文章编号:1000-2367(2023)05-0046-11 DOI:10.16366/j.cnki.1000-2367.2023.05.005

求解逆运动学的多策略蜻蜓算法

黄华娟[a],闵峰[b]

(广西民族大学 a.人工智能学院;b.电子信息学院,南宁 530006)

摘 要:蜻蜓算法(Dragonfly Algorithm,DA)是一种新型群智能算法,存在求解精度不高、收敛速度慢、优化不稳定等不足.基于此,提出了一种多策略改进的蜻蜓算法(Multi-strategy Improved Dragonfly Algorithm,MIDA).首先通过 Logistic 混沌映射改善算法的初始种群,使算法能更快锁定最优解区域;其次融合共生生物搜索算法(Symbiotic Organisms Search,SOS)来增加个体间的信息交流;最后采用了余弦扰动避免蜻蜓算法过早陷入局部最优解,并且采用正交实验验证该算法的可行性.通过 12 个基准函数和求解逆运动学这一实际应用来验证 MIDA 的有效性.结果表明,MIDA 在函数优化方面遥遥领先,在求解逆运动学问题中,优化效果比其他对比算法高 10%~35%.

关键词:蜻蜓算法;混沌映射;共生生物搜索算法;余弦扰动;函数优化;逆运动学

中图分类号:TP18 **文献标志码**:A

最优化是应用数学的一个分支,主要指在一定的条件限制下,选取某种研究方案使目标达到最优的一种方法.最优化问题在当今的军事、工程、管理等领域都有着极其广泛的应用.蜻蜓优化算法(Dragonfly Algorithm,DA)是 MIRJALILI[1] 于 2016 年提出的一种新型智能优化算法.其主要灵感源于自然界中蜻蜓的静态和动态群集行为,在解决优化问题时[2],它存在收敛精度低,容易陷入局部最优解,求解时间长与不稳定等缺点.

针对以上问题,国内外学者做了很多的优化与改进.高旭等[3]提出基于自适应权重的蜻蜓算法,通过自动调整原聚集、壁撞与结对等 3 种权重,改善算法的勘探能力,并应用于瑞雷波频散曲线反演中.薛海峰等[4]针对风电并网协调输电网扩展规划问题,提出基于离散多目标蜻蜓算法,并对风电协调输电网扩展规划模型进行求解.YUAN 等[5]提出了自适应阻力和耐力的蜻蜓算法,用 3 个著名的约束工程问题验证了其可行性.张颖等[6]对蜻蜓算法的迭代进化机制进行改进,将聚类中心等效为蜻蜓个体编码,利用蜻蜓算法的寻优优势,改善模糊 C 均值聚类性能.NAGA LAKSHMI 等[7]基于混合遗传蜻蜓算法对径向配电系统配电电源进行优化.CHANTAR 等[8]通过混合二进制蜻蜓算法,融合模拟退火机制用于特征选择.SALGOTRA 等[9]针对算法搜索性差,内聚和对齐不平衡等问题,提出了变异算子的概念,并提出了 7 个版本的 DA,遵循自适应参数、世代划分、改进的开发阶段和线性递减的种群适应来改进 DA 的搜索、收敛和其他特性.

上述的改进策略能在一定程度上提升蜻蜓算法的迭代寻优能力,但是大多数研究仅对算法的权重进行调整,并没有改善蜻蜓算法易陷入局部解与其优化能力不稳定等缺陷.为此,本文提出了一种多策略改进蜻蜓算法.首先针对算法初始种群的随机性,本文采用 Logistic 混沌映射初始化种群,使算法能够更快地锁定最优解区域.其次,采用共生生物搜索算法,改善蜻蜓算法的勘探与开发能力,增强寻优过程中的信息交流.最后,采用余弦扰动避免算法过早陷入局部最优解.从算法初始化,到改善算法的勘探和开发能力,再到最终的变异策略,全面优化了蜻蜓算法.为了验证 MIDA 的性能,本文采用 12 个基准函数与其他热门的优化算法进行比较,同时应用于逆运动学求解五自由度机械臂的参数中.实验结果表明,MIDA 相较于其他算法,在函数

收稿日期:2022-06-21;**修回日期**:2022-09-08.

基金项目:国家自然科学基金(62266007;61662005);广西自然科学基金(2021GXNSFAA220068;2018GXNSFAA294068).

作者简介(通信作者):黄华娟(1984—),女,广西崇左人,广西民族大学副教授,博士,研究方向为机器学习和数据挖掘,E-mail:hhj-025@163.com.

优化中有着巨大的优势,同时更加稳定,并且在逆运动学中的表现更加精确与稳定.

1　蜻蜓算法

1.1　基本理论

蜻蜓算法是一种新型的元启发式群智能优化算法,其原理是模拟大自然中蜻蜓寻找猎物的行为.该算法源于自然界中蜻蜓动态和静态的智能群行为,对蜻蜓的飞行路线、躲避天敌及寻找食物等生活习性进行数学建模.在动态群中,为获得更好的生存环境,大量的蜻蜓集群朝着共同的方向进行远距离迁徙;在静态群中,为寻找其他飞行猎物,由小部分蜻蜓组成的各个小组,在较小的范围内来回飞行.蜻蜓飞行过程中的局部运动与飞行路径的临时突变是静态群的主要特征.在自然界中,蜻蜓的生活习性可以归纳为 5 类行为方式:排队、结盟、分离、寻找猎物和躲避天敌.这 5 类行为的数学模型如式(1)所示:

$$\begin{cases} A_i = \dfrac{\sum_{j=1}^{N} V_j}{N}, \\ C_j = \dfrac{\sum_{j=1}^{N} X_j}{N} - X, \\ S_i = -\sum_{j=1}^{N} (X - X_j), \\ F_i = X^+ - X, \\ E_i = X^- + X, \end{cases} \tag{1}$$

其中,X 是当前蜻蜓所在位置,N 是邻近蜻蜓的个数,X_j 是第 j 只相邻蜻蜓所处位置,V_j 是第 j 只相邻蜻蜓的飞行速度,X^+ 是待捕食的食物所处的位置,X^- 是天敌所处位置.

A_i 代表第 i 只蜻蜓排队行为的位置向量;C_i 代表第 i 只蜻蜓结盟行为的位置向量;S_i 代表第 i 只蜻蜓分离行为的位置向量;F_i 代表第 i 只蜻蜓猎食行为的位置向量;E_i 代表第 i 只蜻蜓逃避天敌行为的位置向量.

蜻蜓个体飞行步长向量更新公式如下:

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + \omega \Delta X_t, \tag{2}$$

其中 s 代表分离权重;a 代表对齐权重;c 代表凝聚权重;f 代表猎物权重因子;e 代表天敌权重因子;ω 代表惯性权重;t 代表当前迭代次数. 蜻蜓个体的位置更新公式如下:

$$X_{t+1} = X_t + \Delta X_{t+1}, \tag{3}$$

其中 t 是当前迭代次数,X_t 是当前个体的位置向量,ΔX_{t+1} 是步长向量.

上述迭代公式为蜻蜓个体周围有邻近个体,当某只蜻蜓没有相邻个体时,通过使用 Lévy 飞行的方法绕搜索空间飞行,在这种情况下,蜻蜓位置根据以下公式进行更新:

$$X_{t+1} = X_t + \text{Lévy}(d)\Delta X_t, \tag{4}$$

其中 t 表示当前迭代次数,d 代表位置向量的维度.

Lévy 飞行的计算公式如下:

$$\text{Lévy}(d) = 0.01 \frac{r_1 * \delta}{|r_2|^{\frac{1}{\beta}}}, \tag{5}$$

其中 r_1, r_2 是两个[0,1]的随机数,β 是一个常数(本文为 1.5),δ 的计算方法如下:

$$\delta = \left(\frac{\Gamma(1+\beta)\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{\beta}\right)\beta 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}}, \tag{6}$$

其中 $\Gamma(x) = (x-1)!$

1.2　DA 的算法步骤

蜻蜓算法的步骤如下:

步骤 1 初始化种群参数,包括种群数量 n,最大迭代次数 $i\_\max$,位置向量 X,方向向量 ΔX.

步骤 2 计算各个个体的适应度值,确定该种群的食物位置与天敌位置.

步骤 3 更新权重,包括领域半径 r、对齐权重 a、分离权重 s、凝聚权重 c、天敌与猎物权重因子 e,f.

步骤 4 通过式(1)更新各项位置向量.

步骤 5 通过式(2)~(4)更新蜻蜓的位置.

步骤 6 判断新的位置解是否更优,选取更优的解位置.

步骤 7 重复步骤 2 到步骤 6,直到满足最大迭代次数.

2 多策略改进的蜻蜓算法

为了提高算法的收敛速度、最优解的质量以及稳定性,本文提出一种多策略改进的蜻蜓算法(MIDA),从初始解的选取,勘探与开发过程的改进和余弦扰动 3 个方面进行改进.

2.1 Logistic 混沌映射

混沌是非线性动力系统,描述的是任意随时间变化的过程,这个过程是类似随机的、非周期具有收敛性的,同时初始值对其影响极大.其数学表达式如下:

$$X_{n+1}=X_n\mu(1-X_n), \tag{7}$$

其中,Logistic 参数 $\mu\in[0.0,4.0]$,研究表明,当 $x\in[0.0,1.0]$时,Logistic 映射为混沌状态,即:在此映射作用下产生的序列是非周期的.

图 1 为 Logistic 参数 μ 在[2.6,4.0]区间内,X_0 为 0.5 时的 Logistic 映射图,从图 1 中可以看出 μ 值越接近 4.0,X 的取值越无限分布在[0.0,1.0]区间内.因此本文的 μ 值取为 3.999.

表 1 呈现的是 Logistic 的随机分布特性,取 $X_0=0.2$,$\mu=3.999$,迭代 10 000 次的结果.分布情况如表 1 所示.

表 1 映射值的分布范围

Tab. 1 Distribution range of mapped values

分布区间	[0.0,0.1)	[0.1,0.2)	[0.2,0.3)	[0.3,0.4)	[0.4,0.5)	[0.5,0.6)	[0.6,0.7)	[0.7,0.8)	[0.8,0.9)	[0.9,1.0)
个数	1 973	895	739	716	627	657	647	770	912	2 064
占比%	19.73	8.95	7.39	7.16	6.27	6.57	6.47	7.70	9.12	20.64

由图 2 与表 1 可知:Logistic 映射的迭代序列分布并不均匀,呈中间小两头大的情形,虽然分布不均,但足以满足需求.

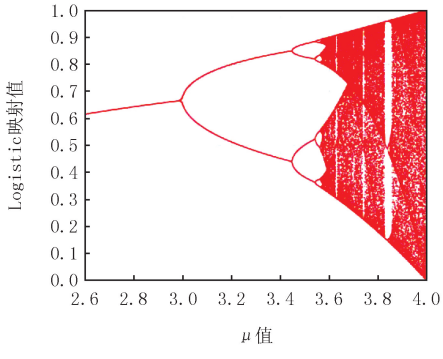

图1 不同 μ 值下的映射范围

Fig. 1 Mapping range under different values

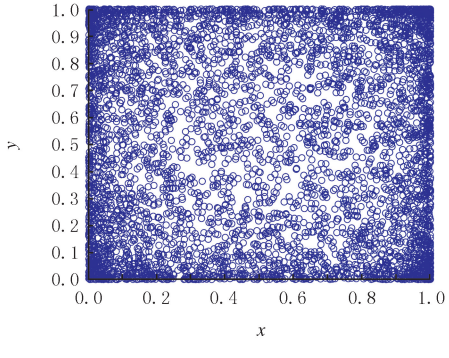

图2 二维映射下的范围

Fig. 2 Range under 2D mapping

Logistic 映射产生的数值在[0.0,1.0)间,本文通过式(8),将范围扩展到解空间中.

$$X_i=X_i\times(X_{\max}-X_{\min})+X_{\min}. \tag{8}$$

2.2 共生生物搜索算法

共生生物搜索算法是一种基于生物学中共生现象的启发式搜索算法,该算法参数少,操作简单、易于实

现且稳定性好.其中包括互利与共栖阶段.

在生态系统中,两个生物保持互利关系,其表达式如下:

$$\begin{cases} X_i = X_i + \text{rand}(0,1) \times (X_{\text{best}} - R_{MV} bf_1), \\ X_j = X_j + \text{rand}(0,1) \times (X_{\text{best}} - R_{MV} bf_2), \\ R_{MV} = (X_i + X_j)/2, \end{cases} \tag{9}$$

式中 R_{MV} 代表了 X_i 和 X_j 的交互关系;rand(0,1) 是[0,1]间的随机数;X_{best} 为最优个体;bf_1 和 bf_2 为利益因子,其值取为 1,表示部分受益.

在共栖阶段,生物 X_i 与 X_j 交互,使得 X_i 受益,X_j 不变.其表达式如下:

$$X_i = X_i + \text{rand}(-1,1)(X_{\text{best}} - X_j), \tag{10}$$

其中 $X_{\text{best}} - X_j$ 表示 X_j 对 X_i 提供的好处,即 X_j 帮助 X_i 提高对生态系统的适应度.

因此,MIDA 位置更新的表达式如下:

$$\begin{cases} X_i^{\text{new}} = X_i + \Delta X_{i+1} + \text{rand}(0,1) \times (X_{\text{best}} - R_{MV} bf_1), \dfrac{i}{\max\_i} \leq 0.5, \\ X_i^{\text{new}} = X_i + \Delta X_{i+1} + \text{rand}(-1,1) \times (X_{\text{best}} - X_i), \dfrac{i}{\max\_i} > 5. \end{cases} \tag{11}$$

在前半部分的迭代中采用互利关系,重点在于蜻蜓个体间的信息交互,有利于勘探过程,在迭代的后半段采用共栖方式,有利于局部解的开发.

2.3　余弦扰动

在蜻蜓算法中,每次的更新迭代采用的是贪心策略,即每次选取较优解作为新位置,但是到了迭代后期陷入局部解的情况下,不对其进行改进便会始终为同一解,使得该算法一直在做无用功.

为改善算法的寻优能力,采用余弦扰动的方式对未取得更优解的个体进行变异操作,采取该方式是因为余弦函数因子具有振荡特性,余弦扰动公式如下:

$$X_i = \cos((\pi \times t)/(\text{Max\_}i)) \times \Phi \times X_i, \tag{12}$$

其中 $\phi \in [-1,1]$ 的随机数,Max_i 是最大迭代次数,X_i 是第 i 代蜻蜓所处位置.

2.4　MIDA 的算法流程

根据以上策略,MIDA 的伪代码如下.

算法　多策略改进的蜻蜓算法

1.混沌初始化蜻蜓在搜索区域中的位置

2.初始化种群参数,包括种群数量 n,最大迭代次数 Max_i,方向向量 ΔX,通过 logistic 映射确定位置的向量 X

3.while 结束条件未满足

4.　计算各个个体的适应度值,确定该种群的食物位置与天敌位置

5.　更新各项权重,包括领域半径 r、对齐权重、分离权重、凝聚权重、天敌与猎物权重因子

6.　通过式(1)更新各项位置向量

7.　通过式(9)～(11)更新蜻蜓的位置

8.　if $X_{\text{new}} < X_i$

9.　　取优解为新的蜻蜓位置

10.　　else

11.　　通过式(12)更新蜻蜓位置

12.　　endif

13.　检查是否达到迭代次数

14.　endwhile

2.5　算法复杂度分析

空间复杂度指的是完成算法需要的空间,Logistic 初始化,互利共栖阶段与余弦扰动并未增加算法所需的空间,因此 DA 与 MIDA 的空间复杂度均取决于种群数量 N 与维度 d,为 $O(N \times d)$.

时间复杂度是指算法运行时间,取决于迭代次数 i,种群个数 n 等,DA 的时间复杂度为 $O(i(n_1 +$

$n_2(n_3+d_1+d_2)))$.

其中各参数含义依次为：i 为总迭代次数；n_1 为计算所有个体的适应度值的循环；n_2 为计算蜻蜓邻居所需的循环；n_3 为邻居个体数的存储循环，d_1 为越界处理；d_2 为存在邻居时，超出位移最大值的处理循环.其中 n_1,n_2,n_3 均为初始种群数，d_1,d_2 为个体的维度，其值为极小的常数.

因为 d_1,d_2 为极小的常数，且 n_1,n_2 的值为种群数量 N. 故 DA 的时间复杂度可简化为 $O(i(N+Nn_3))=O(iNn_3)$.

在 MIDA 中，Logistic 初始化与随机初始化本质相同，未增加算法的复杂度，互利与共栖应用于原算法公式，余弦扰动与位置更新处于同一级，均未增加算法的复杂度，因此 MIDA 的时间复杂度也是 $O(iNn_3)$.

2.6 正交实验检测

为证明上述改进对基本的蜻蜓算法均有所改善，本小节用基本蜻蜓算法(DA)，融合共生生物搜索的蜻蜓算法(SOSDA)，基于余弦扰动的蜻蜓算法(COSDA)与多策略改进的蜻蜓算法(MIDA)进行比较.高维单峰(f_1)、高维多峰(f_2)以及定维多峰测试函数(f_3)各取 1 个用来测试其性能.

其中 DA 如 2.2 小节所示；MIDA 如 2.4 小节所示；SOSDA 即基本蜻蜓算法中式(3)替换为式(11)，COSDA 即基本蜻蜓算法最后加入式(12).

4 个对比算法对表 2 的 3 种函数进行优化，每次迭代 400 代，并独立运行 30 次，优化得出的最小值、最大值、平均值与方差如表 3 所示，本文算法的结果见黑体部分.

表 2 测试函数详细信息

Tab. 2 Function details

Benchmark function	Dim	Range	min
$f_1(x)=\max\{\lvert x_i\rvert,1\leq i\leq n\}$	30	[−100,100]	0.000 0
$f_2(x)=\sum_{i=1}^{N}(\sum_{i=1}^{j}\lvert x_j\sin(x_i)+0.1x_j\rvert$	30	[−10,10]	0.000 3
$f_3(x)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1+6)^2+10(1-\frac{1}{8\pi}\cos x_1+10)$	4	[−5,5]	−10.536 4

表 3 实验结果

Tab. 3 Experimental results

Function	Index	DA	SOSDA	COSDA	MIDA
f_1	Best	7.156E+01	3.630E-04	1.695E-05	**8.618E-40**
	Worst	9.337E+01	2.526E+00	2.083E-01	**5.021E-36**
	Mean	8.434E+01	4.840E-01	3.139E-02	**7.515E-37**
	Std	5.188E+00	5.539E-01	4.661E-02	**1.320E-36**
f_2	Best	4.403E-03	6.915E-04	5.331E-04	**3.075E-04**
	Worst	9.296E-02	3.408E-02	3.698E-02	**1.127E-03**
	Mean	2.950E-02	8.000E-03	9.523E-03	**8.456E-04**
	Std	2.300E-02	9.413E-03	8.548E-03	**3.213E-04**
f_3	Best	−2.089E+00	−4.986E+00	−6.378E+00	**−1.054E+01**
	Worst	−6.276E-01	−1.549E+00	−9.415E-01	**−1.054E+01**
	Mean	−1.008E+00	−3.387E+00	−2.082E+00	**−1.054E+01**
	Std	3.612E-01	1.032E+00	1.298E+00	**8.317E-07**

由表 3 可以看出，无论是 SOSDA 还是 COSDA，他们对函数的优化效果都比 DA 好，而 MIDA 更是远优于其他 3 种算法，在对 f_1 的优化中，效果比其他算法高出 30 个数量级.因此，本文所提出的改进方法的可行性与有效性得到了证明.

3 仿真实验和结果分析

本文通过 12 个基准测试函数来测试改进算法，同时将其应用在逆运动学中求解五自由度机械臂的参

数.为避免实验结果的偶然性，本文对算法独立运行 400 次，种群个数均为 30，同时统计并分析它们的最佳值、最差值、平均值和标准差.

本实验运行环境为 Windows10，处理器：Intel(R)Core(TM)i5-5200U @2.20 GHz；

内存：4.00 GB.所有算法代码都在 MATLAB R2018b 中运行.

3.1　基准函数测试

本文采用了 12 个基准函数，其中高维单峰（f_1～f_4）、高维多峰（f_5～f_8）以及定维多峰（f_9～f_{12}）测试函数各取 4 个来测试其性能如何.同时与灰狼算法[10]（Grey Wolf Optimizer，GWO），花授粉算法[11]（Flower Pollination Algorithm，FPA），飞蛾扑火算法[12]（Moth-flame Optimization Algorithm，MFO）进行对比与分析.

算法的参数为 DA：$r \in [0,1]$，$\beta=1.5$，MIDA：$\mu=3.999$，$\beta=1.5$，$r \in [0,1]$，$bf_1=1$，$\phi \in [-1,1]$，FPA：$p=0.1$，GWO：$r_1, r_2 \in [0,1]$，$A=2ar_1-a$，$C=2r_2$，MFO：$b=1$，$t \in [-1,1]$. 基准函数见附表Ⅰ.

3.2　实验结果分析

用 DA，FPA，GWO，MFO，MIDA 这 5 个算法对 12 个基准函数进行仿真，迭代次数均为 400 代，独立运行 30 次得出的最优值（Best）、最差值（Worst）、平均值（Mean）和方差（Std）如附表Ⅱ所示，最优结果见黑体部分.

每个函数优化出的最优值已经加粗，为了更方便明显地比较各算法的收敛速度，下面给出算法收敛曲线图.为比较各个算法的稳定性，同时给出他们的箱线图.由于篇幅限制，高维单峰、高维多峰与定维多峰测试函数各一张，如图 3～8 为 f_1，f_6，f_{12} 的收敛曲线及与之对应的箱线图.

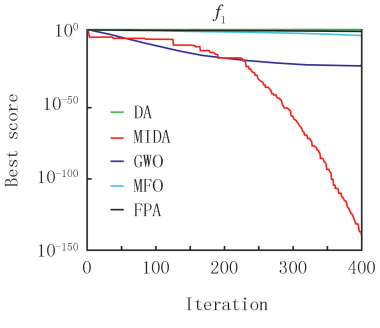

图3　f_1 的收敛曲线

Fig. 3　The convergence curve of f_1

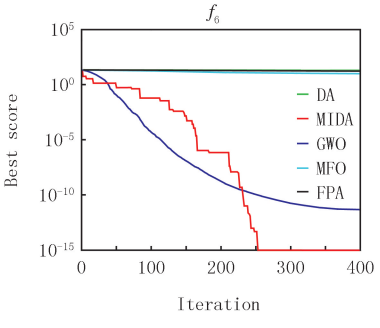

图4　f_6 的收敛曲线

Fig. 4　The convergence curve of f_6

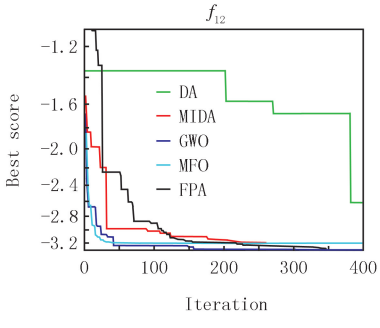

图5　f_{12} 的收敛曲线

Fig. 5　The convergence curve of f_{12}

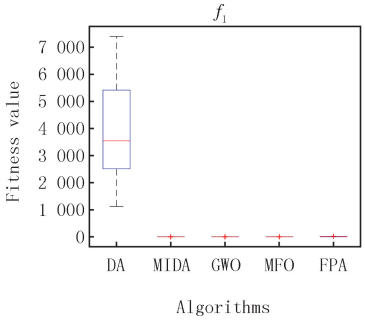

图6　f_1 的箱线图

Fig. 6　Box plot of f_1

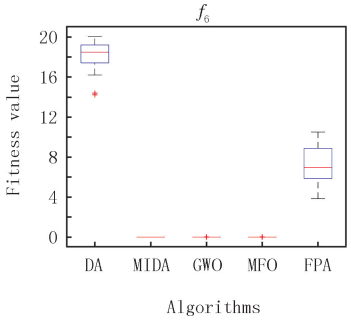

图7　f_6 的箱线图

Fig. 7　Box plot of f_6

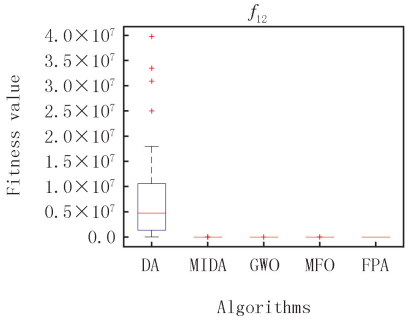

图8　f_{12} 的箱线图

Fig. 8　Box plot of f_{12}

由附表Ⅱ的实验结果可知，MIDA 相较于其他 4 种算法，对高维单峰的函数优化极为显著，不仅效果最好，且优化精度高出其他算法近百个数量级.对于高维多峰函数的优化也稳居第一.在对定维多峰的函数优化中，收敛最优值也稳定第一，但是在稳定性方面不如 GWO.

由图 3 和图 6 可以看出，相较于其他 4 种算法，MIDA 对函数的优化有压倒性的优势，其他 4 种算法的收敛已经趋于稳定，由此可见其他 4 种算法具有一定的局限.而 MIDA 的下降趋势在 400 代仍旧明显.由附

表Ⅱ的实验结果也能看出它强大的收敛能力,不仅仅是对 f_1,MIDA 对所有的高维单峰函数的收敛精度远高于其他算法,同时标准差更小,表明 MIDA 的稳定性极高.

由图 4 和图 7 可知,MIDA 的收敛精度更高,且收敛速度更快,同时由附表Ⅱ的标准差栏目中可以看出 MIDA 所得出的值相当低,由此可知 MIDA 的稳定性极高.在对高维多峰函数优化时,MIDA 得出的最优值都是最好的,由此可见 MIDA 的寻优性能极佳.

由图 5 可以得知,MIDA 对个别函数的收敛能力与其他 4 种算法接近,但是从附表Ⅱ可以看出 MIDA 的精准度更高,优化得出的最优值更好.MIDA 对 f_{12}的收敛速度略低于 GWO,但是 MIDA 最终的最优值与连续运行 30 次所得的方差更优于 GWO,可见 MIDA 在函数的优化中处于遥遥领先地位.附表Ⅱ中的数值也能显示 MIDA 在对定维多峰函数优化时总能保持最佳的收敛性能.图 6～8 的箱线图表明 MIDA 的稳定性明显优于其他 4 种算法.

3.3 Wilcoxon 秩和检验

在评估元启发式算法的性能时,只比较均值与最优值是不够的[13].因此,为证明所提出算法的结果与其他算法是否在统计上有显著不同.需要进行统计测试.为了减少犯第一类错误的概率,本文先通过 Kruskal-Wallis 检验进行多组比较,判断 5 种算法是否有显著性差异.

实验结果得出检验值 $p=9.02\mathrm{E}^{-29}<0.05$,说明在显著水平 0.05 下,检验拒绝了原假设,即 5 种算法具有显著性差异.

在具有显著性差异的基础上,通过显著水平为 0.05 的 Wilcoxon 秩和检验进行多重比较.Wilcoxon 秩和检验的目的是用于比较两个独立样本的差异.在算法中用于比较两种算法有没有差异,有差异才可以比较优劣.在 Wilcoxon 秩和检验中,用 MIDA 和其他算法进行对比,实验结果得出的 p 值小于 0.05,则证明两种算法差异有统计学意义.表 4 为 MIDA 与其他算法采用最优值进行秩和检验的结果.“+”“－”“＝”分别为本文算法与比较算法的成绩相比是好/差/相等.秩和检验显示,当显著性水平为 0.05 时,MIDA 与其他 4 种算法具有显著性差异.

表 4 Wilcocon 秩和检验实验结果

Tab. 4 Experimental results of Wilcocon rank sum test

基准测试函数	MIDA-DA	MIDA-FPA	MIDA-GWO	MIDA-MFO
f_1	3.019 9E-11	3.019 9E-11	3.019 9E-11	3.019 9E-11
f_2	6.680 9E-11	6.680 9E-11	6.680 9E-11	6.680 9E-11
f_3	4.461 8E-11	4.461 8E-11	3.019 9E-11	4.461 8E-11
f_4	4.461 8E-11	4.461 8E-11	4.461 8E-11	4.461 8E-11
f_5	3.019 8E-11	3.019 8E-11	0.002 8	9.918 6E-11
f_6	1.211 7E-12	1.211 7E-12	1.685 2E-14	1.289 0E-12
f_7	1.211 7E-12	1.211 7E-12	1.945 7E-09	1.211 7E-12
f_8	4.080 5E-12	4.080 5E-12	4.080 5E-12	0.001 1
f_9	1.211 7E-12	1.211 7E-12	0.081 5	4.246 6E-12
f_{10}	1.211 7E-12	1.211 7E-12	5.852 1E-09	1.211 7E-12
f_{11}	3.008 5E-11	3.008 5E-11	3.0085 2E-11	1.247 4E-10
f_{12}	2.870 0E-11	2.870 0E-11	2.870 0E-11	4.311 8E-11
+/－/＝	12/0/0	12/0/0	11/1/0	12/0/0

3.4 求解逆运动学中五自由度机械臂的参数问题

从上一小节 MIDA 对函数的优化中可以得知,MIDA 拥有较强的寻优性能,接下来利用 MIDA 解决五自由度机械臂逆运动学的求解.在机械臂的运动控制中,可以分为正运动学与逆运动学,正运动学是给定机器人关节变量的取值来确定末端执行器的位置和姿态.逆运动学是根据给定的末端执行器的位置和姿态来确定机器人关节变量的数值.

DH 建模方法[14]是由 DENAVIT 和 HARTENBERG 提出的一种建模方法,主要用在机器人运动学上.此方法仅仅通过 4 个参数便可以确定机械臂的各个节点的位姿状态.对每个关节建立坐标系,关节总是绕着 z 轴旋转,同时,4 个关节变量可以确定机械臂的位置.其中 θ 表示绕 z 轴的旋转角度,d 表示在 z 轴上两条相邻公垂线的距离,a 表示每一条公垂线的长度,也叫做关节变量,a 表示两个相邻 z 轴的角度,又称为关节扭转.图 9 标注了各个关节变量的信息.

本文使用标准型 D-H 参数(STD)建模[15],其建模步骤如下:

(1)绕 z_{i-1} 轴旋转 θ_i,使 x_{i-1} 与 x_i 平行;

(2)沿 z_{i-1} 轴平移 d_i,使 x_{i-1} 与 x_i 重合;

(3)沿 z_i 轴平移 a_i,使 z_{i-1} 与 z_i 重合;

(4)绕 x_i 轴旋转 α_i,使 z_{i-1} 与 z_i 共线;

通过每个旋转和平移的步骤得到的旋转矩阵:

$$\mathrm{Rot}(z_{i-1},\theta_i)=\begin{pmatrix}\cos\theta_i & -\sin\theta_i & 0 & 0\\ \sin\theta_i & \cos\theta_i & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix},\mathrm{Trans}(z_{i-1},d_i)=\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & d_i\\ 0 & 0 & 0 & 1\end{pmatrix},\tag{13}$$

$$\mathrm{Trans}(x_i,a_i)=\begin{pmatrix}1 & 0 & 0 & a_i\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix},\mathrm{Rot}(x_i,\alpha_i)=\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & \cos\alpha_i & -\sin\theta_i & 0\\ 0 & \sin\alpha_i & \cos\alpha_i & 0\\ 0 & 0 & 0 & 1\end{pmatrix},\tag{14}$$

由此可以得到变换矩阵:

$$T_i^{i-1}=\mathrm{Rot}(z_{i-1},\theta_i)\times\mathrm{Trans}(z_{i-1},d_i)\times\mathrm{Trans}(x_i,a_i)\times\mathrm{Rot}(x_i,\alpha_i)=\begin{pmatrix}\cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i\\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i\\ 0 & \sin\alpha_i & \cos\alpha_i & d_i\\ 0 & 0 & 0 & 1\end{pmatrix}.\tag{15}$$

以图 10 的五自由度机械臂为例,选定空间中的一点(x,y,z),通过 MIDA,优化得出最佳的 4 个关节变量.首先通过模型图建立每个节点的坐标系,然后得出 D-H 参数表如表 5 所示.

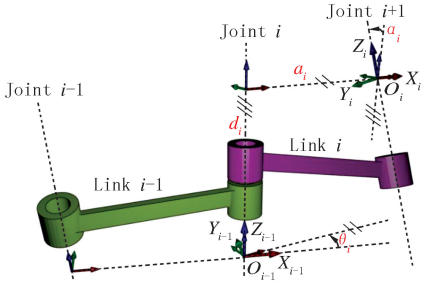

图9　4个关节变量概要图

Fig. 9　Schematic diagram of the four joint variables

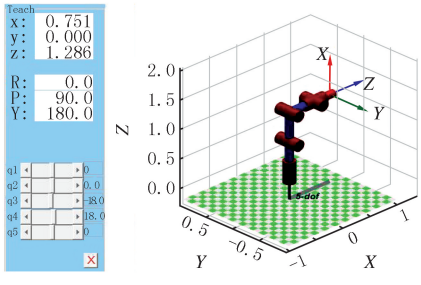

图10　五自由度机械臂模型图

Fig. 10　Model diagram of a five-degree-of-freedom robotic arm

D-H 参数表的填写方式如下:

(1)a_i:沿 X_i 轴方向,从 Z_i 移动到 Z_{i+1} 的距离.

(2)α_i:从 Z_i 旋转到 Z_{i+1} 的角度,转轴为 Z_i,遵循右手定则,以 X_i 方向为大拇指方向.即杆 i 与杆 $i+1$ 的夹角.

(3)d_i:沿 Z_i 轴从 X_{i+1} 移动到 X_i 的距离.

(4)θ_i:绕 Z_i 轴从 X_{i+1} 转到 X_i 的角度.

表 5 五自由度机械臂 D-H 参数

Tab. 5 D-H parameters of the five-degree-of-freedom manipulator

i	θ_i	a_i	α_i	d_i	i	θ_i	a_i	α_i	d_i
1	θ_1	0	$-p_i/2$	$L_1=0.5$	4	θ_4	$L_4=0.2$	0	0
2	θ_2	0	0	0	5	θ_5	$L_5=0.2$	0	0
3	θ_3	$L_3=0.3$	0	0					

通过 D-H 参数表与变换矩阵 $\boldsymbol{T}_i^{i-1}$,可以求出每个节点的传递矩阵 $\boldsymbol{T}_0^1$、$\boldsymbol{T}_1^2$、$\boldsymbol{T}_2^3$、$\boldsymbol{T}_3^4$、$\boldsymbol{T}_4^5$,则末端节点的位姿相对于基坐标系的变换矩阵 $\boldsymbol{T}_0^5$ 如下所示:

$$\boldsymbol{T}_0^5=\boldsymbol{T}_0^1\boldsymbol{T}_1^2\boldsymbol{T}_2^3\boldsymbol{T}_3^4\boldsymbol{T}_4^5=\begin{pmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{16}$$

其中 (P_x,P_y,P_z) 表示机械臂末端位姿相对基座坐标系的位置.

通过 D-H 参数表求得 $\boldsymbol{T}_0^5$ 与通式对应,则

$$\begin{cases} P_x=\cos(\theta_1)(L_5\cos(\theta_2+\theta_3+\theta_4+\theta_5)+L_5\cos(\theta_2+\theta_3)+L_4\cos(\theta_2+\theta_3+\theta_4)), \\ P_y=\sin(\theta_1)(L_5\cos(\theta_2+\theta_3+\theta_4+\theta_5)+L_3\cos(\theta_2+\theta_3)+L_4\cos(\theta_2+\theta_3+\theta_4)), \\ P_z=L_1-L_5\sin(\theta_2+\theta_3+\theta_4+\theta_5)-L_3\sin(\theta_2+\theta_3)-L_4\sin(\theta_2+\theta_3+\theta_4). \end{cases} \tag{17}$$

在上述公式中,通过各个 θ 的值可以确定机械臂末端的位姿,这便是正运动学.而已知末端点的坐标(x,y,z)来求得各个角度是相当复杂的,这便是逆运动学[16],本章使用 MIDA 来求解逆运动学这一复杂问题.

适应度函数是启发式优化算法的重要内容之一,通过适应度函数值,算法才能不断地更新最佳值.在本研究中,使用欧氏距离作为适应度函数[17].利用 MIDA 获得了最接近预定目标点的位置.其适应度函数 f 如下:

$$f=\sqrt{(x-P_x)^2+(y-P_y)^2+(z-P_z)^2}, \tag{18}$$

其中,(x,y,z) 与(P_x,P_y,P_z) 分别为目标点的空间位置与机械臂末端位姿的空间位置.

3.5 实验结果

本文使用 MIDA 与 DA,GWO, FPA,MFO 同时对上述问题进行求解.设定目标位置为(0.5,0.5,0.5),独立运行 30 次,所得的最优值、最差值、平均值和标准差如表 6 黑体部分.

表 6 实验结果

Tab. 6 Experimental results

Algorithm	DA	FPA	GWO	MFO	MIDA
BEST	0.008 264 652	0.012 531 46	0.010 995 456	0.007 906 788	**0.007 106 781**
WORST	0.072 961 202	0.065 628 714	0.139 687 07	0.101 431 085	**0.065 280 687**
AVE	0.0404 833 95	0.035 141 242	0.062 052 231	0.038 704 294	**0.027 521 551**
STD	0.020 829 173	**0.012 771 128**	0.028 245 656	0.045 061 065	0.017 208 819

实验结果显示,MIDA 对五自由度机械臂逆运动学的优化稳定性略低于 FPA,但是在优化效果方面稳居第一,MIDA 优化所得的最优结果比 MFO 高出 10.12%,比 GWO 高出 35%,更是比 FPA 与 GWO 高出一个数量级.FPA 的稳定性虽然与 MIDA 接近,但是最优值的优化效果远不及 MIDA.图 11、12 为 MIDA 针对两个目标点的优化收敛曲线图,可以得知改进算法 MIDA 相较其他 4 个对比算法有显著优势.本文的实验结果表明 MIDA 在函数方面的优化效果极佳,同时在逆运动学方面的求解优势极为显著.

4 结 论

本文通过对初始种群的选取,使算法更快定位到最优解区域;对迭代寻优过程进行优化,使算法在勘探和开发过程更加全面;对结果扰动变异,使算法跳脱局部解,在这 3 个方面对蜻蜓算法进行了优化,并通过

12 个基准函数的测试与求解逆运动学这一实际应用证明了 MIDA 具有更强的收敛能力、更快的收敛速度和极佳的稳定性.在优化效果方面精度高出 10.12%.在实际应用中，需要机械臂准确高效获得抓取目标物品所需的关节变量.但是在本文中只应用了五自由度的机械臂，在空间中的搜索能力有限.未来将会对经典的七自由度机械臂[18]，甚至更多自由度的机械臂进行研究.

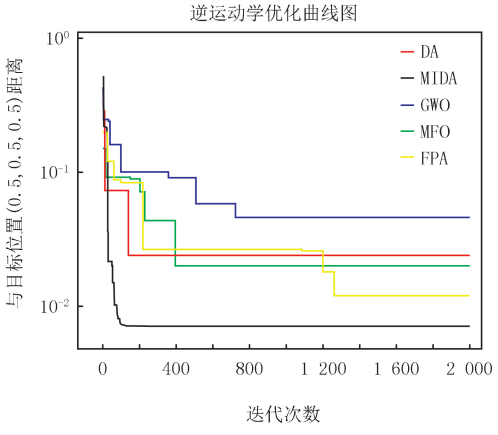

图11　目标点为(0.5, 0.5, 0.5)的收敛曲线图

Fig. 11　Convergence curve with target points of(0.5, 0.5, 0.5)

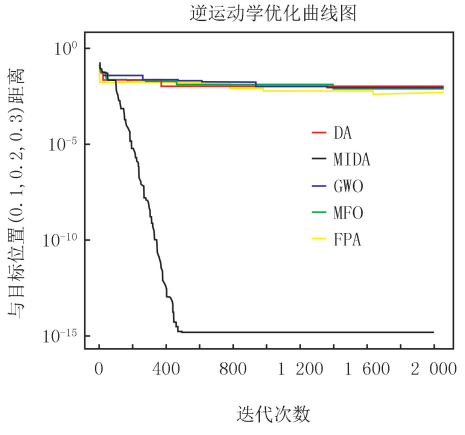

图12　目标点为(0.1, 0.2, 0.3)的收敛曲线图

Fig. 12　Convergence curve with target points of(0.1, 0.2, 0.3)

附　录

附表Ⅰ、Ⅱ见电子版(DOI:10.16366/j.cnki.1000-2367.2023.05.005).

参　考　文　献

[1] MIRJALILI S.Dragonfly algorithm:a new meta-heuristic optimization technique for solving single-objective,discrete,and multi-objective problems[J].Neural Computing and Applications,2016,27(4):1053-1073.

[2] 陈勇，吴彩娥，熊智新.基于衰减消去蜻蜓算法的小麦粉蛋白质近红外特征波长优选[J].食品科学，2022，43(14)：219-225.
CHEN Y，WU C E，XIONG Z X.Selection of near infrared wavelengths using attenuation elimination-binary dragonfly algorithm for wheat flour protein content prediction[J].Food Science，2022，43(14)：219-225.

[3] 高旭，于静，李学良，等.自适应权重蜻蜓算法及其在瑞雷波频散曲线反演中的应用[J].石油地球物理勘探，2021，56(4)：745-757.
GAO X，YU J，LI X L，et al. Rayleigh wave dispersion curve inversion based on adaptive weight dragonfly algorithm[J].Oil Geophysical Prospecting，2021，56(4)：745-757.

[4] 薛海峰，武晓云.基于离散多目标蜻蜓算法和改进 FCM 的风电协调输电网扩展规划研究[J].智慧电力，2021，49(6)：83-90.
XUE H F，WU X Y.Extended planning of wind power coordinated transmission network based on discrete multi-objective dragonfly algorithm and improved FCM[J].Smart Power，2021，49(6)：83-90.

[5] YUAN Y L，WANG S，LYU L Y，et al.An adaptive resistance and stamina strategy-based dragonfly algorithm for solving engineering optimization problems[J].Engineering Computations，2021，38(5)：2228-2251.

[6] 张颖，彭然.基于改进蜻蜓优化多核模糊聚类算法的异常检测[J].数学的实践与认识，2021，51(19)：208-219.
ZHANG Y，PENG R.Anomaly detection based on improved dragonfly algorithm and multi-core fuzzy clustering algorithm[J].Mathematics in Practice and Theory，2021，51(19)：208-219.

[7] NAGA LAKSHMI G V，JAYALAXMI A，VEERAMSETTY V.Optimal placement of distribution generation in radial distribution system using hybrid genetic dragonfly algorithm[J].Technology and Economics of Smart Grids and Sustainable Energy，2021，6(1)：9.

[8] CHANTAR H，TUBISHAT M，ESSGAER M，et al.Hybrid binary dragonfly algorithm with simulated annealing for feature selection[J].SN Computer Science，2021，2(4)：295.

[9] SALGOTRA R，SINGH U，SINGH S，et al.A new set of mutation operators for dragonfly algorithm[J].Arabian Journal for Science and Engineering，2021，46(9)：8761-8802.

[10] 黄云云，吴健，王斌，等.基于改进灰狼算法的混合发电系统优化设计[J].福州大学学报(自然科学版)，2021，49(6)：775-781.
HUANG Y Y，WU J，WANG B，et al.Sizing optimization of hybrid generation system based on improved grey wolf optimization[J].Journal of Fuzhou University(Natural Science Edition)，2021，49(6)：775-781.

[11] XIAO H H，WAN C X.An improved flower pollination algorithm based on multiple strategies[J].Journal of Software，2021，32(10)：3151-3175.

[12] ZHANG B D,ZHANG Y N,GUO L M,et al.Moth-to-fire optimization algorithm based on crossover operator and non-uniform mutation operator[J].Computer & Digital Engineering,2020,48(11):2622-2627.

[13] LIN T,CHEN T,LIU J Y,et al.Extending the Mann-Whitney-Wilcoxon rank sum test to survey data for comparing mean ranks[J].Statistics in Medicine,2021,40(7):1705-1717.

[14] 陶春,孙乾政.五自由度机械臂运动学系统建模与仿真[J].内燃机与配件,2020(1):261-262.
TAO C,SUN Q Z.Modeling and simulation of kinematics system of five-degree-of-freedom manipulator[J].Internal Combustion Engine & Parts,2020(1):261-262.

[15] 邓晓燕,林灿光,施翔宇,等.五自由度机械臂三维建模与仿真实验平台的构建[J].实验技术与管理,2018,35(3):118-122.
DENG X Y,LIN C G,SHI X Y,et al.3D modeling of 5-DOF mechanical arm and construction of simulation experimental platform[J].Experimental Technology and Management,2018,35(3):118-122.

[16] 李万莉,李宁.一种五自由度机械臂的运动学分析与逆运动学求解[J].机电一体化,2017,23(11):3-7.
LI W L,LI N.Kinematics analysis and inverse kinematics solution of a 5-DOF manipulator[J].Mechatronics,2017,23(11):3-7.

[17] 刘国平,杨先永,钟飞飞,等.六自由度机械臂运动学旋量逆解及简化算法[J].组合机床与自动化加工技术,2021(9):11-15.
LIU G P,YANG X Y,ZHONG F F,et al.Kinematics screw inverse solution and simplified algorithm of 6-DOF manipulator[J].Modular Machine Tool & Automatic Manufacturing Technique,2021(9):11-15.

[18] CHANF J,WANG Y Z,LI B.Fine manipulation method of 7-DOF robotic arm based on force/position hybrid algorithm[J].Robot,2016,38(5):531-539.

Multi-strategy dragonfly algorithm for solving inverse kinematics

Huang Huajuan[a], Min Feng[b]

(a. College of Artificial Intelligence; b. College of Electronic Information, Guangxi Minzu University, Nanning 530006, China)

Abstract: Dragonfly algorithm is a new type of swarm intelligence algorithm, which has shortcomings such as low solution accuracy, slow convergence speed, and unstable optimization. In this regard, a multi-strategy improved Dragonfly algorithm is proposed in this paper. Firstly, the initial population of the algorithm is improved through logistic chaotic mapping, so that the algorithm can lock the optimal solution area faster. secondly, the symbiotic search algorithm is integrated to increase the information exchange between individuals. Finally, the cosine disturbance is used to avoid the premature trapping of the dragonfly algorithm local optimal solution. And the feasibility of the algorithm is verified by orthogonal experiments. This paper verifies the effectiveness of MIDA with 12 benchmark functions and the practical application of solving inverse kinematics. The results show that MIDA is far ahead in function optimization, and its optimization effect is 10%-35% higher than other comparison algorithms in solving inverse kinematics problems.

Keywords: dragonfly algorithm; chaotic mapping; symbiotic organisms search; cosine perturbation; function optimization; inverse kinematics

[责任编校 陈留院 赵晓华]

附表Ⅰ　基准测试函数详细信息

Attached tab. Ⅰ　Benchmark function details

Benchmark function	Dim	Range	min
$f_1=\sum_{i=1}^{n}x_i^2$	30	[−100,100]	0.000 0
$f_2=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	[−10,10]	0.000 0
$f_3=\sum_{i=1}^{n}(\sum_{j=1}^{i}x_j)^2$	30	[−100,100]	0.000 0
$f_4=\max\{\lvert x_i\rvert,1\leq i\leq n\}$	30	[−100,100]	0.000 0
$f_5=\sum_{i=1}^{n}x_i^4+\mathrm{random}(0,1)$	30	[−1.28,1.28]	0.000 0
$f_6=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+2$	30	[−32,32]	0.000 0
$f_7=\frac{1}{4\ 000}\sum_{i=1}^{n}(x_i^2)-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	[−600,600]	0.000 0
$f_8=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)^{-1}$	2	[−65,65]	1.000 0
$f_9=\sum_{i=1}^{n}(\sum_{i=1}^{j}\lvert x_j\sin(x_i)+0.1x_j\rvert)$	4	[−10,10]	0.000 3
$f_{10}=\sum_{i=1}^{n}x_i^2(\sum_{j=1}^{n}0.5ix_i)^2+(\sum_{i=1}^{n}0.5ix_i)^2$	2	[−5,10]	0.398 0
$f_{11}=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]\times$ $[30+(2x_2-3x_2)^2(18-32x_1+15x_1^2+48x_2-36x_1x_2+27x_2^2)]$	2	[−5,5]	−3.860 0
$f_{12}=-\sum_{i=1}^{4}C_i\exp[\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2]$	4	[0.1]	−3.320 0

附表Ⅱ　实验结果

Attached tab. Ⅱ　Experimental results

Function	Index	DA	FPA	GWO	MFO	MIDA
f_1	Best	1.400E+03	7.582E-01	4.004E-83	4.346E-33	**1.065E-152**
	Worst	7.000E+03	1.709E+01	4.086E-73	7.478E-26	**6.537E-135**
	Mean	4.086E+03	5.689E+00	2.556E-74	5.660E-27	**2.673E-136**
	Std	1.656E+03	3.760E+00	9.158E-74	1.541E-26	**1.196E-135**
f_2	Best	7.207E+00	1.576E-01	8.047E-47	1.497E-17	**9.198E-77**
	Worst	3.033E+01	1.214E+00	1.524E-41	1.672E-14	**2.811E-68**
	Mean	1.470E+01	7.086E-01	1.694E-42	9.016E-16	**9.468E-70**
	Std	4.950E+00	2.469E-01	3.437E-42	3.063E-15	**5.131E-69**
f_3	Best	6.330E+02	1.387E+00	4.063E-48	8.975E-18	**2.538E-154**
	Worst	1.107E+04	1.486E+01	1.776E-39	5.563E-10	**1.280E-134**
	Mean	3.922E+03	4.484E+00	1.130E-40	4.799E-11	**7.638E-136**
	Std	2.295E+03	2.781E+00	3.658E-40	1.201E-10	**2.906E-135**
f_4	Best	2.147E+01	1.231E+00	3.050E-33	3.472E-12	**6.213E-75**
	Worst	5.190E+01	8.314E+00	3.223E-27	3.100E-06	**1.506E-67**
	Mean	3.873E+01	4.164E+00	2.853E-28	1.039E-07	**7.678E-69**
	Std	8.172E+00	1.672E+00	7.005E-28	5.659E-07	**2.902E-68**
f_5	Best	2.271E-01	2.732E-03	4.248E-05	2.594E-04	**6.139E-06**
	Worst	1.788E+00	3.582E-02	1.645E-03	7.427E-03	**6.501E-04**
	Mean	9.823E-01	1.398E-02	4.411E-04	2.034E-03	**1.751E-04**
	Std	4.559E-01	7.668E-03	3.962E-04	1.547E-03	**1.617E-04**
f_6	Best	1.233E+01	4.634E+00	4.441E-15	**8.882E-16**	**8.882E-16**
	Worst	2.002E+01	9.247E+00	4.441E-15	1.750E-13	**8.882E-16**
	Mean	1.751E+01	6.826E+00	4.441E-15	2.481E-14	**8.882E-16**
	Std	2.143E+00	1.208E+00	0.000E+00	3.451E-14	**0.000E+00**
f_7	Best	7.884E+00	3.529E-01	0.000E+00	2.958E-02	**0.000E+00**
	Worst	7.655E+01	9.828E-01	1.536E-01	3.769E-01	**0.000E+00**
	Mean	3.374E+01	6.205E-01	3.051E-02	1.254E-01	**0.000E+00**
	Std	1.469E+01	1.588E-01	3.602E-02	7.393E-02	**0.000E+00**
f_8	Best	1.305E+00	9.980E-01	**9.980E-01**	**9.980E-01**	**9.980E-01**
	Worst	2.211E+01	**2.747E+00**	1.267E+01	1.076E+01	2.982E+00
	Mean	1.089E+01	1.396E+00	4.261E+00	2.116E+00	**1.230E+00**
	Std	4.590E+00	**5.536E-01**	4.217E+00	2.023E+00	5.641E-01
f_9	Best	2.404E-03	5.924E-04	3.078E-04	6.430E-04	**3.075E-04**
	Worst	4.726E-02	2.533E-03	2.036E-02	8.334E-03	**1.166E-03**
	Mean	1.702E-02	1.197E-03	3.134E-03	1.343E-03	**7.296E-04**
	Std	1.052E-02	4.065E-04	6.877E-03	1.396E-03	**3.768E-04**
f_{10}	Best	−9.603E-01	**−1.032E+00**	**−1.032E+00**	**−1.032E+00**	**−1.032E+00**
	Worst	3.725E-01	−1.032E+00	−1.032E+00	−1.032E+00	**−1.032E+00**
	Mean	−4.368E-01	−1.032E+00	−1.032E+00	−1.032E+00	**−1.032E+00**
	Std	3.661E-01	9.589E-07	3.187E-08	**0.000E+00**	**0.000E+00**
f_{11}	Best	3.035E+00	3.000E+00	3.000E+00	3.000E+00	**3.000E+00**
	Worst	2.941E+01	3.056E+00	3.000E+00	3.000E+00	**3.000E+00**
	Mean	1.286E+01	3.005E+00	3.000E+00	3.000E+00	**3.000E+00**
	Std	7.972E+00	1.093E-02	8.445E-05	**4.801E-15**	1.754E-14
f_{12}	Best	−3.855E+00	**−3.863E+00**	**−3.863E+00**	**−3.863E+00**	**−3.863E+00**
	Worst	−3.410E+00	−3.861E+00	−3.855E+00	−3.855E+00	−3.863E+00
	Mean	−3.712E+00	−3.862E+00	−3.862E+00	−3.863E+00	−3.863E+00
	Std	1.228E-01	3.465E-04	1.839E-03	1.439E-03	**3.159E-14**