN F. An effective obstacle avoidance and motion planning design for underwater telescopic arm robots based on a tent chaotic dung beetle algorithm [J]. Electronics, 2023, 12(19): 4128.
- [17] YUAN Q, WU L, HUANG Y, et al. Water-body detection from spaceborne SAR images with DBO-CNN [C]. IEEE Geoscience and Remote Sensing Letters, 2023.
- [18] 班多晗, 吕鑫, 王鑫元. 基于一维混沌映射的高效图像加密算法[J]. 计算机科学, 2020, 47(4):278-284.
- [19] 徐大也, 胡立坤, 王小勇, 等. 基于概率路线图法的窄道采样与轨迹优化[J]. 国外电子测量技术, 2023, 42(2): 1-8.

作者简介

万怡华, 硕士, 助教, 主要研究方向为智能控制与优化。
E-mail: wyh1572020@126.com
张雪梅, 博士, 副教授, 主要研究方向为热电材料与器件。