

采用多策略改进黑猩猩算法的 农业机器人路径规划

艾尔肯·亥木都拉, 穆占海, 郑威强

(新疆大学智能制造现代产业学院, 830047, 乌鲁木齐)

摘要: 为缩短农业机器人全局路径规划的长度和时间并获得更好的安全路径,提出了一种采用多策略改进黑猩猩算法(MICHOA)的路径规划方法。首先,对传统的黑猩猩算法(ChOA)进行改进,在种群初始化阶段引入佳点集策略提高种群的多样性;其次,根据黑猩猩实际寻优过程提出正态随机余弦收敛因子策略,平衡了算法全局勘探与局部开发能力;然后,在算法陷入局部最优停滞时采用贪婪竞争机制的停滞扰动策略,加快算法跳出局部最优并准确找到全局最优解;最后,利用标准测试函数验证 MICHOA 算法的寻优性能,建立了具有 111 个障碍物的环境栅格地图开展仿真实验,将 MICHOA 算法应用于农业机器人路径规划,并与其他 4 种较为优秀的改进 ChOA 算法进行比较。结果表明:MICHOA 算法在单峰和复杂多峰函数上均具有较高的寻优精度、稳定的收敛性和良好的鲁棒性;MICHOA 算法的路径搜索性能优于其他 4 种改进 ChOA 算法,其中路径长度缩短了 28.01%,寻优时间和迭代次数分别减少了 54.58%和 85.87%。

关键词: 农业机器人;黑猩猩优化算法;路径规划;局部最优

中图分类号: TP242.6 **文献标志码:** A

DOI: 10.7652/xjtuxb202308016 **文章编号:** 0253-987X(2023)08-0161-11

Agricultural Robot Path Planning Using Multi-Strategy Improved Chimp Optimization Algorithm

Aierken HAIMUDULA, MU Zhanhai, ZHENG Weiqiang

(School of Smart Manufacturing Modern Industrial, Xinjiang University, Urumqi 830047, China)

Abstract: To shorten the length and time of global path planning for agricultural robots and to obtain better safe paths, a multi-strategy improved chimp optimization algorithm (MICHOA) is proposed. Firstly, a good point set strategy is introduced in the population initialisation stage to improve the population diversity, thus improving the traditional chimp optimization algorithm; then, a normal stochastic cosine convergence factor strategy is proposed based on the actual chimp search process to balance the global exploration and local exploitation capability of the algorithm; secondly, a stagnation perturbation strategy with a greedy competition mechanism is used when the algorithm falls into a local optimum stagnation to speed up the algorithm's jumping out of the local optimum and finding the global optimum. Finally, a standard test function is used to verify the performance of the MICHOA algorithm in finding the global optimum, and an environmental raster map with 111 obstacles is created for corresponding experimental simulations. The MICHOA algorithm is applied to agricultural robot path planning and compared with

the four relatively excellent improved ChOA algorithms. The results show that the MICHOA algorithm features high search accuracy, stable convergence and good robustness in terms of both single-peak and complex multi-peak functions; the MICHOA algorithm outperforms the other four improved ChOA algorithms in path search performance, with a 28.01% reduction in path length, 54.58% reduction in search time and 85.87% reduction in the number of iterations.

Keywords: agricultural robot; chimp optimization algorithm; path planning; local optimum

机器人技术研究领域中,路径规划一直是重要研究课题之一^[1]。随着机器人应用越来越广泛,有毒、高温、高湿等挑战性环境,农业温室巡检^[2-3]等一系列新型的任务需求,都对路径规划算法的研究提出了挑战。传统的全局路径规划算法包括 Dijkstra 算法^[4-6]、A* 算法^[7-9]等,其优点是计算速度快,可以获得无碰撞的有效路径,但计算复杂度较高且目的地无法到达的缺点也非常明显^[10]。近年来,群体智能算法为路径规划问题提供了新的发展思路^[11],粒子群算法(PSO)^[12-13]、蚁群算法(CA)^[14-15]、灰狼算法(GWO)^[16-17]等均属于典型的群体智能算法。与传统规划方法相比,群体智能算法利用全局搜索来寻找最优近似解,其核心是探索 and 开发。群体智能算法将随机算法和局部搜索算法相结合,在解决移动机器人路径规划研究问题上表现良好。

黑猩猩优化(ChOA)算法^[18]是一种新型的启发式群智能算法,灵感来自于黑猩猩的狩猎行为。黑猩猩优化算法综合了灰狼优化算法、蜻蜓优化算法和鲸鱼优化算法的优点,其原理简单,参数少,易于操作,稳定性高,全局搜索能力强大,被广泛应用于工程领域来解决全局优化问题。Kaur 等^[19]利用正弦-余弦函数在搜索过程中更新方程,出色完成了 6 个工程规划方案。Khishe 等^[20]提出了一种加权黑猩猩算法,利用大规模数值优化问题中的低收敛速度和局部最优捕获来解决高维问题,并通过 13 个高维和 10 个真实优化问题对其进行了验证。刘成汉等^[21]提出一种融合多策略的黄金正弦黑猩猩算法,解决了 2 个实际工程优化问题。何庆等^[22]提出了混合改进策略的黑猩猩算法,用于优化拉伸、压缩弹簧的机械设计问题。

然而,对于农业机器人的路径规划问题,传统的 ChOA 算法缺点显著。Du 等^[23]提出了一种基于空斗觅食策略自适应权重的黑猩猩算法来解决无人机三维路径规划问题,既避免了与障碍物发生碰撞,又能在复杂环境中找出到达目标位置的最佳路径。在 ChOA 算法运行过程中,相比于文献[13]提出的改进策略,还需要保证初始种群均匀排布,考虑非线性寻

优过程能否更好地拟合不同黑猩猩狩猎位置的控制参数,以及解决同组内黑猩猩互相竞争等问题。

综上所述,虽然 ChOA 算法的改进机制层出不穷,但只适用于解决部分优化问题,在非线性问题上表现欠佳。因此,为了提高 ChOA 算法在机器人全局路径规划中的寻优性能,本文增加了 3 种改进策略,提出了一种采用多策略改进黑猩猩算法(MI-ChOA)的路径规划方法。首先,使用佳点集策略初始化黑猩猩优化算法种群的位置,提高初始种群在算法搜索过程的遍历性;其次,提出一种正态随机余弦收敛因子更新策略,在种群迭代过程中平衡全局搜索与局部搜索能力;最后,引入贪婪竞争机制的停滞扰动策略,帮助种群跳出局部最优。

1 ChOA 算法

ChOA 算法灵感来自于黑猩猩的狩猎行为。黑猩猩群体主要分为驱动者、障碍者、追逐者和攻击者 4 种类型,正因为这些位置的多样性,黑猩猩才可以在捕猎过程中更好地协同狩猎直至成功。标准的 ChOA 算法中,攻击者是种群的领导者,其他 3 类黑猩猩协助狩猎,社会地位依次下降。黑猩猩驱逐和追逐猎物的数学模型如下

$$d = |CX_{\text{prey}}(t) - mX_{\text{chimp}}(t)| \quad (1)$$

$$X_{\text{chimp}}(t+1) = X_{\text{prey}}(t) - ad \quad (2)$$

式中: d 为黑猩猩距离猎物的位置; t 为当前迭代次数; $X_{\text{prey}}(t)$ 为猎物位置向量; $X_{\text{chimp}}(t)$ 为黑猩猩位置向量; a 、 m 、 C 为系数变量,计算公式如下

$$a = 2fr_1 - f \quad (3)$$

$$m = \text{chaos}(3, 1, 1) \quad (4)$$

$$C = 2r_2 \quad (5)$$

式中: a 为决定黑猩猩与猎物距离的系数变量,其值在 $[-f, f]$ 之间; f 为收敛因子,随迭代次数从 2.5 非线性减小到 0; r_1 和 r_2 分别为 $[0, 1]$ 间的随机数; m 代表黑猩猩在狩猎过程中性动机的影响; chaos 是基于混沌映射的混沌函数,用来更新黑猩猩的位置; C 为黑猩猩驱逐和追赶猎物的系数,其值为 $[0, 2]$ 之间的随机数。

种群初始化后,依据适应度值的大小依次选择4个最优解作为4组的位置,其数学模型描述如下

$$\mathbf{X}_1 = \mathbf{X}_{\text{attacker}} - a_1 |C_1 \mathbf{X}_{\text{attacker}} - m_1 \mathbf{X}_{\text{chimp}}| \quad (6)$$

$$\mathbf{X}_2 = \mathbf{X}_{\text{barrier}} - a_2 |C_2 \mathbf{X}_{\text{barrier}} - m_2 \mathbf{X}_{\text{chimp}}| \quad (7)$$

$$\mathbf{X}_3 = \mathbf{X}_{\text{chaser}} - a_3 |C_3 \mathbf{X}_{\text{chaser}} - m_3 \mathbf{X}_{\text{chimp}}| \quad (8)$$

$$\mathbf{X}_4 = \mathbf{X}_{\text{driver}} - a_4 |C_4 \mathbf{X}_{\text{driver}} - m_4 \mathbf{X}_{\text{chimp}}| \quad (9)$$

$$\mathbf{X}(t+1) = (\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3 + \mathbf{X}_4)/4 \quad (10)$$

式中: $\mathbf{X}_{\text{attacker}}$ 、 $\mathbf{X}_{\text{barrier}}$ 、 $\mathbf{X}_{\text{chaser}}$ 和 $\mathbf{X}_{\text{driver}}$ 分别为攻击者、障碍者、追逐者、驱赶者的位置向量; $\mathbf{X}(t+1)$ 为黑猩猩种群的位置向量。

2 MICHOA 算法

2.1 佳点集策略初始化

ChOA 算法在迭代前,需要对黑猩猩种群初始化随机排布,但这种随机初始化的方法无法确保初始黑猩猩种群的多样性,可能会导致算法出现局部收敛现象和搜索效率低等问题。

佳点集是均匀选点的有效途径。在二维搜索空间中,根据佳点集法原理^[24],相比于随机初始化种群,佳点集策略能够有效且均匀地在搜索空间中分布初始种群,提高种群全局搜索效率。

2.2 正态随机余弦收敛因子

在实际寻优过程中,ChOA 算法简单的非线性收敛因子并不能有效地平衡算法搜索前期和后期的变化情况。为此,本文提出了一种正态随机余弦收敛因子法代替原算法更新策略。正态非线性变化过程能更好地拟合不同黑猩猩狩猎位置的控制参数,从而更准确地平衡全局与局部搜索过程。更新后的正态随机余弦收敛因子公式如下

$$f = f_{\min} + (f_{\max} - f_{\min}) \left(1 - \cos\left(\frac{\pi t}{2t_{\max}}\right)\right) \text{randn}(0, 1) \quad (11)$$

式中: $f_{\min}$ 和 $f_{\max}$ 分别为收敛因子的开始值和停止值; $t_{\max}$ 为最大迭代数;randn 为正态随机分布的随机数。

2.3 贪婪竞争机制的停滞扰动策略

2.3.1 Levy 飞行扰动策略

针对 ChOA 算法陷入局部最优时容易出现寻优停滞的问题,本文利用 Levy 飞行策略的长短距离变化,随机发生扰动避免搜索过程陷入局部最优。此策略可以最大程度地实现搜索域的多样化,进一步增强算法的全局最优搜索能力。特殊的随机游走策略既能用长时间的短距离搜索轨迹来模拟小范围的精细搜索,避免漏掉任何目标,又能采用偶尔的长距离搜索轨迹相互穿插,避免仅限于局部搜索。步长变化分布密度函数如下

$$L(\beta) \sim |s|^{-\beta}, 1 < \beta \leq 3 \quad (12)$$

$$s = \frac{\mu}{|\nu|^{1/\beta}} \quad (13)$$

式中: $L(\beta)$ 表示服从 Levy 分布的路径; s 为 Levy 运动步长;参数 $\beta = 1.5$; μ 和 ν 是符合正态分布的随机数,即 μ 服从于 $N(0, \sigma_\mu^2)$, ν 服从于 $N(0, \sigma_\nu^2)$ 。

2.3.2 贪婪竞争机制

由于两个攻击者可能因为争取猎物或者雌性黑猩猩而发生竞争,为了进一步提高种群质量,提出了采用大猩猩部队优化器^[25]中对成年女性的争夺来改进种群位置公式。这种竞争机制虽然可以进一步提高种群的位置,但不能确保得到的新解适应度优于原解。因此,将贪婪策略与竞争机制相关联,使用适者生存的概念,得到位置更新公式如下

$$\mathbf{B}(t+1) = \mathbf{B}^* + L[\mathbf{B}(t)]Q + Q|\mathbf{B}^* - \mathbf{B}(t)| \quad (14)$$

$$L[\mathbf{B}(t)] = s|\mathbf{B}^* - \mathbf{B}(t)| \quad (15)$$

$$Q = 2r_1 - 1 \quad (16)$$

$$\mathbf{X}(t+1) =$$

$$\begin{cases} \mathbf{B}(t+1), & F[\mathbf{B}(t)] \leq F[\mathbf{X}(t+1)] \\ \mathbf{X}(t+1), & F[\mathbf{B}(t+1)] > F[\mathbf{X}(t+1)] \end{cases} \quad (17)$$

式中: $\mathbf{B}(t)$ 为改进后种群的位置向量; $\mathbf{B}^*$ 为改进后种群位置的当前全局最优值,且随着迭代不断更新; Q 为当前最优黑猩猩对于迭代最优解的冲击因子; $F[\]$ 为适应度函数。

2.4 MICHOA 算法流程

综合上述改进方法,本文提出的 MICHOA 算法实现的步骤如下所示。

步骤 1 输入种群规模 N , 搜索维度 D , 种群边界 l_b 、 u_b , 最大迭代次数 $t_{\max}$ 等;

步骤 2 初始化黑猩猩个体的适应度 $F(\mathbf{X}_{\text{attacker}})$ 、 $F(\mathbf{X}_{\text{barrier}})$ 、 $F(\mathbf{X}_{\text{chaser}})$ 、 $F(\mathbf{X}_{\text{driver}})$ 和位置空间 $\mathbf{X}_{\text{attacker}}$ 、 $\mathbf{X}_{\text{barrier}}$ 、 $\mathbf{X}_{\text{chaser}}$ 和 $\mathbf{X}_{\text{driver}}$;

步骤 3 利用佳点集策略初始化种群;

步骤 4 计算黑猩猩个体适应度 $F(\mathbf{X}_i)$, 根据个体适应度排序, 其中 $i = 1, 2, 3, \dots, N$;

步骤 5 更新黑猩猩适应度, 确定 4 个最优解 $\mathbf{X}_{\text{attacker}}$ 、 $\mathbf{X}_{\text{barrier}}$ 、 $\mathbf{X}_{\text{chaser}}$ 和 $\mathbf{X}_{\text{driver}}$;

步骤 6 根据式(11)更新正态随机余弦收敛因子, 根据式(3)、(4)、(5)更新参数 a 、 m 、 C ;

步骤 7 根据式(15)对更新后的黑猩猩位置采用 Levy 飞行策略扰动, 并根据式(17)所述的贪婪竞争机制选择适应度低的解;

步骤 8 判断是否达到最大迭代次数, 若没有达到, 继续执行步骤 3; 否则, 迭代结束并输出最优解和最佳位置。

3 MIChOA 算法仿真实验

将本文提出的 MIChOA 算法、ChOA 算法与加权黑猩猩算法(WChOA)、正弦与余弦黑猩猩算法(SChOA)、黄金正弦黑猩猩算法(ICHOA)、混合改进策略黑猩猩算法(SLWChOA)这 4 种较为优秀的改进 ChOA 算法进行对比,具体参数设置见表 1。所选的 6 种算法均采用相同的实验参数,即群大小 $N=30$,维数 $D=30$,最大迭代次数 $t_{\max}=500$,每种算法独立运行 30 次后记录下结果。仿真环境的硬件配置为 Windows10 专业操作系统,CPU 为 Intel(R) Core(TM) i5-7200U CPU @ 2.50 GHz 处理器,内存为 8 GB,软件配置为 MATLAB R2021a。

表 1 各算法在仿真实验中的参数设置
Table 1 Parameter settings for each algorithm in the simulation experiment

算法	参数
MIChOA	$r_1 = [0, 1], r_2 = [0, 1], f = [0, 2.5]$ $Q = [-1, 1]$
ChOA	$r_1 = [0, 1], r_2 = [0, 1], f = [0, 2.5]$
WChOA	$r_1 = [0, 1], r_2 = [0, 1], f = [0, 2.5]$
SChOA	$r_1 = [0, 1], r_2 = [0, 1]$ $r_3 = [0, 2\pi], f = [0, 2.5]$
ICHOA	$r_1 = [0, 1], r_2 = [0, 1], f = [0, 2.5]$ $\delta_1 = 0.3, \delta_2 = 300, \delta_3 = 1.8, \rho_1 = 0.1$ $\rho_2 = 0.05, \rho_3 = 0.3, \epsilon = 300$
SLWChOA	$r_1 = [0, 1], r_2 = [0, 1], f = [0, 2.5]$ $\eta_{\max} = 10, \eta_{\min} = 1$

注: $\delta_1 \sim \delta_3, \rho_1 \sim \rho_3$ 为常数系数; ϵ 为指定迭代次数; $\eta_{\max}, \eta_{\min}$ 分别为最大、最小伸缩因子。

3.1 MIChOA 算法精度分析

实验选取 8 个标准测试函数^[26],详细信息如表 2 所示。

由于测试函数的平均值和标准差可以直观地凸显搜索算法的准确性和稳定性,因此采用选取的测试函数能够验证 MIChOA 算法的有效性。

3.1.1 MIChOA 算法不同改进策略分析

不同的改进策略对优化结果的影响不同,他们对整个改进算法的贡献也有所不同。为了分析 MIChOA 算法改进策略的有效性,将 ChOA 算法与佳点集初始化、正态随机余弦收敛因子,贪婪竞争机制的停滞扰动相结合,分别命名为 ChOA1、ChOA2、ChOA3 策略算法,研究包括 ChOA 算法在内的 5 种策略对算法有效性的影响。每种算法独立运行 30 次,统计平均值和标准差,就能直观地看出每种改进策略的改进效果,结果见表 3。

表 2 标准测试函数列表
Table 2 Table of standard test functions

标准函数	搜索范围	维度	标准函数最小值
F_1	$[-100, 100]$	30	0
F_2	$[-10, 10]$	30	0
F_3	$[-100, 100]$	30	0
F_4	$[-100, 100]$	30	0
F_5	$[-30, 30]$	30	0
F_6	$[-100, 100]$	30	0
F_7	$[-1.28, 1.28]$	30	0
F_8	$[-500, 500]$	30	0

注: $F_1(x) = \sum_{i=1}^{30} x_i^2; F_2(x) = \sum_{i=1}^{30} |x_i| + \prod_{i=1}^{30} |x_i|; F_3(x) = \sum_{i=1}^{30} (\sum_{j=1}^i x_j)^2; F_4(x) = \max\{|x_i|, 1 \leq i \leq 30\}; F_5(x) = \sum_{i=1}^{29} [100(x_{i+1} - x_i^2) + (x_i - 1)^2]; F_6(x) = \sum_{i=1}^{30} (\lfloor x_i + 0.5 \rfloor)^2;$
 $F_7(x) = \sum_{i=1}^{30} ix_i^4 + \text{random}[0, 1); F_8(x) = -\sum_{i=1}^{30} (x_i \sin(\sqrt{|x_i|}))。$

表 3 不同策略算法的测试结果
Table 3 Test results based on different strategies

算法	标准函数	适应度均值	适应度标准差
ChOA	F_1	2.37×10^{-5}	4.21×10^{-16}
ChOA1		1.26×10^{-58}	9.86×10^{-54}
ChOA2		5.67×10^{-121}	3.09×10^{-120}
ChOA3	F_2	0	0
MIChOA		0	0
ChOA		3.55×10^{-30}	1.69×10^{-29}
ChOA1	F_3	1.02×10^{-31}	2.24×10^{-31}
ChOA2		3.66×10^{-66}	8.41×10^{-66}
ChOA3		8.21×10^{-253}	0
MIChOA	F_4	0	0
ChOA		6.30×10^0	1.41×10^{-17}
ChOA1		6.51×10^{-48}	3.45×10^{-47}
ChOA2	F_5	5.40×10^{-113}	2.86×10^{-112}
ChOA3		3.12×10^{-321}	0
MIChOA		0	0
ChOA	F_6	1.21×10^{-2}	6.62×10^{-18}
ChOA1		3.82×10^{-21}	4.78×10^{-21}
ChOA2		8.16×10^{-55}	2.32×10^{-54}
ChOA3	F_7	2.24×10^{-313}	0
MIChOA		0	0

续表

算法	标准函数	适应度均值	适应度标准差
ChOA	F_5	2.86×10^1	1.54×10^{-14}
ChOA1		2.89×10^1	1.31×10^{-2}
ChOA2		2.87×10^1	1.64×10^{-2}
ChOA3		1.51×10^{-7}	8.21×10^{-3}
MICHOA		1.16×10^{-7}	5.78×10^{-27}
ChOA	F_6	3.72×10^0	3.10×10^{-15}
ChOA1		4.09×10^{-1}	4.09×10^{-1}
ChOA2		4.97×10^{-1}	2.02×10^{-1}
ChOA3		1.03×10^{-8}	1.52×10^{-7}
MICHOA		3.47×10^{-8}	5.32×10^{-26}
ChOA	F_7	9.08×10^{-4}	1.98×10^{-6}
ChOA1		2.21×10^{-4}	1.86×10^{-4}
ChOA2		8.93×10^{-5}	6.54×10^{-5}
ChOA3		1.32×10^{-7}	1.35×10^{-8}
MICHOA		3.14×10^{-7}	1.17×10^{-6}
ChOA	F_8	-5 657.03	1.41×10^6
ChOA1		-6 025.84	2.68×10^2
ChOA2		-6 072.03	6.53×10^1
ChOA3		-12 569.49	1.78×10^{-2}
MICHOA		-12 569.38	6.97×10^{-21}

从表 3 的均值与标准差的维度可以看出,不同策略对 ChOA 算法的性能均有一定程度提高,ChOA1 策略算法在搜索空间中能够有效、均匀地分布初始种群,能在每次迭代产生的新种群中选择最佳的新迭代种群,保证种群多样性的同时不断提高算法精度。ChOA2 策略算法中,正态非线性变化过程能更好地拟合不同黑猩猩狩猎位置的控制参数,在狩猎过程中不断地正态随机扰动改进后的余弦非线性因子,使算法在平衡全局与局部搜索过程中始终保持稳定性,避免由于后期收敛

速度过快而失去局部最优状态。ChOA3 策略算法利用 Levy 飞行策略的长短距离变化,随机发生扰动避免搜索过程陷入局部最优,并将贪婪策略与竞争机制相关联,利用适者生存的概念,选择更优适应度的黑猩猩进行下次迭代;此策略在算法停滞时进行扰动与竞争,在提高 ChOA 算法的性能方面发挥了更大作用。

3.1.2 其他改进 ChOA 算法的对比分析

为了进一步评价 MICHOA 算法的性能,对包括 MICHOA、ChOA^[18]、WChOA^[19]、SChOA^[20]、IChOA^[21]和 SLWChOA^[22]在内的 6 种算法进行标准函数测试对比。每个算法运行 30 次,将适应度平均值和标准差作为评价标准,结果见表 4。

由表 4 可见,与其他改进的 ChOA 算法相比,MICHOA 算法在目标函数值的精度和稳定性方面都表现优异。对于函数 F_1 、 F_2 、 F_3 和 F_4 ,可以稳定且准确地找到理论最优值为 0。实验观察到,算法迭代前期种群大范围搜索时,利用非线性收敛因子的缓慢缩减,能使种群更好地搜索全局最优解;算法迭代后期,快速缩减的收敛因子则有利于算法局部寻找最优解。MICHOA 算法利用佳点集策略保证种群多样性,并使用了一个正态随机余弦收敛因子,拟合黑猩猩种群全局和局部的搜索速度。

观察测试函数 F_5 、 F_6 和 F_7 后发现,MICHOA 算法的均值和标准差仍优于其他改进 ChOA 算法,且在寻优精度方面领先于其他改进 ChOA 算法几个数量级。同时从标准差可以看出,在保证精度的同时,相比于其他改进 ChOA 算法,MICHOA 算法能够保持较好的稳定性。针对复杂多峰函数 F_8 ,MICHOA 算法跳出局部最优,收敛到理论最优值-12 569.48 附近,从而验证了引入贪婪竞争机制的停滞扰动策略扰动种群位置,可以加快算法收敛和提高收敛精度,同时能协助算法跳出局部最优。

表 4 不同算法在标准函数中的测试结果

Table 4 Test results of different algorithms in standard test functions

算法	标准函数	适应度最优值	适应度最差值	适应度均值	适应度标准差
MICHOA	F_1	0	0	0	0
ChOA		7.39×10^{-5}	9.91×10^{-4}	2.37×10^{-5}	4.21×10^{-16}
WChOA		4.07×10^{-318}	2.26×10^{-306}	7.71×10^{-308}	0
SChOA		1.74×10^{-152}	7.52×10^{-65}	4.81×10^{-66}	1.62×10^{-65}
IChOA		0	0	0	0
SLWChOA		0	0	0	0

续表

算法	标准函数	适应度最优值	适应度最差值	适应度均值	适应度标准差
MIChOA	F_2	0	0	0	0
ChOA		3.08×10^{-5}	2.605×10^{-5}	3.55×10^{-5}	1.69×10^{-15}
WChOA		1.65×10^{-163}	9.97×10^{-160}	7.32×10^{-161}	1.86×10^{-160}
SChOA		7.98×10^{-82}	9.13×10^{-34}	4.67×10^{-35}	1.78×10^{-34}
IChOA		0	0	0	0
SLWChOA		0	2.69×10^{-268}	8.99×10^{-270}	0
MIChOA	F_3	0	0	0	0
ChOA		6.31×10^0	6.47×10^0	6.30×10^0	1.41×10^{-17}
WChOA		9.24×10^{-304}	1.32×10^{-230}	4.41×10^{-232}	0
SChOA		1.89×10^{-150}	2.52×10^{-63}	8.93×10^{-65}	4.59×10^{-64}
IChOA		0	0	0	0
SLWChOA		0	1.28×10^{-307}	5.73×10^{-309}	0
MIChOA	F_4	0	0	0	0
ChOA		0.88×10^{-2}	6.94×10^{-2}	1.21×10^{-2}	6.62×10^{-18}
WChOA		4.05×10^{-152}	1.30×10^{-147}	8.60×10^{-149}	2.51×10^{-148}
SChOA		4.21×10^{-68}	5.56×10^{-27}	2.03×10^{-28}	1.01×10^{-27}
IChOA		0	0	0	0
SLWChOA		0	0	0	0
MIChOA	F_5	1.09×10^{-7}	7.46×10^{-6}	1.16×10^{-7}	5.87×10^{-27}
ChOA		2.87×10^1	2.83×10^1	2.86×10^1	1.54×10^{-14}
WChOA		29	29	29	5.87×10^{-7}
SChOA		2.88×10^1	29	2.89×10^1	4.42×10^{-2}
IChOA		1.18×10^{-4}	1.67×10^{-4}	3.44×10^{-4}	3.21×10^{-17}
SLWChOA		1.41×10^{-5}	1.65×10^{-2}	9.88×10^{-4}	3.05×10^{-2}
MIChOA	F_6	1.03×10^{-8}	9.32×10^{-7}	3.47×10^{-8}	5.32×10^{-26}
ChOA		3.72×10^0	3.69×10^0	3.72×10^0	3.10×10^{-15}
WChOA		3.13×10^0	4.88×10^0	3.92×10^0	3.48×10^{-6}
SChOA		1.91×10^0	7.50×10^0	4.33×10^0	1.43×10^0
IChOA		3.36×10^{-5}	2.30×10^{-4}	3.91×10^{-4}	5.60×10^{-5}
SLWChOA		7.56×10^{-9}	1.91×10^{-5}	4.47×10^{-6}	6.04×10^{-6}
MIChOA	F_7	1.48×10^{-7}	9.78×10^{-6}	3.14×10^{-7}	1.17×10^{-43}
ChOA		2.32×10^{-3}	7.66×10^{-3}	9.08×10^{-4}	1.98×10^{-6}
WChOA		1.08×10^{-6}	3.84×10^{-3}	8.90×10^{-5}	7.90×10^{-5}
SChOA		3.56×10^{-6}	7.99×10^{-5}	1.85×10^{-6}	1.89×10^{-4}
IChOA		7.48×10^{-6}	3.90×10^{-5}	4.38×10^{-5}	1.02×10^{-12}
SLWChOA		9.54×10^{-6}	6.18×10^{-4}	1.15×10^{-4}	1.23×10^{-4}
MIChOA	F_8	-12 569.48	-12 568.98	-12 569.38	6.97×10^{-21}
ChOA		-5 661.67	-5 646.47	-5 657.03	1.41×10^{-6}
WChOA		-6 154.51	-5 961.82	-5 504.50	3.13×10^{-1}
SChOA		-9 867.46	-2 169.08	-3 811.58	180.02
IChOA		-12 555.48	-12 556.78	-12 555.98	1.51×10^{-11}
SLWChOA		-12 559.48	-12 555.46	-12 558.90	9.98×10^{-4}

3.2 MIChOA 算法收敛性分析

算法的收敛曲线是直接比较算法收敛性的重要指标之一,图 1 给出了 6 种算法在解决不同测试函

数适应度均值时,随机选取的 1 次迭代收敛曲线。

由图 1(a)~(d)可见,MIChOA 算法的收敛曲线随着迭代次数增加迅速减小到左下角,表明相比于其

他算法,改进的 MIChOA 算法既能保证找到理论寻优值 0,还具有稳定的最少迭代次数。由图 1(e)~(g)可以看出,寻优过程中 MIChOA 算法不断地进入局部最优,利用改进策略又跳出局部最优并找到最优位置。与其他算法相比,MIChOA 算法的寻优精度高出了 2~3 个数量级,寻优效果出色。如图 1(h)所

示,对比于其他改进 ChOA 算法,MIChOA 算法能够最大程度地接近理论值,收敛速度也非常可观。综上所述,MIChOA 算法在面对多个局部最优点的测试函数时,收敛曲线呈阶梯式下降,算法具备跳出局部最优的能力,避免收敛于局部最优或出现算法停滞状况,在寻找最优全局解时收敛精度较高。

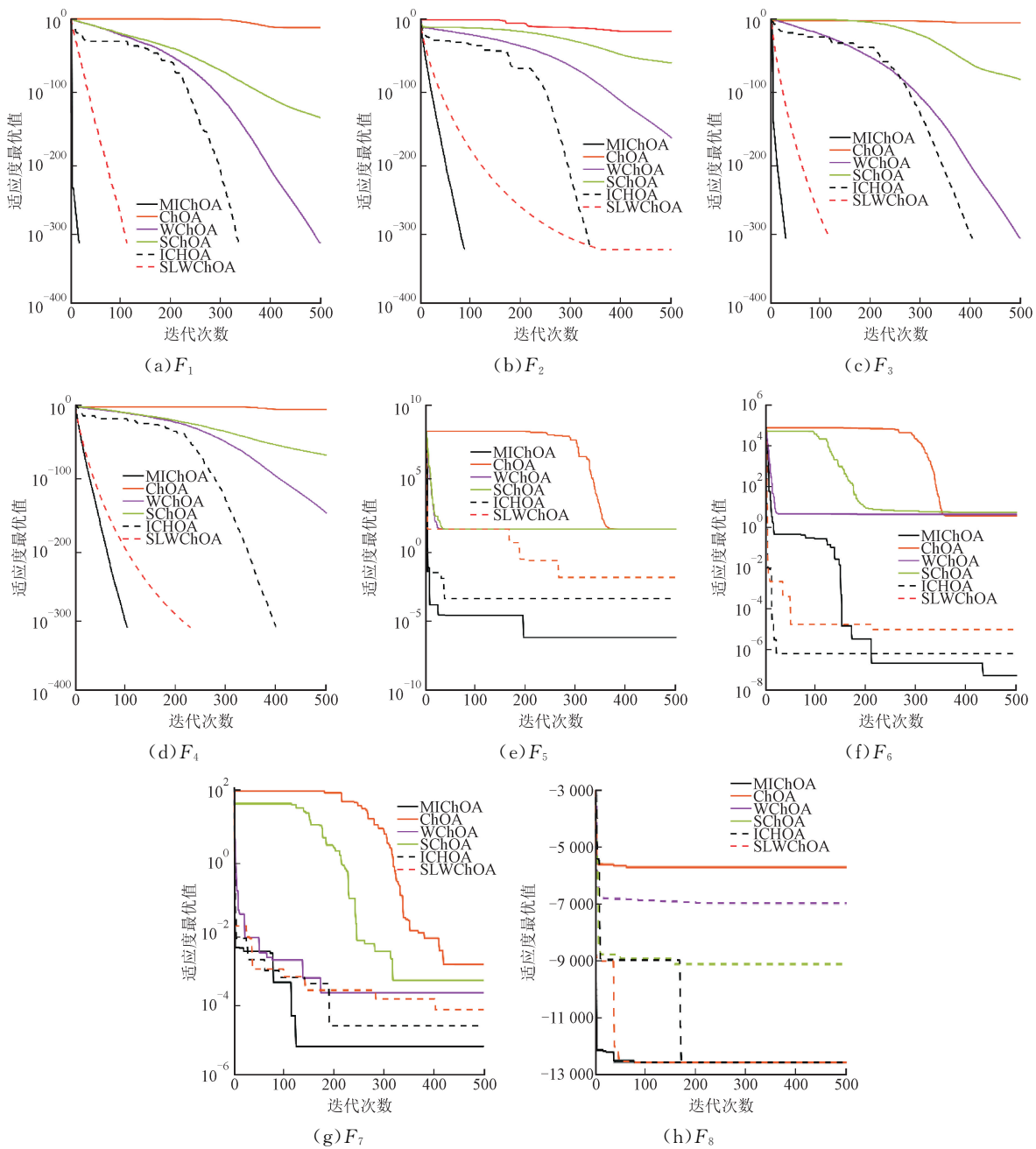


图 1 不同算法在标准测试函数中的收敛曲线

Fig. 1 Convergence curves of different algorithms in standard test functions

3.3 时间复杂度分析

计算算法的时间复杂度作为性能判别的标准之一,能够直接得到算法的计算效率。MIChOA 算法

中,若种群规模为 N ,最大迭代次数为 $t_{\max}$,个体空间维度 D 决定算法的时间复杂度。通过计算时间复杂度得知,ChOA1、ChOA2、ChOA3、ChOA 和 MIChOA 算法的

复杂度均为 $O(t_{\max}ND)$ ，虽然算法的时间复杂度相同，但 3 种改进策略算法的运行效率得到了提高。

在寻优过程中,通过本文的改进策略,首先在初始种群时选择“佳点集合”进行初始化,以降低后期迭代过程中的运算种群分布复杂度;其次利用拟合迭代过程的收敛因子平衡迭代过程;最后根据组内竞争避免搜索过程陷入局部最优,对比指定最优解与同组内最优解,在算法停滞时跳出局部最优,提升了传统 ChOA 算法的计算效率。

4 基于 MICHOA 算法的路径规划

在农业机器人规划路径之前,需要对农业机器人的工作环境进行建模和预处理,网格法^[27]是农业机器人路径规划工作环境建模所采用的一种传统方法。本文先将环境地图转换为栅格地图,再将 MICHOA 算法与传统的网格方法相结合,建立了一个约束优化问题来搜索最优机器人路径。

4.1 路径规划环境搭建

为了验证本文算法的有效性和准确性,采用 MATLAB R2021a 搭建了实际环境的栅格地图。绘制 1 张 20 m×20 m 的环境地图,在全局环境中设置 111 个障碍物,如图 2 所示。设定黑色栅格为障碍物区域,白色栅格为农业机器人可以通过的空闲区域,栅格长度为 1 m,第 i 个栅格位置坐标描述为 (x_i, y_i) 。

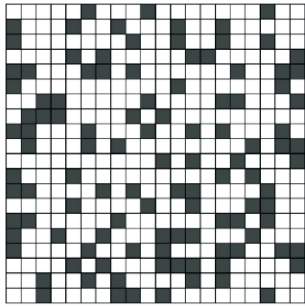


图 2 基于实际环境的建模和预处理栅格图
Fig. 2 Realistic environment-based modelling and pre-processing of raster maps

将路径长度作为每条路径的适应度函数,应用于解决机器人路径规划问题。机器人在行进过程中的无碰撞路径长度计算如下

$$L = \sum_{i=0}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (18)$$

4.2 路径规划仿真

4.2.1 改进 ChOA 路径规划算法对比

采用 MICHOA、ChOA、WChOA、SChOA、IChOA 和 SLWChOA 6 种路径规划算法进行对比实验,各算法的参数设置仍如表 1 所示。随机选取 1 次路径的结果如图 3 所示,图中左下角为起始点,右上角为目标终点,设定路径规划实验次数为 30 次。

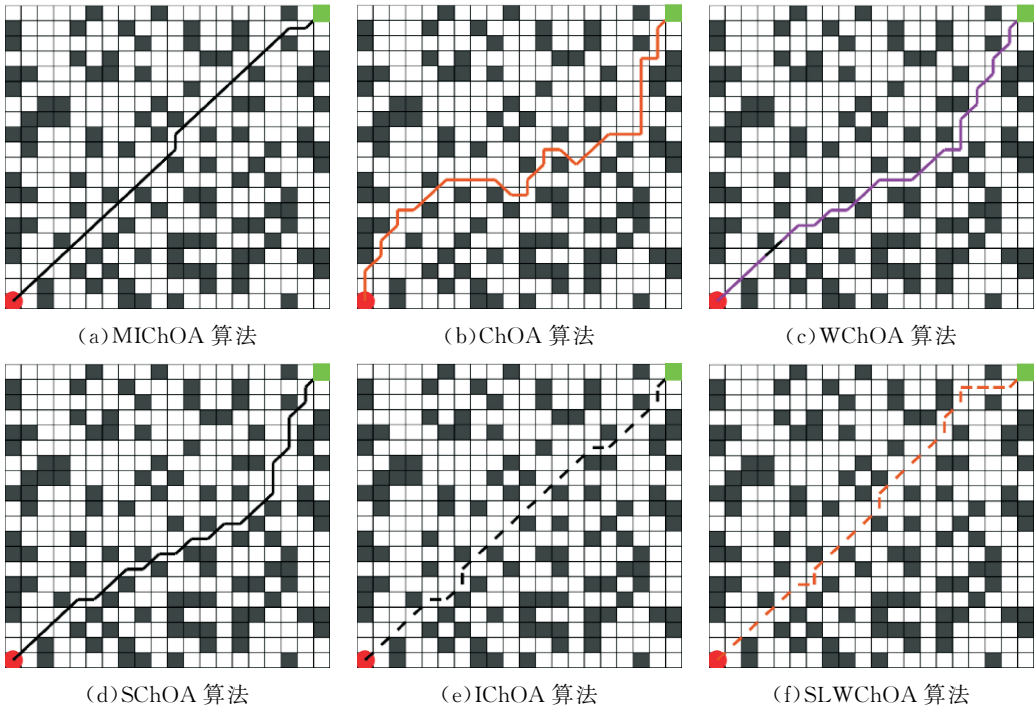


图 3 不同算法的规划路径结果对比

Fig. 3 Comparison of the results of the different algorithms for planning paths

从图 3 可以看出,传统 ChOA 算法在机器人行走过程中存在“走弯路”现象,有时会直接导致搜索停滞而陷入局部极值,这种情况会使机器人路径寻优值出现不稳定。对于 WChOA 和 SChOA 算法,出现了与 ChOA 算法类似的“绕道”情况,说明这 2 种算法的改进策略并不适合实际路径规划,缺少针对性的路径寻优策略。IChOA 和 SLWChOA 算法的路径规划地图表明,虽然改进策略在寻优过程中有所提高,但还是存在一些不必要的路径选择。只有改进后的 MIChOA 算法在具有多个障碍物的地图中,能够轻松判定路线,使其跳出局部最优。

图 4 给出了每种算法规划地图路线的收敛曲线。由图 4 可以看到:在复杂环境中,MIChOA 算法能够很快地完成收敛并达到稳定状态;其他对比算法的收敛曲线虽然出现阶梯型寻优过程,但在寻优过程中还是陷入了局部最优状态,对收敛精度影响较大;与 MIChOA 算法相比,其他算法的迭代次数较多,导致迭代速度降低,从而影响农业机器人的工作进程。

表 5 不同算法路径寻优性能对比

Table 5 Comparison of the path finding performance of different algorithms

算法	p_{best}/m	p_{worse}/m	p_{mean}/m	p_{std}/m	T_{mean}/s	T_{std}/s	$C_{gv}/次$
MIChOA	28.93	29.77	29.44	0.55	4.16	0.41	51
ChOA	30.02	42.52	40.90	4.42	9.16	0.63	361
WChOA	29.89	35.45	31.45	2.51	7.16	0.73	238
SChOA	30.01	34.74	31.67	2.59	7.89	0.80	312
IChOA	29.04	30.69	29.51	0.98	4.43	1.10	137
SLWChOA	29.23	30.97	29.96	1.09	4.68	0.89	212

相比于 ChOA 算法,MIChOA 算法的平均路径长度缩短了 28.01%,寻优时间减少了 54.58%,迭代次数减少了 85.87%,路径长度标准差与寻优时间标准差大大降低。从路径长度看,MIChOA 改进算法的最优路径、最差路径和平均路径相差不大,路径长度标准差为 0.55,在所有对比算法中最小。从运行时间看,MIChOA 算法的平均寻优耗时和耗时标准差在所有算法中最小,表明改进算法的计算复杂度小且运行时间较为稳定。从迭代次数看,MIChOA 算法远远低于其他对比算法的迭代次数,验证了 MIChOA 算法具有较快的收敛速度。综上所述,虽然其他算法也可以规划一定的路径,但其避障能力较弱,影响了求解的质量和稳定性。

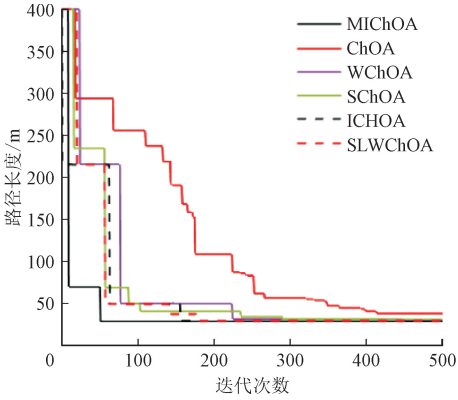


图 4 不同算法的规划路径迭代收敛曲线
Fig. 4 Optimal iteration curves of planning paths for different algorithms

4.2.2 路径规划算法性能对比分析

为了进一步分析改进算法的路径规划性能,将上述 6 种 ChOA 路径规划算法分别单独运行 30 次,统计出最优路径 p_{best} 、最差路径 p_{worse} 、平均路径 p_{mean} 、路径长度标准差 p_{std} 、平均寻优耗时 T_{mean} 、耗时标准差 T_{std} 和平均迭代次数 C_{gv} 等 7 个维度数据,如表 5 所示。

5 结 论

本文提出了一种多策略改进的黑猩猩算法 MIChOA,将其与传统的网格方法相结合,建立了一个约束优化问题来搜索最优机器人路径。选择相对优秀的其他 4 种改进 ChOA 算法,结合标准测试函数验证结果和复杂仿真实验场景下的路径规划对比结果,得出以下结论:

(1)本文所提 MIChOA 算法,针对传统 ChOA 算法路径寻优缺陷,通过引入佳点集初始化策略、正态随机余弦收敛因子和贪婪竞争机制的停滞扰动策略,加快了算法在路径寻优初始阶段的收敛速度和精度,平衡了路径规划过程中避障和开发勘探之间

的同步性。

(2) 与传统 ChOA 算法以及其也 4 种改进 ChOA 算法相比,可以看出对于一般的函数寻优, MICHOA 算法具有较好的收敛效果和寻优精度,且规划路径寻优性能的 7 项指标均表现优异。由此可知,本文所提 MICHOA 算法在复杂环境下的路径避障能力、收敛精度、路径寻优结果稳定性以及寻优时间等方面的表现均优于其他算法。

然而对于动态路径规划,该算法的优点并不明显,因此导致其应用范围有限。未来将进一步改进动态路径规划算法和轨迹优化,提高农业机械机器人的导航效率和稳定性。

参考文献:

- [1] COSTA M M, SILVA M F. A survey on path planning algorithms for mobile robots [C]//2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC). Piscataway, NJ, USA: IEEE, 2019: 1-7.
- [2] 于丰华,周传琦,杨鑫,等. 日光温室番茄采摘机器人设计与试验 [J]. 农业机械学报, 2022, 53(1): 41-49.
YU Fenghua, ZHOU Chuanqi, YANG Xin, et al. Design and experiment of tomato picking robot in solar greenhouse [J]. Transactions of the Chinese Society for Agricultural Machinery, 2022, 53(1): 41-49.
- [3] 刘勇. 果蔬大棚巡检机器人移动平台的设计及关键技术研究 [D]. 南昌: 江西农业大学, 2020.
- [4] 钟华庚, 罗高生, 王芳, 等. 一种 Halton 序列的 HDRRT 移动机器人融合规划算法 [J/OL]. 控制与决策: 1-9[2023-03-01]. <https://doi.org/10.13195/j.kzyjc.2021.1709>.
ZHONG Huageng, LUO Gaosheng, WANG Fang, et al. A Halton sequence fusion planning algorithm for HDRRT mobile robots [J/OL]. Control and Decision: 1-9[2023-03-01]. <https://doi.org/10.13195/j.kzyjc.2021.1709>.
- [5] WANG Can, CHENG Chensheng, YANG Dianyue, et al. Path planning in localization uncertain environment based on Dijkstra method [J]. Frontiers in Neurobotics, 2022, 16: 821991.
- [6] LUO Min, HOU Xiaorong, YANG Jing, et al. Surface optimal path planning using an extended Dijkstra algorithm [J]. IEEE Access, 2020, 8: 147827-147838.
- [7] 程传奇, 郝向阳, 李建胜, 等. 融合改进 A* 算法和动态窗口法的全局动态路径规划 [J]. 西安交通大学学报, 2017, 51(11): 137-143.
CHENG Chuanqi, HAO Xiangyang, LI Jiansheng, et al. Global dynamic path planning based on fusion of improved A* algorithm and dynamic window approach [J]. Journal of Xi'an Jiaotong University, 2017, 51(11): 137-143.
- [8] SUN Yicheng, ZHAO Xianliang, YU Yazhou. Research on a random route-planning method based on the fusion of the A* algorithm and dynamic window method [J]. Electronics, 2022, 11(17): 2683.
- [9] NIU Chuanhu, LI Aijuan, HUANG Xin, et al. Research on global dynamic path planning method based on improved A* algorithm [J]. Mathematical Problems in Engineering, 2021, 2021: 1-13.
- [10] YUAN Zhiheng, YANG Zhengmao, LV Lingling, et al. A bi-level path planning algorithm for multi-AGV routing problem [J]. Electronics, 2020, 9(9): 1351.
- [11] ZHU Haoran, WANG Yunhe, MA Zhiqiang, et al. A comparative study of swarm intelligence algorithms for UCAV path-planning problems [J]. Mathematics, 2021, 9(2): 1-31.
- [12] ZENG Nianyin, ZHANG Hong, CHEN Yanping, et al. Path planning for intelligent robot based on switching local evolutionary PSO algorithm [J]. Assembly Automation, 2016, 36(2): 120-126.
- [13] SONG Baoye, WANG Zidong, ZOU Lei. An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve [J]. Applied soft Computing, 2021, 100: 106960.
- [14] MIAO Changwei, CHEN Guangzhu, YAN Chengliang, et al. Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm [J]. Computers & Industrial Engineering, 2021, 156: 107230.
- [15] WU Bin, CHI Xiaonan, ZHAO Congcong, et al. Dynamic path planning for forklift AGV based on smoothing A* and improved DWA hybrid algorithm [J]. Sensors, 2022, 22(18): 7079.
- [16] 刘志强, 何丽, 袁亮, 等. 采用改进灰狼算法的移动机器人路径规划 [J]. 西安交通大学学报, 2022, 56(10): 49-60.
LIU Zhiqiang, HE Li, YUAN Liang, et al. Path planning of mobile robot based on TGWO algorithm [J]. Journal of Xi'an Jiaotong University, 2022, 56(10): 49-60.
- [17] DONG Lin, YUAN Xianfeng, YAN Bingshuo, et al. An improved grey wolf optimization with multi-strategy ensemble for robot path planning [J]. Sensors, 2022, 22(18): 6843.
- [18] KHISHE M, MOSAVI M R. Chimp optimization al-

gorithm [J]. Expert Systems with Applications, 2020, 149: 113338.

[19] KAUR M, KAUR R, SINGH N, et al. SChoA: a newly fusion of sine and cosine with chimp optimization algorithm for HLS of datapaths in digital filters and engineering applications [J]. Engineering with Computers, 2022, 38(2): 975-1003.

[20] KHISHE M, NEZHADSHAHBODAGHI M, MO-SAVI M R, et al. A weighted chimp optimization algorithm [J]. IEEE Access, 2021, 9: 158508-158539.

[21] 刘成汉, 何庆. 融合多策略的黄金正弦黑猩猩优化算法 [J/OL]. 自动化学报: 1-14[2022-11-21]. <https://doi.org/10.16383/j.aas.c210313>.
LIU Chenghan, HE Qing. Golden sine chimp optimization algorithm integrating multiple strategies [J/OL]. Acta Automatica Sinica: 1-14 [2022-11-21]. <https://doi.org/10.16383/j.aas.c210313>.

[22] 何庆, 罗仕杭. 混合改进策略的黑猩猩优化算法及其机械应用 [J]. 控制与决策, 2023, 38(2): 354-364.
HE Qing, LUO Shihang. Chimp optimization algorithm based on hybrid improvement strategy and its mechanical application [J]. Control and Decision, 2023, 38(2): 354-364.

[23] DU Nating, ZHOU Yongquan, DENG Wu, et al. Improved chimp optimization algorithm for three-dimensional path planning problem [J]. Multimedia Tools and Applications, 2022, 81(19): 27397-27422.

[24] 刘香品, 宣士斌, 刘峰. 引入佳点集和猴群翻过程的人工蜂群算法 [J]. 模式识别与人工智能, 2015, 28(1): 80-89.
LIU Xiangpin, XUAN Shibin, LIU Feng. Artificial bee colony algorithm with good point set and turn process of monkey algorithm [J]. Pattern Recognition and Artificial Intelligence, 2015, 28(1): 80-89.

[25] ABDOLLAHZADEH B, SOLEIMANIAN GHARE-HCHOPOGH F, MIRJALILI S. Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems [J]. International Journal of Intelligent Systems, 2021, 36(10): 5887-5958.

[26] NARUEI I, KEYNIA F. Wild horse optimizer: a new meta-heuristic algorithm for solving engineering optimization problems [J]. Engineering with Computers, 2022, 38(4): 3025-3056.

[27] ZHANG Bin, KANEKO M, LIM H O. Robust 2D mapping integrating with 3D information for the autonomous mobile robot under dynamic environment [J]. Electronics, 2019, 8(12): 1503.

(编辑 李慧敏 刘杨)