基于多策略融合蜣螂优化算法的工业机器人运动学参数辨识方法

许佳璐[1] 刘笑楠[1*] 李朋超[2] 刘振宇[1]
1.沈阳工业大学信息科学与工程学院,沈阳,110870
2.中国科学院沈阳自动化研究所机器人学国家重点实验室,沈阳,110016

摘要:针对蜣螂优化(DBO)算法在工业机器人运动学参数标定过程中存在的全局探索和局部开发能力不平衡、求解精度低等问题,提出了一种基于局部指数积(LPOE)运动学模型的多策略融合蜣螂优化算法(MSFDBO)。首先建立基于 LPOE 模型的运动学参数辨识模型;然后采用 Piecewise 混沌映射和精英反向学习策略进行种群初始化,得到分布更加均匀的种群;融入鱼鹰探索行为,提高 DBO 算法的全局探索能力,通过随机扰动机制扩大搜索范围,减少 DBO 算法陷入局部最优的可能性。为测试算法性能,使用 12 个基准测试函数对 MSFDBO 算法的搜索性能进行实验评估,结果表明该算法具有良好的寻优性能。对 4 台 T6A-19 型工业机器人的运动学参数进行辨识并补偿验证,实验结果表明,绝对位置平均误差、均方根平均误差分别降低了 85.47%、83.92%。

关键词:运动学参数标定;蜣螂优化算法;精英反向学习;鱼鹰探索行为;随机扰动机制
中图分类号:TP242.2
DOI:10.3969/j.issn.1004-132X.2025.02.012 **开放科学(资源服务)标识码(OSID)**:

Kinematics Parameter Identification for Industrial Robots Based on Multi-strategy Fusion DBO Algorithm

XU Jialu[1] LIU Xiaonan[1*] LI Pengchao[2] LIU Zhenyu[1]
1.School of Information Science and Engineering, Shenyang University of Technology, Shenyang, 110870
2.State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016

Abstract: Aiming at the DBO algorithm's imbalance between global exploration and local exploitation capabilities and low solution accuracy in the calibrating processes for kinematics parameters of industrial robots, a multi-strategy fusion(MSFDBO) algorithm was presented based on local product of exponential(LPOE) kinematics model. Firstly, a kinematics parameter identification model was established based on the LPOE model. Secondly, Piecewise chaotic mapping and elite inverse learning strategy were used for population initialization to obtain a more uniformly distributed population, incorporating the exploration behavior of the osprey to enhance the global exploration ability of the DBO algorithm, and expanding the search range through the stochastic perturbation mechanism to reduce the possibility of the DBO algorithm falling into a local optimum. To test the performance of the algorithm, the search performance of the MSFDBO algorithm was experimentally evaluated using 12 benchmark test functions. The results show that the algorithm performs well in terms of optimization. The compensation of kinematic parameters was identifed and verified for four T6A-19 industrial robots. The experimental results show that the mean absolute position errors are reduced by an average of 85.47% and the root mean square errors are reduced by an average of 83.92%.

Key words: kinematics parameter calibration; dung beetle optimization(DBO) algorithm; elite opposition-based learning; osprey exploratory behavior; stochastic perturbation mechanism

0 引言

在机器人制造系统中,机器人的绝对定位精度是工业机器人应用离线编程方法时面临的一个主要问题,由于机器人固有的运动学误差,如生产

收稿日期:2024-01-24
基金项目:国家重点研发计划(2021YFB3201600);辽宁省自然科学基金(20180520022)

过程中的加工误差、装配误差和使用过程中的磨损等，机器人的实际运动学参数与其理论运动学参数不同[1]，而通过参数标定可以有效提高工业机器人的绝对定位精度[2]。机器人运动学参数标定分为四个步骤：建模、测量、辨识参数误差以及补偿模型误差[3]。传统的 D-H 模型存在相邻关节轴线平行或垂直时出现奇异点的问题，基于指数积(product of exponential，POE)的标定模型误差参数完备、变化平滑、无模型奇异性，但存在微分误差模型相对复杂的问题[4]。本文采用局部指数积(local product of exponential，LPOE)模型[5]进行机器人运动学建模，可在保持误差参数完备且连续的同时避免微分误差模型相对复杂的问题。

辨识参数误差需根据建立的误差模型构建多参数优化目标函数，通过传统优化算法或智能优化算法进行求解[6]。传统的辨识算法主要有最小二乘法[7]、Levenberg-Marquardt 法[8]、扩展卡尔曼滤波法[9]、极大似然法[10]等，这些方法既受雅可比矩阵奇异性的影响，又受迭代初值选取的影响[11]，智能优化算法能够较好地解决以上两个问题。近年来，有许多学者应用智能优化算法解决工业机器人运动学参数标定问题，如温秀兰等[12]提出了基于拟随机序列产生初始位置的改进乌鸦搜索算法(improved crow search algorithm，ICSA)用于标定 Staubli TX60 机器人几何参数。姜一舟等[13]提出了一种改进差分进化算法，使用 Metropolis 接受准则，以获得更好的收敛性，同时提出种群多样性评价函数和二次变异操作，以避免陷入局部优化。乔贵方等[14]提出了一种基于天牛须搜索和粒子群优化(beetle antennea search-particle swarm optimization，BAS-PSO)算法的运动学参数辨识方法，并通过实验验证了不同误差模型构建方法的辨识精度和稳定性。寇斌等[15]提出了一种基于两段式的动态粒子群算法，用以解决工业机器人几何误差标定问题中存在的收敛速度慢的问题。乔贵方等[16]先利用改进灰狼优化算法进行参数误差粗辨识，再通过 Levenberg-Marquard 算法进行参数误差的精辨识。

2022 年由 XUE 等[17]提出的蜣螂优化(dung beetle optimizer，DBO)算法在经典的复杂函数优化方面已被证明在求解精度和收敛速度等性能上均优于粒子群优化算法和遗传算法等经典算法，并成功应用于瑞雷波频散曲线反演[18]、光伏阵列故障诊断[19]等实际工程问题中。然而，没有一种元启发式算法适合解决所有的优化问题，潘劲成等[20]通过融合改进正弦算法、混沌映射和变异算子，提高 DBO 算法的全局探索能力。郭琴等[21]在 DBO 算法中改进雏球和偷窃蜣螂对局部最优解和全局最优解的接受程度，以及融合麻雀搜索算法中的追随者位置更新机制和柯西高斯变异策略，以提升算法的寻优能力。然而当 DBO 算法应用于工业机器人运动学参数标定时，存在全局探索和局部开发能力不平衡的问题，搜索空间的过快减小会使种群的个体快速同化，导致算法迭代求解出的运动学误差参数陷入局部最优解。

本文提出一种多策略融合蜣螂优化算法(multi-strategy fusion dung beetle optimizer，MSFDBO)用以标定基于 LPOE 运动学模型的工业机器人运动学参数，通过 Piecewise 混沌映射和精英反向学习策略以及鱼鹰探索行为提升 DBO 算法的全局探索能力，利用随机扰动机制减少 DBO 算法陷入局部最优的可能性，平衡 DBO 算法的全局探索和局部开发能力，并以新松 T6A-19 型机器人为例，验证 MSFDBO 算法辨识工业机器人运动学参数的有效性和可行性。

1　六轴工业机器人运动学模型

本文以六自由度工业机器人为基础建立运动学参数标定系统，如图 1 所示。

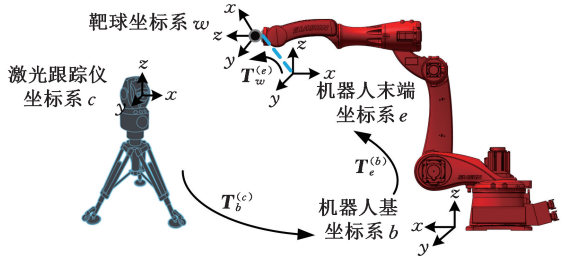

图 1　运动学参数标定系统

Fig.1　Kinematic parameter calibration system

1.1　机器人 LPOE 模型

在 LPOE 模型中，各关节旋量被描述在局部连杆坐标系下，其关节 i 与关节 $i-1$ 之间的坐标转换矩阵为

$$\boldsymbol{T}_i^{(i-1)}(\theta_i)=\exp(\hat{\boldsymbol{s}}_i\theta_i)\boldsymbol{T}_i^{(i-1)}(0) \tag{1}$$

其中，$\boldsymbol{T}_i^{(i-1)}(0)\in SE(3)$ 为坐标系$\{i\}$相对于坐标系$\{i-1\}$的初始位姿；θ_i 为第 i 个关节角度变量；$\hat{\boldsymbol{s}}_i$ 为关节 i 的旋量在坐标系$\{i-1\}$下的表示，有

$$\hat{\boldsymbol{s}}_i=\begin{bmatrix}\hat{\boldsymbol{w}}_i & \boldsymbol{v}_i\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix} \tag{2}$$

$$\hat{\boldsymbol{w}}_i=\begin{bmatrix}0 & -w_{iz} & w_{iy}\\ w_{iz} & 0 & -w_{ix}\\ -w_{iy} & w_{ix} & 0\end{bmatrix} \tag{3}$$

$$\boldsymbol{s}_i = [\boldsymbol{w}_i \quad \boldsymbol{v}_i]^{\mathrm{T}} \tag{4}$$

其中，$\boldsymbol{w}_i = (w_{ix}, w_{iy}, w_{iz})^{\mathrm{T}}$ 表示各关节轴线在 $\{i\}$ 坐标系中的方向向量；$\boldsymbol{s}_i$ 表示关节旋量的旋量坐标；$\boldsymbol{v}_i = (v_{ix}, v_{iy}, v_{iz})^{\mathrm{T}} \in \mathbf{R}^{3\times 1}$。

根据式(1)得到 n 自由度串联机器人 LPOE 模型：

$$\begin{aligned}\boldsymbol{T}_{\mathrm{tool}}^{(0)}(\theta_1,\theta_2,\cdots,\theta_n) = \\ \exp(\hat{\boldsymbol{s}}_1\theta_1)\boldsymbol{T}_1^{(0)}(0)\exp(\hat{\boldsymbol{s}}_2\theta_2)\boldsymbol{T}_2^{(1)}(0)\cdots \\ \exp(\hat{\boldsymbol{s}}_n\theta_n)\boldsymbol{T}_n^{(n-1)}(0)\exp(\hat{\boldsymbol{s}}_{\mathrm{tool}})\boldsymbol{T}_{\mathrm{tool}}^{(n)}\end{aligned} \tag{5}$$

其中，$\boldsymbol{T}_{\mathrm{tool}}^{(n)} \in SE(3)$ 为最后一个连杆坐标系到工具坐标系的位姿变换矩阵。

1.2 绝对位置误差模型

在实际测量过程中，将激光跟踪仪、机器人和靶球构造成统一的辨识系统。如图 1 所示，$\boldsymbol{T}_b^{(c)}$ 表示激光跟踪仪坐标系 $\{c\}$ 相对于基坐标系 $\{b\}$ 的变换矩阵，$\boldsymbol{T}_e^{(b)}$ 表示机器人末端坐标系 $\{e\}$ 相对于基坐标系 $\{b\}$ 的变换矩阵，$\boldsymbol{T}_w^{(e)}$ 表示靶球坐标系 $\{w\}$ 相对于机器人末端坐标系 $\{e\}$ 的变换矩阵。靶球坐标系相对于机器人基坐标系的变换矩阵 $\boldsymbol{T}_{\mathrm{tool}}^{0}$ 表示为

$$\boldsymbol{T}_{\mathrm{tool}}^{(0)} = \boldsymbol{T}_e^{(b)}\boldsymbol{T}_w^{(e)} = \exp(\hat{\boldsymbol{s}}_{\mathrm{tool}})\boldsymbol{T}_{\mathrm{tool}}^{(6)}\prod_{i=1}^{6}\exp(\hat{\boldsymbol{s}}_i\theta_i)\boldsymbol{T}_i^{(i-1)} \tag{6}$$

其中，$\boldsymbol{T}_{\mathrm{tool}}^{(6)}$ 为工具坐标系相对于机器人末端坐标系的齐次变换矩阵。激光跟踪仪坐标系下测量点的实际值 $\boldsymbol{P}_r$ 转换到机器人基坐标系下为

$$\boldsymbol{p} = \boldsymbol{T}_b^{(c)}\boldsymbol{P}_r \tag{7}$$

在机器人运动学参数辨识时，只考虑机器人末端的位置变化，改变机器人各关节的转动角度，可以得到靶球相对于机器人基坐标系的位置信息。将机器人运动学参数和位置关系表示为高维非线性优化问题：

$$F = \| h(\boldsymbol{x}+\boldsymbol{\varepsilon},\theta_i) - \boldsymbol{p} \|_2 \tag{8}$$

其中，F 为机器人末端第 i 个点的理论值与实际值之间的距离；$\boldsymbol{x}$ 为机器人运动学参数；$\boldsymbol{\varepsilon}$ 为待标定的运动学参数误差，表示为

$$\boldsymbol{\varepsilon} = [\Delta\boldsymbol{w}_i \quad \Delta\boldsymbol{v}_i] \tag{9}$$

其中，$\Delta\boldsymbol{w}_i = [\Delta w_{ix} \quad \Delta w_{iy} \quad \Delta w_{iz}]$，$\Delta\boldsymbol{v}_i = [\Delta v_{ix} \quad \Delta v_{iy} \quad \Delta v_{iz}]$($i = 1,2,\cdots,6,\mathrm{tool}$)，共计 42 项待标定的运动学参数。

2 DBO 算法

在 DBO 算法中，每个蜣螂种群由滚球蜣螂种群、育雏蜣螂种群、小蜣螂种群和偷窃蜣螂种群构成，根据优化问题按一定的比例进行分配。

2.1 滚球蜣螂

蜣螂在滚动过程中需要通过天体线索(太阳位置或风向等)导航，以保持粪球沿直线滚动。滚球蜣螂的位置更新表示为

$$\left.\begin{aligned}x_i^{(t+1)} &= x_i^{(t)} + \alpha k x_i^{(t-1)} + q\Delta x \\ \Delta x &= |\, x_i^{(t)} - x^{\mathrm{w}} \,|\end{aligned}\right\} \tag{10}$$

式中：t 为当前迭代次数；$x_i^{(t)}$ 为第 t 次迭代时第 i 只滚球蜣螂的位置信息；$k \in (0,0.2]$ 为指示偏转系数的常数值；q 为(0,1)区间内的常数值；α 为被分配 -1 或 1 的自然系数；x^{w} 为全局最差位置；Δx 用于模拟光强的变化。

当蜣螂遇到障碍物无法前进时，使用正切函数模仿舞蹈行为来获得新的滚动方向，以获得新的路线。跳舞蜣螂的位置更新公式为

$$x_i^{(t+1)} = x_i^{(t)} + \tan\varphi \,|\, x_i^{(t)} - x_i^{(t-1)} \,| \tag{11}$$

其中，φ 为偏转角，$\varphi \in [0,\pi]$。如果 φ 等于 0、$\pi/2$ 或 π，则跳舞蜣螂的位置不会更新。

2.2 育雏蜣螂

在自然界中，粪球被滚到安全的地方并被蜣螂隐藏。为了给后代提供一个安全的环境，选择合适的产卵地点对蜣螂来说至关重要。通过一种边界选择策略模拟雌蜣螂产卵的区域，该区域定义为

$$\left.\begin{aligned}{}^{\mathrm{L}}b^* &= \max(x^*(1-R), {}^{\mathrm{L}}b) \\ {}^{\mathrm{U}}b^* &= \min(x^*(1+R), {}^{\mathrm{U}}b) \\ R &= 1 - t/T_{\max}\end{aligned}\right\} \tag{12}$$

其中，x^* 为当前的局部最优位置；${}^{\mathrm{L}}b^*$ 和 ${}^{\mathrm{U}}b^*$ 分别为产卵区域的下界和上界；$T_{\max}$ 为最大迭代次数；${}^{\mathrm{L}}b$ 和 ${}^{\mathrm{U}}b$ 分别为优化问题的下界和上界。

育雏蜣螂位置更新公式为

$$B_i^{(t+1)} = x^* + \boldsymbol{b}_1(B_i^{(t)} - {}^{\mathrm{L}}b^*) + \boldsymbol{b}_2(B_i^{(t)} - {}^{\mathrm{U}}b^*) \tag{13}$$

其中，$B_i^{(t)}$ 为第 t 次迭代时第 i 只育雏蜣螂的位置信息；$\boldsymbol{b}_1$ 和 $\boldsymbol{b}_2$ 为大小为 $1\times D$ 的两个独立随机向量；D 为优化问题的维数。

2.3 小蜣螂

一些小蜣螂从地里爬出来寻找食物，需要建立最佳觅食区来引导觅食的小蜣螂，最佳觅食区域的边界定义为

$$\left.\begin{aligned}{}^{\mathrm{L}}_{\mathrm{B}}b &= \max(x^{\mathrm{B}}(1-R), {}^{\mathrm{L}}b) \\ {}^{\mathrm{U}}_{\mathrm{B}}b &= \min(x^{\mathrm{B}}(1+R), {}^{\mathrm{U}}b)\end{aligned}\right\} \tag{14}$$

式中：x^{B} 为全局最佳位置；${}^{\mathrm{L}}_{\mathrm{B}}b$、${}^{\mathrm{U}}_{\mathrm{B}}b$ 分别为最佳觅食区域的下限和上限。

小蜣螂的位置更新公式为

$$x_i^{(t+1)} = x_i^{(t)} + C_1(x_i^{(t)} - {}^{\mathrm{L}}_{\mathrm{B}}b) + C_2(x_i^{(t)} - {}^{\mathrm{U}}_{\mathrm{B}}b) \tag{15}$$

式中：$x_i^{(t)}$ 为第 t 次迭代时第 i 只小蜣螂的位置信息；C_1 为服从正态分布的随机数；C_2 为随机数，$C_2 \in (0,1)$。

2.4 偷窃蜣螂

偷窃蜣螂会从其他蜣螂处偷取粪球，在最佳食物来源进行偷窃时，偷窃蜣螂位置信息更新公

式为

$$x_i^{(t+1)} = x^{\mathrm{B}} + s\boldsymbol{g}(|x_i^{(t)} - x^*| + |x_i^{(t)} - x^{\mathrm{B}}|) \quad (16)$$

其中，$x_i^{(t)}$ 表示第 t 次迭代时第 i 只偷窃蜣螂的位置信息；$\boldsymbol{g}$ 是遵循正态分布的大小为 $1\times D$ 的随机向量；s 表示常数值。

3　DBO 算法的改进策略

3.1　Piecewise 混沌映射和精英反向学习策略

由于蜣螂初始种群的质量在很大程度上影响到整体的寻优性能，为避免随机生成种群带来的不均匀性，降低初始种群多样性，利用 Piecewise 映射在[0,1]内生成混沌序列，再将其映射到算法的搜索空间，使得蜣螂种群能够更好地遍布整个搜索空间。Piecewise 映射公式如下：

$$o_{i,j+1} = \begin{cases} o_{i,j}/P & 0 \leqslant o_{i,j} < P \\ \dfrac{o_{i,j}-P}{0.5-P} & P \leqslant o_{i,j} < 0.5 \\ \dfrac{1-P-o_{i,j}}{0.5-P} & 0.5 \leqslant o_{i,j} < 1-P \\ (1-o_{i,j})/P & 1-P \leqslant o_{i,j} < 1 \end{cases} \quad (17)$$

其中，$o_{i,j}$ 为(0,1)内的随机数；P 为常数且 $P=0.3$；i 为种群数量；j 为混沌变量序号且 $j=1,2,\cdots,D$。将混沌变量 $o_{i,j+1}$ 映射到搜索空间，得到种群初始解 $x_{i,j}$：

$$x_{i,j} = {}^{\mathrm{L}}b_j + o_{i,j+1}({}^{\mathrm{U}}b_j - {}^{\mathrm{L}}b_j) \quad (18)$$

其中，${}^{\mathrm{U}}b_j$ 为搜索空间上界；${}^{\mathrm{L}}b_j$ 为搜索空间下界；$x_{i,j}$ 为第 i 个种群个体在 j 维空间中的位置。

同时，利用精英反向学习策略[22]生成反向种群，以提高种群的收敛速度和全局搜索能力。该策略首先根据初始种群选择靠近最优位置的个体作为精英个体，并将其用于生成反向种群。通过这种方式，反向种群中的个体距离最优解更近，有助于加速收敛过程。此外，精英反向学习策略还能够搜索更多有效区域，增加种群的多样性，从而提高算法的全局搜索能力。

设当前种群中的一般个体对应的自身极值点为精英个体 $x'_{i,j}$，其反向对应解 $x^*_{i,j}$ 定义为

$$x^*_{i,j} = rand\cdot({}^{\mathrm{L}}b_m + {}^{\mathrm{U}}b_m) - x'_{i,j} \quad (19)$$

$${}^{\mathrm{L}}b_m = \min(x'_{i,j}) \qquad {}^{\mathrm{U}}b_m = \max(x'_{i,j})$$

其中：${}^{\mathrm{L}}b_m$ 和 ${}^{\mathrm{U}}b_m$ 分别为动态边界的下界和上界，$rand$ 为(0,1)内的随机数。

3.2　鱼鹰探索行为

针对 DBO 算法中滚球蜣螂全局探索能力弱的问题，引入鱼鹰探索行为策略，鱼鹰因其强大的视力能够探测到水下鱼类的位置，确定鱼的位置并进行捕食，使得鱼鹰探索行为在识别最优区域和逃离局部最优方面具有极强的探索能力。

在鱼鹰算法(osprey optimization algorithm，OOA)[23]中，对于每只鱼鹰，搜索空间中具有更好目标函数值的其他鱼鹰的位置被视为水下鱼类。每只鱼鹰的鱼集合位置使用下式指定：

$$P_i^{\mathrm{F}} = \{x_k \mid k \in \{1,2,\cdots,N\} \wedge F_k < F_i\} \cup \{x_{\mathrm{best}}\} \quad (20)$$

式中：P_i^{F} 为第 i 只鱼鹰的鱼的位置集合；x_{best} 为最佳鱼鹰的位置。

鱼鹰随机检测其中一条鱼的位置并攻击它。基于鱼鹰向鱼的运动模拟，计算相应鱼鹰的新位置：

$$\left.\begin{aligned} x_{i,j}^{\mathrm{P1}} &= x_{i,j} + r_{i,j}(F_{i,j}^{\mathrm{S}} - I_{i,j}x_{i,j}) \\ x_{i,j}^{\mathrm{P1}} &= \begin{cases} x_{i,j}^{\mathrm{P1}} & {}^{\mathrm{L}}b_j \leqslant x_{i,j}^{\mathrm{P1}} \leqslant {}^{\mathrm{U}}b_j \\ {}^{\mathrm{L}}b_j & x_{i,j}^{\mathrm{P1}} < {}^{\mathrm{L}}b_j \\ {}^{\mathrm{U}}b_j & x_{i,j}^{\mathrm{P1}} > {}^{\mathrm{U}}b_j \end{cases} \end{aligned}\right\} \quad (21)$$

如果新位置的目标函数值更优，则根据下式替换鱼鹰的先前位置：

$$X_i = \begin{cases} X_i^{\mathrm{P1}} & F_i^{\mathrm{P1}} < F_i \\ X_i & \text{其他} \end{cases} \quad (22)$$

式中：X_i^{P1} 为鱼鹰探索阶段时第 i 只鱼鹰的新位置，$x_{i,j}^{\mathrm{P1}}$ 为其第 j 维的新位置；F_i^{P1} 为 X_i^{P1} 对应的适应度值；F_i^{S} 为第 i 只鱼鹰所选的鱼，$F_{i,j}^{\mathrm{S}}$ 为其第 j 个维度；$r_{i,j}$ 为[0,1]之间的随机数；$I_{i,j}$ 为集合{1,2}中的随机数。

当跳舞蜣螂遇到障碍物，偏转角度取值为 0、π/2 或 π 时，跳舞蜣螂位置更新陷入停滞，容易导致局部最优停滞。因此，采用指数因子对最佳位置周围进行搜索：

$$x_i^{(t+1)} = x^{\mathrm{B}}\left(\frac{2}{\exp\left(\dfrac{4t}{(randn\cdot T_{\max})^2}\right)}\right) \quad (23)$$

式中：$randn$ 为(0,1)之间符合标准正态分布的伪随机数。

3.3　随机扰动机制

虽然蜣螂的产卵和觅食区域随迭代次数动态调整，但是线性减少的策略难以表征其在繁殖与觅食时的复杂情况，因此，采用非线性边界收敛因子在前期对全局可行区域进行更为广泛的搜索，而在后期更加注重在最优解附近的局部开发，加快其收敛速度。非线性边界收敛因子 R 的表达式为

$$R = 0.5 + 0.5\sin(0.5\pi + \pi t/T_{\max}) \quad (24)$$

在 DBO 算法迭代的后期，育雏蜣螂和小蜣螂个体的快速同化，蜣螂群体迅速聚集到当前的最优位置附近，容易出现局部最优解和搜索停滞。通过随机扰动机制对蜣螂个体进行干扰，增大最优解的勘探空间，实现种群个体多维信息的重组和突变，提高蜣螂个体质量与种群多样性，增大 DBO 算法跳出局部最优解的概率。使用 Lévy 飞行函数更新当前蜣螂个体的位置生成新解来反映

蜣螂的游走轨迹。

$L(\lambda)$ 表示服从 Lévy 分布，Lévy 飞行步长计算公式为

$$st = \mu / |v|^{1/\varepsilon} \tag{25}$$

其中，μ，v 服从标准正态分布，其定义如下：

$$\left.\begin{aligned}\mu &\sim N(0,\sigma_\mu^2)\\ v &\sim N(0,\sigma_v^2)\end{aligned}\right\} \tag{26}$$

$$\sigma_\mu = \left[\frac{\Gamma(1+\varepsilon)\cdot\sin\frac{\pi\varepsilon}{2}}{\varepsilon\Gamma(1+\frac{\varepsilon}{2})\cdot 2^{(\varepsilon-1)/2}}\right]^{1/\varepsilon} \tag{27}$$

$$\sigma_v = 1 \qquad \varepsilon = 1.5$$

由于 Lévy 飞行具有步长距离长短相间和跳跃方向多变的特点，可以在相应范围增强蜣螂跳出局部最优的能力，但也可能会因跳跃太大而导致最优蜣螂个体位置信息的丢失，故在融合 Lévy 飞行特征的同时，以一定概率引入自适应 t 分布扰动算子对蜣螂最佳个体进行扰动，采用 $i_{ter} = \exp(4(t/T_{max})^2)$ 为 t 分布的自由度参数，对育雏蜣螂和小蜣螂觅食行为进行扰动，使得蜣螂算法在迭代前期具有较好的全局开发能力，在迭代后期具有良好的局部探索能力，并提高算法的收敛速度。改进后的育雏蜣螂位置更新公式为

$$B_i^{(t+1)} = \begin{cases} x^* + \gamma L(\lambda)(B_i^{(t)} - x^{\mathrm{B}}) + b_1(B_i^{(t)} - {}^{\mathrm{L}}b^*) + b_2(B_i^{(t)} - {}^{\mathrm{U}}b^*) & r \geqslant 0.5 \\ x^* + x^* t(i_{ter}) & r < 0.5 \end{cases} \tag{28}$$

改进后的小蜣螂位置更新公式为

$$x_i^{(t+1)} = \begin{cases} x_i^{(t)} + \gamma L(\lambda)(x_i^{(t)} - x^{\mathrm{B}}) + C_1(x_i^{(t)} - {}^{\mathrm{L}}_{\mathrm{B}}b) + C_2(x_i^{(t)} - {}^{\mathrm{U}}_{\mathrm{B}}b) & r \geqslant 0.5 \\ x^{\mathrm{B}} + x^{\mathrm{B}} t(i_{ter}) & r < 0.5 \end{cases} \tag{29}$$

$$\gamma = w_{min} + (w_{max} - w_{min})t/T_{max}$$

其中，w_{min} 取 0.01；w_{max} 取 0.5。

3.4 MSFDBO 算法流程

本文提出的 MSFDBO 算法的实现步骤如下，算法流程图见图 2。

1）设置种群规模、最大迭代次数等参数，根据 Piecewise 混沌映射初始化蜣螂种群个体的位置 $x_i(i=1,2,\cdots,n)$；

2）根据精英反向学习策略更新蜣螂个体的位置；

3）根据适应度函数计算每种蜣螂位置的目标函数值，得到当前的最优函数值和相应的最优个体位置；

4）令 $\delta = rand(1)$，若 $\delta < 0.8$，根据鱼鹰搜索策略更新滚球蜣螂种群位置，否则根据式(23)更新跳舞蜣螂种群位置，计算蜣螂个体位置的目标函数值；

5）根据式(28)、式(29)、式(16)分别更新育雏蜣螂、小蜣螂和偷窃蜣螂种群的位置，计算蜣螂个体位置的目标函数值，与上一代最优函数值进行比较。若较优，则更新当前种群最优个体位置；

6）当 $t > T_{max}$ 成立，执行步骤 7），否则跳转至步骤 2）；

7）输出全局最优个体位置和最优函数值。

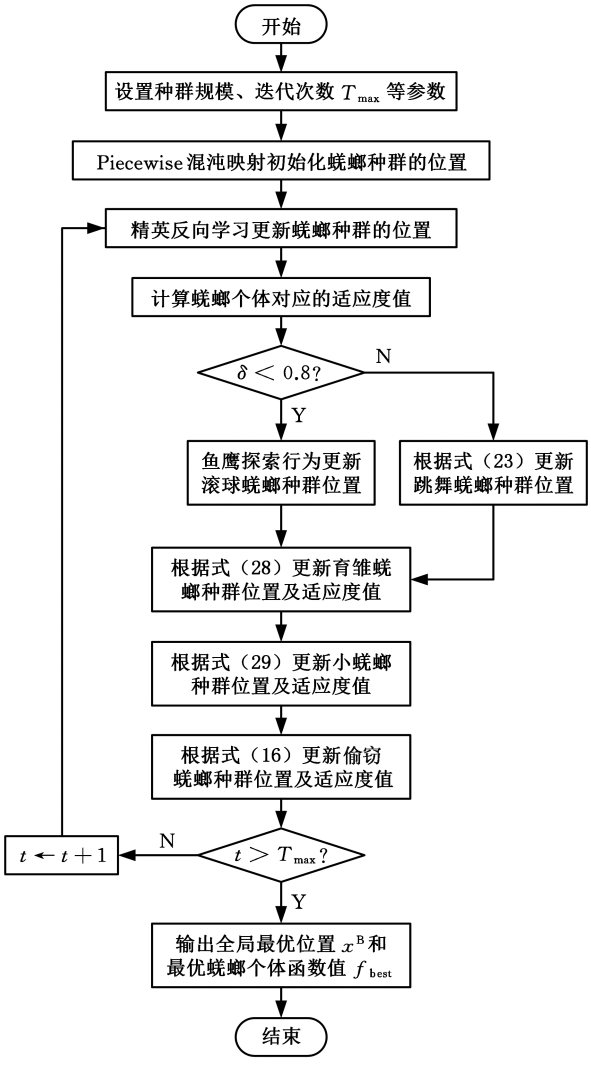

图 2 MSFDBO 算法流程图

Fig.2 MSFDBO algorithm flowchart

4 实验验证

4.1 改进策略有效性分析

为了验证本文提出的不同改进策略的独立有效性，将 DBO 与 3 种改进策略分别融合得到 3 种不同的算法：①在 DBO 中加入 Piecewise 混沌映射和精英反向学习策略得到 PDBO；②在 DBO 中采用鱼鹰探索行为得到 ODBO；③在 DBO 中加入随机扰动机制得到 RDDBO。

将 MSFDBO 与 PDBO、ODBO、RDDBO、

DBO 等算法对表 1 中的 12 个基准测试函数[24]进行寻优实验。同时利用 30 次独立实验的均值结果消除实验偶然性，不同改进策略算法的实验统计结果见表 2。另外，表 2 中还将本文算法与文献[20]中的 MSADBO 算法和文献[21]中的 MIDBO 两种改进 DBO 算法进行对比。其中，F_1～F_4 为单峰测试函数，F_5～F_8 为多峰测试函数，F_9～F_{12} 为固定维多峰测试函数。通过在 12 个基准测试函数中比较均值(Mean)和标准差(Std)来验证所提出的 MSFDBO 算法的有效性，Mean 反映了算法的精确性，Std 反映了算法的稳定性，加粗数据表示最优值。

表 1　基准测试函数

Tab.1　Benchmark function

函数	名称	维数	搜索范围	最优值
F_1	Schwefel's Problem 1.2	30	[−100,100]	0
F_2	Schwefel's Problem 2.21	30	[−100,100]	0
F_3	Step Function	30	[−100,100]	0
F_4	Quartic Function with Noise	30	[−1.28,1.28]	0
F_5	Generalized Rastrigin's Function	30	[−5.12,5.12]	0
F_6	Ackley's Function	30	[−32,32]	0
F_7	Generalized Griewank's Function	30	[−600,600]	0
F_8	Generalized Penalized Function	30	[−50,50]	0
F_9	Kowalik's Function	4	[−5,5]	0.000 307 5
F_{10}	Six-Hump Camel-Back Function	2	[−5,5]	−1.031 628 5
F_{11}	Hartman's Family	3	[0,1]	−3.86
F_{12}	Shekel's Family	4	[0,10]	−10.1532

表 2　不同改进策略的基准测试函数结果

Tab.2　Benchmarking function results for various improvement strategies

基准函数	统计值	DBO	PDBO	ODBO	RDDBO	MSADBO[20]	MIDBO[21]	MSFDBO
F_1	Mean	2.56×10^{-65}	2.88×10^{-81}	**0**	**0**	**0**	**0**	**0**
	Std	1.40×10^{-64}	1.58×10^{-80}	**0**	**0**	**0**	**0**	**0**
F_2	Mean	1.03×10^{-45}	6.32×10^{-51}	3.07×10^{-280}	2.51×10^{-211}	2.49×10^{-278}	**0**	**0**
	Std	5.65×10^{-45}	3.46×10^{-50}	**0**	**0**	**0**	**0**	**0**
F_3	Mean	6.61×10^{-4}	5.89×10^{-4}	1.34×10^{-5}	5.14×10^{-5}	4.25×10^{-2}	2.32×10^{-8}	$\mathbf{1.42\times10^{-8}}$
	Std	1.35×10^{-3}	1.86×10^{-3}	2.74×10^{-5}	6.08×10^{-5}	8.97×10^{-2}	1.11×10^{-7}	$\mathbf{6.25\times10^{-8}}$
F_4	Mean	1.49×10^{-3}	2.02×10^{-3}	4.87×10^{-4}	7.81×10^{-4}	$\mathbf{9.61\times10^{-5}}$	2.95×10^{-4}	2.56×10^{-4}
	Std	1.22×10^{-3}	1.18×10^{-3}	3.54×10^{-4}	7.58×10^{-4}	$\mathbf{6.78\times10^{-5}}$	2.17×10^{-4}	2.21×10^{-4}
F_5	Mean	5.64×10^{-1}	**0**	**0**	**0**	**0**	**0**	**0**
	Std	2.33	**0**	**0**	**0**	**0**	**0**	**0**
F_6	Mean	1.01×10^{-15}	$\mathbf{8.88\times10^{-16}}$	$\mathbf{8.88\times10^{-16}}$	$\mathbf{8.88\times10^{-16}}$	$\mathbf{8.88\times10^{-16}}$	$\mathbf{8.88\times10^{-16}}$	$\mathbf{8.88\times10^{-16}}$
	Std	6.49×10^{-16}	**0**	**0**	**0**	**0**	**0**	**0**
F_7	Mean	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	Std	**0**	**0**	**0**	**0**	**0**	**0**	**0**
F_8	Mean	3.56×10^{-3}	3.00×10^{-5}	2.45×10^{-7}	3.61×10^{-6}	1.46×10^{-3}	2.03×10^{-9}	$\mathbf{4.64\times10^{-10}}$
	Std	1.89×10^{-2}	7.76×10^{-5}	4.78×10^{-7}	6.84×10^{-6}	2.98×10^{-3}	6.65×10^{-9}	$\mathbf{1.08\times10^{-9}}$
F_9	Mean	8.13×10^{-4}	3.82×10^{-4}	6.17×10^{-4}	6.28×10^{-4}	3.73×10^{-4}	3.19×10^{-4}	$\mathbf{3.10\times10^{-4}}$
	Std	4.28×10^{-4}	1.90×10^{-4}	3.16×10^{-4}	3.20×10^{-4}	1.21×10^{-4}	1.05×10^{-5}	$\mathbf{1.01\times10^{-6}}$
F_{10}	Mean	**−1.03**	**−1.03**	**−1.03**	**−1.03**	**−1.03**	**−1.03**	**−1.03**
	Std	5.68×10^{-16}	1.09×10^{-7}	6.32×10^{-16}	5.11×10^{-12}	3.81×10^{-9}	3.49×10^{-5}	$\mathbf{5.45\times10^{-16}}$
F_{11}	Mean	**−3.86**	**−3.86**	**−3.86**	**−3.86**	**−3.86**	−3.26	**−3.86**
	Std	2.72×10^{-3}	2.40×10^{-3}	6.74×10^{-5}	2.62×10^{-15}	6.68×10^{-7}	7.65×10^{-2}	$\mathbf{2.60\times10^{-15}}$
F_{12}	Mean	−7.05	$\mathbf{-1.02\times10^{1}}$	−6.59	−9.89	-1.01×10^{1}	$\mathbf{-1.02\times10^{1}}$	$\mathbf{-1.02\times10^{1}}$
	Std	2.59	$\mathbf{5.65\times10^{-15}}$	2.63	1.37	5.23×10^{-2}	6.59×10^{-3}	5.96×10^{-15}

首先是 DBO 算法与 PDBO、ODBO、RDDBO、MSFDBO 算法的对比分析，由表 2 分析可知，PDBO 算法在函数 F_1、F_2、F_5、F_6、F_8、F_9 和 F_{12} 中的均值和标准差小于 DBO 算法的相应值，说明 Piecewise 混沌映射和精英反向学习策略可以提高群体的多样性，增强算法的全局搜索能力，在函数 F_3 中，PDBO 算法的均值小于 DBO 算法，在函数 F_4 和 F_{11} 中，PDBO 算法的标准差小于 DBO 算法，在函数 F_{10} 中，PDBO 算法的标准差大于 DBO 算法。ODBO 算法在函数 F_1～F_6、F_8、F_9 中的均值和标准差小于 DBO 算法的相应值，说明在 DBO 算法中引入鱼鹰探索行为和指数因子可以提高全局搜索能力，并能提高算法的准确性和稳定性，仅在函数 F_{10} 中，ODBO 算法的

标准差略大于 DBO 算法，在函数 F_{11} 中，ODBO 算法的标准差小于 DBO 算法，在函数 F_{12} 中，ODBO 算法的寻优能力差于 DBO 算法。RDDBO 算法在函数 F_1～F_6、F_8、F_9 和 F_{12} 中的均值和标准差小于 DBO 算法，说明在 DBO 算法中引入随机扰动机制在迭代前期具有较好的全局开发能力，在迭代后期具有良好的局部探索能力，并能提高算法的收敛速度，仅在函数 F_{10} 中，RDDBO 算法的标准差大于 DBO 算法，在函数 F_{11} 中，RDDBO 算法的标准差小于 DBO 算法。与 DBO 算法相比，融合了三种改进方法的 MSFDBO 算法在 F_1～F_{12} 上的平均收敛精度和标准差都比 DBO 算法有所提升，说明三种改进策略的融合是有效的。

其次是 MSFDBO 与 MSADBO、MIDBO 算法的对比分析，在函数 F_1 中，三种算法都能搜索到理论最优解；在函数 F_2 中，只有 MIDBO 和 MSFDBO 算法能搜索到理论最优解；在函数 F_3、F_8 和 F_9 中，MSFDBO 算法的均值和标准差都小于 MSADBO 和 MIDBO 算法；在函数 F_4 中，MSADBO 算法的均值和标准差最小，而 MSFDBO 算法的均值小于 MIDBO 算法，但标准差略大于 MIDBO 算法；在函数 F_5、F_6 和 F_7 中，三种算法的均值和标准差都能达到相同的搜索精度；在函数 F_{10} 中，三种算法均能搜索到理论最优解，但 MSFDBO 算法的标准差最小；在函数 F_{11} 中，只有 MSADBO 和 MSFDBO 算法搜索到理论最优解，而 MSFDBO 算法的标准差最小；在函数 F_{12} 中，只有 MIDBO 和 MSFDBO 算法更接近理论最优解，而 MSFDBO 算法的标准差最小。

综上所述，MSFDBO 算法在不同函数上能够体现出明显的不同幅度搜索精度的提升，相较于 MSADBO 和 MIDBO 算法，MSFDBO 算法平衡了全局搜索和局部开发的能力，寻优精度和收敛速度得到了稳定提升。

4.2 不同群智能算法性能对比

将 DBO 算法、鲸鱼优化算法（whale optimization algorithm，WOA）[25]、灰狼优化算法（grey wolf optimizer，GWO）[26]、哈里斯鹰优化算法（Harris hawks optimization，HHO）[27]、天鹰优化算法（aquila optimizer，AO）[28] 和麻雀优化算法（sparrow search algorithm，SSA）[29] 与 MSFDBO 算法进行实验对比，各种算法的参数设定和实现参见文献[17]和文献[25-29]，对 12 个基准函数分别进行 30 次独立实验并取其均值，加粗数据表示最优值，种群数量设为 30，最大迭代次数 T_{max} 为 500，实验结果见表 3。

表 3 不同群智能算法的基准测试函数结果

Tab.3 Benchmarking the results of various swarm intelligence algorithms for a given function

函数	统计值	MSFDBO	DBO	SSA	WOA	GWO	HHO	AO
F_1	Mean	**0**	1.37×10^{-63}	2.39×10^{-73}	4.45×10^{4}	9.07×10^{-6}	2.12×10^{-70}	4.11×10^{-103}
	Std	**0**	7.51×10^{-63}	1.31×10^{-72}	1.34×10^{4}	1.65×10^{-5}	1.16×10^{-69}	2.25×10^{-102}
F_2	Mean	**0**	4.42×10^{-53}	4.09×10^{-31}	6.16×10^{1}	8.05×10^{-7}	6.62×10^{-49}	7.88×10^{-56}
	Std	**0**	1.61×10^{-52}	2.24×10^{-30}	2.76×10^{1}	6.02×10^{-7}	2.15×10^{-48}	4.31×10^{-55}
F_3	Mean	$\mathbf{9.32\times10^{-9}}$	7.29×10^{-4}	6.69×10^{-7}	3.84×10^{-1}	6.69×10^{-1}	2.40×10^{-4}	8.18×10^{-5}
	Std	$\mathbf{3.02\times10^{-8}}$	1.75×10^{-3}	9.62×10^{-7}	1.80×10^{-1}	2.89×10^{-1}	4.44×10^{-4}	1.60×10^{-4}
F_4	Mean	2.61×10^{-4}	1.27×10^{-3}	3.75×10^{-4}	4.36×10^{-3}	2.03×10^{-3}	1.54×10^{-4}	$\mathbf{1.08\times10^{-4}}$
	Std	2.95×10^{-4}	8.61×10^{-4}	2.99×10^{-4}	4.80×10^{-3}	1.04×10^{-3}	1.30×10^{-4}	$\mathbf{8.68\times10^{-5}}$
F_5	Mean	**0**	8.28×10^{-1}	**0**	**0**	2.05	**0**	**0**
	Std	**0**	3.36	**0**	**0**	2.57	**0**	**0**
F_6	Mean	$\mathbf{8.88\times10^{-16}}$	$\mathbf{8.88\times10^{-16}}$	$\mathbf{8.88\times10^{-16}}$	3.97×10^{-15}	1.03×10^{-13}	$\mathbf{8.88\times10^{-16}}$	8.63×10^{-14}
	Std	**0**	**0**	**0**	2.03×10^{-15}	1.73×10^{-14}	**0**	4.68×10^{-13}
F_7	Mean	**0**	**0**	**0**	**0**	6.04×10^{-3}	**0**	**0**
	Std	**0**	**0**	**0**	**0**	1.09×10^{-2}	**0**	**0**
F_8	Mean	$\mathbf{1.90\times10^{-8}}$	2.43×10^{-4}	1.10×10^{-7}	2.16×10^{-2}	5.33×10^{-2}	1.23×10^{-5}	3.11×10^{-6}
	Std	$\mathbf{1.01\times10^{-7}}$	1.23×10^{-3}	1.74×10^{-7}	9.03×10^{-3}	3.08×10^{-2}	2.53×10^{-5}	5.62×10^{-6}
F_9	Mean	$\mathbf{3.07\times10^{-4}}$	7.77×10^{-4}	3.18×10^{-4}	6.56×10^{-4}	3.80×10^{-3}	3.46×10^{-4}	5.16×10^{-4}
	Std	$\mathbf{1.13\times10^{-8}}$	4.77×10^{-4}	1.59×10^{-5}	4.51×10^{-4}	7.54×10^{-3}	3.64×10^{-5}	1.24×10^{-4}
F_{10}	Mean	**−1.03**	**−1.03**	**−1.03**	**−1.03**	**−1.03**	**−1.03**	**−1.03**
	Std	$\mathbf{5.45\times10^{-16}}$	6.32×10^{-16}	3.56×10^{-10}	3.75×10^{-9}	2.12×10^{-8}	3.99×10^{-9}	2.15×10^{-4}
F_{11}	Mean	**−3.86**	**−3.86**	**−3.86**	**−3.86**	**−3.86**	**−3.86**	**−3.86**
	Std	$\mathbf{2.58\times10^{-15}}$	2.99×10^{-3}	4.30×10^{-5}	1.05×10^{-2}	1.65×10^{-3}	5.07×10^{-3}	5.77×10^{-3}
F_{12}	Mean	$\mathbf{-1.02\times10^{1}}$	−6.61	$\mathbf{-1.02\times10^{1}}$	−7.70	−9.25	−5.21	-1.01×10^{1}
	Std	$\mathbf{5.83\times10^{-15}}$	2.36	1.10×10^{-5}	2.78	2.37	8.89×10^{-1}	2.41×10^{-2}

由表 3 分析可知，在函数 F_1 和 F_2 中，MSFDBO 算法的均值和标准差都为 0，达到了理论最优解，寻优能力远远领先于表中其他算法；在函数 F_3、F_8 和 F_9 中，虽然 7 种算法都没有寻找到

理论最优解，但 MSFDBO 算法搜索到的平均收敛精度和标准差是所有算法中最小的；在函数 F_4 中，MSFDBO 算法的平均收敛精度和标准差稍逊于 AO 和 HHO 算法；在函数 F_5 中，MSFDBO、SSA、WOA、HHO 和 AO 算法均能寻找到理论最优解；在函数 F_6 中，MSFDBO、DBO、SSA、HHO 算法的平均收敛精度相同，其标准差均为 0；在函数 F_7 中，除 GWO 外的其他算法均能搜索到理论最优解，标准差均为 0；在函数 F_{10} 和 F_{11} 中，7 种算法均能搜索到理论最优解，但 MSFDBO 算法的标准差搜索精度优于另外 6 种算法；在函数 F_{12} 中，MSFDBO 和 SSA 算法更接近理论最优解，但 MSFDBO 算法的标准差远小于 SSA 算法，其搜索稳定性更强。

综上所述，本文算法在多次寻优的过程中，其稳定性和鲁棒性明显优于其他算法，更接近理论值的最优解。这说明本文算法在搜索能力方面具有明显优势，能充分高效地探寻搜索空间，并保证较强的全局寻优能力和局部探索能力。

4.3　机器人本体验证

本文采用新松 T6A-19 型机器人进行实验验证，在机器人运动范围内随机选取均匀分布的 50 个测量点，采用 FARO Vantage E 激光跟踪仪测量机器人末端位置，实验环境如图 3 所示。新松 T6A-19 型六自由度工业机器人重复定位精度为 ±0.08 mm，额定负载为 6 kg，激光跟踪仪测量精度为 16 μm+0.8 μm/m。T6A-19 型工业机器人 D-H 参数见表 4。

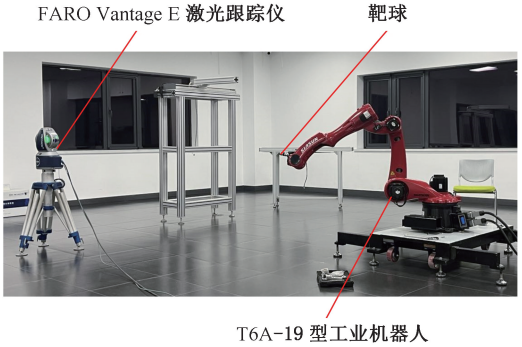

图 3　实验装置布局图

Fig.3　Layout of the experimental setup

表 4　T6A-19 机器人名义运动学参数

Tab.4　T6A-19 nominal kinematic parameters for robots

关节 i	a_i/mm	α_i/(°)	d_i/mm	θ_i/(°)
1	385	90	0	0
2	600	0	0	90
3	200	90	0	0
4	0	−90	925	0
5	0	90	0	0
6	0	0	92	90

$\boldsymbol{s}_i$ 的理论值如下：

$$\boldsymbol{s}_1 = [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0]^{\mathrm{T}}$$
$$\boldsymbol{s}_2 = [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0]^{\mathrm{T}}$$
$$\boldsymbol{s}_3 = [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0]^{\mathrm{T}}$$
$$\boldsymbol{s}_4 = [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0]^{\mathrm{T}}$$
$$\boldsymbol{s}_5 = [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0]^{\mathrm{T}}$$
$$\boldsymbol{s}_6 = [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0]^{\mathrm{T}}$$
$$\boldsymbol{s}_{\mathrm{tool}} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^{\mathrm{T}}$$

4.3.1　误差模型的适应度函数

采用本文算法优化计算工业机器人的几何参数误差时，其适应度函数为

$$f = \min\left(\sum_{j=1}^{N} \sqrt{(\delta p_{x,j})^2 + (\delta p_{y,j})^2 + (\delta p_{z,j})^2}\right) \tag{30}$$

式中：δp_x、δp_y、δp_z 为机器人实际位置相对于名义位置的误差；j 为测量位置点数。

4.3.2　实验结果分析

设置 MSFDBO 算法最大迭代次数 $T_{\max}$ 为 3000，种群数量设为 100，在 DBO、GWO、WOA 和 MSFDBO 等算法迭代求解的过程中，根据式(30)计算每一次迭代过程中的最优适应度函数值，通过 MSFDBO 算法迭代求解得到的函数值与使用 DBO、GWO 和 WOA 等算法求解的函数值进行对比，其最优适应度函数值变化如图 4 所示。由图 4 分析可知，在运动学参数辨识中，MSFDBO 算法相比其他算法收敛精度更高、收敛速度更快，整个过程中相比其他算法，具有更好的全局搜索能力和局部搜索能力。使用 MSFDBO 算法求解的 T6A-19 型机器人几何参数误差结果见表 5。

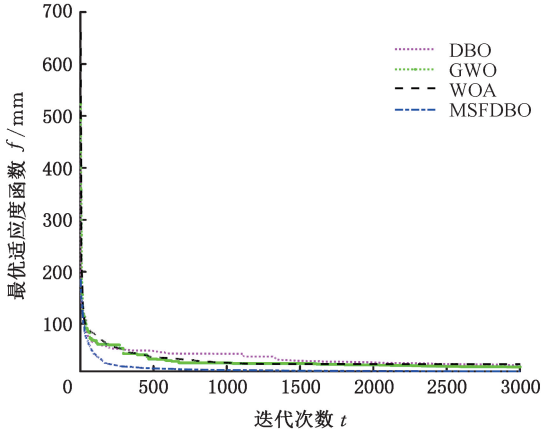

图 4　算法迭代曲线对比

Fig.4　Comparison of algorithmic iteration curves

为了验证本文算法辨识的运动学参数误差是否准确，在机器人的工作空间中随机选取 50 个点，将 DBO、GWO、WOA 和 MSFDBO 算法辨识出的运动学参数误差代入其中对比辨识结果。不同算法辨识前后的绝对位置误差如图 5 所示，对图 5 中辨识的点进行统计分析，统计结果见表 6。

表 5 T6A-19 机器人运动学参数辨识结果

Tab.5 T6A-19 robot kinematic parameter identification results

关节 i	Δw_x/rad	Δw_y/rad	Δw_z/rad	Δv_x/mm	Δv_y/mm	Δv_z/mm
1	-8.84×10^{-5}	6.47×10^{-4}	-1.72×10^{-4}	−0.58	0.58	0.49
2	-3.02×10^{-5}	7.94×10^{-4}	8.94×10^{-4}	−0.20	0.16	−0.19
3	2.70×10^{-4}	6.06×10^{-5}	4.90×10^{-5}	−1.14	0.22	−0.49
4	-1.31×10^{-3}	-2.92×10^{-4}	4.92×10^{-4}	0.30	1.00	−0.07
5	-5.88×10^{-3}	-7.78×10^{-4}	-1.62×10^{-3}	−1.49	2.28	−0.03
6	-5.59×10^{-3}	-1.74×10^{-3}	-4.06×10^{-4}	−0.18	1.80	0.05
tool	4.05×10^{-3}	-1.81×10^{-3}	1.27×10^{-2}	0.29	2.03	2.99

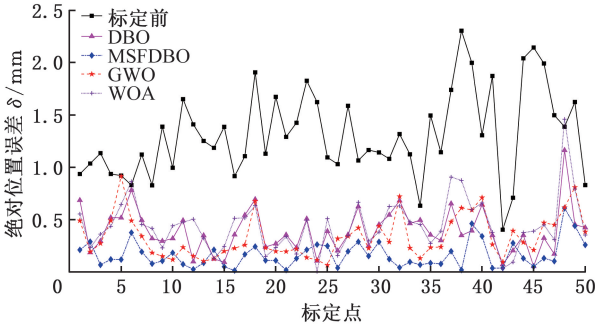

图 5 绝对位置误差辨识结果对比

Fig.5 Comparison of absolute position error identification results

表 6 绝对位置距离误差统计分析

Tab.6 Statistical analysis of absolute position distance error

参数标定算法	最大误差/mm	误差降低率/%	平均误差/mm	误差降低率/%	均方根误差/mm	误差降低率/%
标定前	2.3029		1.3117		1.3750	
DBO	1.1615	49.56	0.3919	70.12	0.4438	67.72
GWO	0.9140	60.31	0.3350	74.46	0.3909	71.57
WOA	1.4585	36.67	0.4448	66.09	0.5110	62.84
MSFDBO	0.6101	73.51	0.1653	87.40	0.2078	84.89

为了验证 MSFDBO 算法的准确性与泛化性，本文另外对 4 台 T6A-19 型工业机器人进行运动学参数辨识实验，绝对位置误差结果见表 7。

表 7 四台机器人的绝对位置误差

Tab.7 Absolute position errors of the four robots

T6A-19 型工业机器人	参数标定算法	最大误差/mm	误差降低率/%	平均误差/mm	误差降低率/%	均方根误差/mm	误差降低率/%
1	标定前	2.0093		1.0886		1.1721	
	DBO	0.6776	66.28	0.2990	72.53	0.3369	71.26
	GWO	0.7413	63.11	0.2987	72.56	0.3434	70.70
	WOA	0.7770	61.33	0.3455	68.26	0.3861	67.06
	MSFDBO	0.5263	73.81	0.1801	83.46	0.2194	81.28
2	标定前	1.9639		0.8213		0.8793	
	DBO	0.9672	50.75	0.3103	62.22	0.3589	59.18
	GWO	0.8735	55.52	0.4434	46.01	0.4739	46.10
	WOA	0.9382	52.23	0.4170	49.23	0.4919	44.06
	MSFDBO	0.3846	80.42	0.1597	80.56	0.1873	78.70
3	标定前	3.2486		1.7928		1.8650	
	DBO	0.6090	81.25	0.2617	85.40	0.3087	83.45
	GWO	0.5154	84.13	0.2394	86.65	0.2687	85.59
	WOA	0.6851	78.91	0.2297	87.19	0.2751	85.25
	MSFDBO	0.4737	85.42	0.1465	91.83	0.1743	90.65
4	标定前	3.2430		1.3552		1.4908	
	DBO	1.2540	61.33	0.4051	70.11	0.4632	68.93
	GWO	1.6450	49.28	0.5542	59.11	0.6521	56.26
	WOA	1.2292	62.10	0.5390	60.23	0.6066	59.31
	MSFDBO	0.6068	81.29	0.1892	86.04	0.2232	85.03

由表 6 可知，对机器人运动学参数进行补偿后，得到辨识前后绝对位置距离误差的绝对值的集合。根据统计学方法从最大误差、平均误差和均方根误差分析，MSFDBO 算法的辨识结果均优于其他算法，最大误差从辨识前的 2.3029 mm 减至 0.6101 mm，降低了 73.51%；平均误差从 1.3117 mm减至 0.1653 mm，降低了 87.40%；均方根误差从 1.3750 mm 减至0.2078 mm，降低了 84.89%。通过辨识前后数据的统计、分析和对比可知，MSFDBO 算法的辨识方法提高了工业机器人的绝对位置定位精度。

由表 7 中的绝对位置误差结果分析可知，4 台 T6A-19 型工业机器人经 MSFDBO 算法参数辨识后，绝对位置平均误差平均降低了 85.47%，均方根误差平均降低了 83.92%。

5　结语

本文根据高端制造业对工业机器人离线编程绝对定位精度的要求，提出了基于 MSFDBO 算法的工业机器人几何参数标定方法，通过建立机器人几何参数的 LPOE 模型，应用 MSFDBO 算法对 T6A-19 型工业机器人随机选定的测量点进行标定。实验结果表明，本文方法高效且准确地完成了工业机器人运动学参数辨识，4 台 T6A-19 型工业机器人的绝对位置平均误差平均降低了 85.47%，均方根误差平均降低了 83.92%，MSFDBO 算法的鲁棒性强、稳定性高，经标定的工业机器人末端绝对位置精度有大幅提高。

参考文献：

[1] CHEN Gang, LI Tong, CHU Ming, et al. Review on Kinematics Calibration Technology of Serial Robots[J]. International Journal of Precision Engineering and Manufacturing, 2014, 15(8):1759-1774.

[2] NGUYEN H N, ZHOU Jian, KANG H J. A New Full Pose Measurement Method for Robot Calibration[J]. Sensors, 2013, 13(7):9132-9147.

[3] JOUBAIR A, BONEV I A. Kinematic Calibration of a Six-axis Serial Robot Using Distance and Sphere Constraints[J]. The International Journal of Advanced Manufacturing Technology, 2015, 77(1):515-523.

[4] 谷乐丰，杨桂林，方灶军，等. 一种新型机器人自标定装置及其算法[J]. 机器人，2020，42(1):100-109.
GU Lefeng, YANG Guilin, FANG Zaojun, et al. The Calibration Algorithms for Industrial Robots Based on a Novel Self-calibration Device[J]. Robot, 2020, 42(1):100-109.

[5] CHEN I M, YANG Guilin, TAN C T, et al. Local POE Model for Robot Kinematic Calibration[J]. Mechanism and Machine Theory, 2001, 36(11/12):1215-1239.

[6] 张永贵，黄玉美，高峰. 基于遗传算法的机器人运动学参数误差识别[J]. 农业机械学报，2008，39(9):153-157.
ZHANG Yonggui, HUANG Yumei, GAO Feng. Robotic Kinematics Parameters Error Identification Based on Genetic Algorithm[J]. Transactions of the Chinese Society for Agricultural Machinery, 2008, 39(9):153-157.

[7] NUBIOLA A, BONEV I A. Absolute Calibration of an ABB IRB 1600 Robot Using a Laser Tracker[J]. Robotics and Computer-Integrated Manufacturing, 2013, 29(1):236-245.

[8] LIGHTCAP C, HAMNER S, SCHMITZ T, et al. Improved Positioning Accuracy of the PA10-6CE Robot with Geometric and Flexibility Calibration [J]. IEEE Transactions on Robotics, 2008, 24(2):452-456.

[9] 王君臣，王田苗，杨艳，等. 基于无迹卡尔曼滤波的机器人手眼标定[J]. 机器人，2011，33(5):621-627.
WANG Junchen, WANG Tianmiao, YANG Yan, et al. Robot Hand-eye Calibration Using Unscented Kalman Filtering[J]. Robot, 2011, 33(5):621-627.

[10] RENDERS J M, ROSSIGNOL E, BECQUET M, et al. Kinematic Calibration and Geometrical Parameter Identification for Robots[J]. IEEE Transactions on Robotics and Automation, 1991, 7(6):721-732.

[11] WANG Hongbo, GAO Tianqi, KINUGAWA J, et al. Finding Measurement Configurations for Accurate Robot Calibration: Validation with a Cable-driven Robot[J]. IEEE Transactions on Robotics, 2017, 33(5):1156-1169.

[12] 温秀兰，康传帅，宋爱国，等. 基于全位姿测量优化的机器人精度研究[J]. 仪器仪表学报，2019，40(7):81-89.
WEN Xiulan, KANG Chuanshuai, SONG Aiguo, et al. Study on Robot Accuracy Based on Full Pose Measurement and Optimization[J]. Chinese Journal of Scientific Instrument, 2019, 40(7):81-89.

[13] 姜一舟，于连栋，常雅琪，等. 基于改进差分进化算法的机器人运动学参数标定[J]. 光学精密工程，2021，29(7):1580.
JIANG Yizhou, YU Liandong, CHANG Yaqi, et al. Robot Calibration Based on Modified Differential Evolution Algorithm[J]. Optics and Precision Engineering, 2021, 29(7):1580-1588.

[14] 乔贵方，吕仲艳，张颖，等. 基于 BAS-PSO 算法的机器人定位精度提升[J]. 光学精密工程，2021，29(4):763-771.
QIAO Guifang, LÜ Zhongyan, ZHANG Ying, et al. Improvement of Robot Kinematic Accuracy Based on BAS-PSO Algorithm[J]. Optics and Precision Engineering, 2021, 29(4):763-771.

[15] 寇斌，郭士杰，任东城. 改进粒子群算法的工业机器人几何参数标定[J]. 哈尔滨工业大学学报，2022，54(1):9-13.
KOU Bin, GUO Shijie, REN Dongcheng. Geometric Parameter Calibration of Industrial Robot Based on Improved Particle Swarm Optimization [J]. Journal of Harbin Institute of Technology, 2022, 54(1):9-13.

[16] 乔贵方，杜宝安，张颖，等. 基于 POE 模型的工业机器人运动学参数二次辨识方法研究[J]. 农业机械学报，2024，55(1)：419-425.
QIAO Guifang，DU Baoan，ZHANG Ying，et al. Quadratic Identification Method of Kinematic Parameters of Industrial Robots Based on POE Model[J]. Transactions of the Chinese Society for Agricultural Machinery，2024，55(1)：419-425.

[17] XUE Jiankai，SHEN Bo. Dung Beetle Optimizer：a New Meta-heuristic Algorithm for Global Optimization[J]. The Journal of Supercomputing，2023，79(7)：7305-7336.

[18] 董奕含，喻志超，胡天跃，等. 基于改进蜣螂优化算法的瑞雷波频散曲线反演方法[J]. 油气地质与采收率，2023，30(4)：86-97.
DONG Yihan，YU Zhichao，HU Tianyue，et al. Inversion of Rayleigh Wave Dispersion Curve Based on Improved Dung Beetle Optimizer Algorithm[J]. Petroleum Geology and Recovery Efficiency，2023，30(4)：86-97.

[19] 李斌，高鹏，郭自强. 改进蜣螂算法优化 LSTM 的光伏阵列故障诊断[J]. 电力系统及其自动化学报，2024，36(8)：70-78.
LI Bin，GAO Peng，GUO Ziqiang. Improved Dung Beetle Optimizer to Optimize LSTM for Photovoltaic Array Fault Diagnosis[J]. Proceedings of the CSU-EPSA，2024，36(8)：70-78.

[20] 潘劲成，李少波，周鹏，等. 改进正弦算法引导的蜣螂优化算法[J]. 计算机工程与应用，2023，59(22)：92-110.
PAN Jincheng，LI Shaobo，ZHOU Peng，et al. Dung Beetle Optimization Algorithm Guided by Improved Sine Algorithm[J]. Computer Engineering and Applications，2023，59(22)：92-110.

[21] 郭琴，郑巧仙. 多策略改进的蜣螂优化算法及其应用[J]. 计算机科学与探索，2024，18(4)：930-946.
GUO Qin，ZHENG Qiaoxian. Multi-strategy Improved Dung Beetle Optimizer and Its Application[J]. Journal of Frontiers of Computer Science and Technology，2024，18(4)：930-946.

[22] ZHOU Yongquan，WANG Rui，LUO Qifang. Elite Opposition-based Flower Pollination Algorithm[J]. Neurocomputing，2016，188：294-310.

[23] DEHGHANI M，TROJOVSKY P. Osprey Optimization Algorithm：a New Bio-inspired Metaheuristic Algorithm for Solving Engineering Optimization Problems[J]. Frontiers in Mechanical Engineering，2023，8：1126450.

[24] SUGANTHAN P N，HANSEN N，LIANG J，et al. Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization[J]. KanGAL Report，2005005：2005.

[25] MIRJALILI S，LEWIS A. The Whale Optimization Algorithm[J]. Advances in Engineering Software，2016，95：51-67.

[26] MIRJALILI S，MIRJALILI S M，LEWIS A. Grey Wolf Optimizer[J]. Advances in Engineering Software，2014，69：46-61.

[27] HEIDARI A A，MIRJALILI S，FARIS H，et al. Harris Hawks Optimization：Algorithm and Applications[J]. Future Generation Computer Systems，2019，97：849-872.

[28] ABUALIGAH L，YOUSRI D，ABD ELAZIZ M，et al. Aquila Optimizer：a Novel Meta-heuristic Optimization Algorithm[J]. Computers & Industrial Engineering，2021，157：107250.

[29] XUE Jiankai，SHEN Bo. A Novel Swarm Intelligence Optimization Approach：Sparrow Search Algorithm[J]. Systems Science & Control Engineering，2020，8(1)：22-34.

（编辑 陈 勇）

作者简介：许佳璐，男，1999 年生，硕士研究生。研究方向为工业机器人运动控制。刘笑楠*（通信作者），女，1979 年生，副教授。研究方向为图像处理、计算机视觉。E-mail：liuxiaonan@sut.edu.cn。

本文引用格式：
许佳璐，刘笑楠，李朋超，等. 基于多策略融合蜣螂优化算法的工业机器人运动学参数辨识方法[J]. 中国机械工程，2025，36(2)：294-304.
XU Jialu，LIU Xiaonan，LI Pengchao，et al. Kinematics Parameter Identification for Industrial Robots Based on Multi-strategy Fusion DBO Algorithm[J]. China Mechanical Engineering，2025，36(2)：294-304.