http://bhxb.buaa.edu.cn jbuaa@buaa.edu.cn
DOI: 10.13700/j.bh.1001-5965.2022.0492

多策略融合改进的自适应蜉蝣算法

蒋宇飞，许贤泽*，徐逢秋，高波
(武汉大学 电子信息学院，武汉 430072)

摘　　要：为改进蜉蝣算法全局搜索能力较差、种群多样性较小和自适应能力弱等问题，提出一种多策略融合改进的自适应蜉蝣算法 (MIMA)。采用 Sin 混沌映射初始化蜉蝣种群，使种群能够均匀分布在解空间中，提高初始种群质量，增强全局搜索能力；引入 Tent 混沌映射和高斯变异对种群个体进行调节，增加种群多样性的同时调控种群密度，增强局部最优逃逸能力；引入不完全伽马函数，重构自适应动态调节的重力系数，建立全局搜索和局部开发能力之间更好的平衡，进而提升算法收敛精度，有利于提高全局搜索能力；采用随机反向学习 (ROBL) 策略，增强全局搜索能力，提高收敛速度并增强稳定性。利用经典测试函数集进行算法对比，并利用 Wilcoxon 秩和检验分析算法的优化效果，证明改进的有效性和可靠性。实验结果表明：所提算法与其他算法相比，寻优精度、收敛速度、稳定性都取得了较大提升。

关 键 词：蜉蝣算法；混沌映射；高斯变异；自适应动态调节；随机反向学习
中图分类号：TP301.6
文献标志码：A　　**文章编号**：1001-5965(2024)04-1416-11

元启发式算法是对启发式算法的改进，是随机算法与局部搜索算法相结合的产物，如粒子群优化 (particle swarm optimization, PSO) 算法[1]、遗传算法 (genetic algorithm, GA)、麻雀搜索算法 (sparrow search algorithm, SSA)[2]、灰狼 (grey wolf optimization, GWO) 算法[3]、萤火虫算法 (firefly algorithm, FA)[4] 等。

蜉蝣算法 (mayfly algorithm, MA) 受蜉蝣的飞行行为和交配行为启发，在结构上结合 PSO、GA、FA 算法进行改进，是一种综合群体智能和进化算法主要优点的新型元启发式算法。改进蜉蝣算法 (improved mayfly algorithm, IMA) 由 Zervoudakis 和 Tsafarakis[5] 提出，针对 MA 的收敛稳定性问题、开发与探索之间不平衡的问题进行改进：①增加速度限制；②为重力系数设置阻尼；③为婚礼舞蹈和随机飞行设置阻尼；④对子代基因随机变异，IMA 在包括 13 个 CEC2017 测试函数的 38 个数学基准函数上均优于 MA。IMA 虽然继承了 MA 良好的性能和优势，并取得了较好的效果，但前期全局搜索能力不理想，导致后期即使具备较好的开发能力也难以达到理想的求解精度和算法稳定性，变异过程简单，种群多样性较小，易陷入局部最优解。

近年来，针对群体智能算法的缺点，研究人员提出了不同策略优化改进方案，以改善其全局搜索能力，增加种群多样性。文献 [6] 利用 Circle 混沌映射初始化鲸鱼优化算法的种群，使种群初始化过程中有更好的遍历性，降低了群体的盲目性，且引入反向学习 (opposition-based learning, OBL) 策略对鲸鱼的位置求出反向解，提高了鲸鱼优化算法的搜索能力和种群的多样性；文献 [7] 利用 Sin 混沌映射初始化 MA 的种群，获得更高质量初始解，证明了混沌映射方法对 MA 的有效性；文献 [8] 采用 Tent 混沌扰动和柯西变异优化种群，避免种群过于“集中”或“分散”，提高了种群的多样性，增强了探索能力；文献 [9] 通过引入不完全的伽马函数和

收稿日期：2022-06-16；**录用日期**：2022-09-23；**网络出版时间**：2022-10-10 10:36
网络出版地址：link.cnki.net/urlid/11.2625.V.20221009.1050.010
基金项目：国家自然科学基金 (51975422)
* **通信作者**. E-mail：xxz@whu.edu.cn

引用格式：*蒋宇飞，许贤泽，徐逢秋，等. 多策略融合改进的自适应蜉蝣算法 [J]. 北京航空航天大学学报，2024，50（4）：1416-1426.*
JIANG Y F，XU X Z，XU F Q，et al. Multi-strategy fusion improved adaptive mayfly algorithm [J]. Journal of Beijing University of Aeronautics and Astronautics，2024，50（4）：1416-1426（in Chinese）.

Beta 累加分布函数, 构建自适应动态调节的重力系数, 增强全局搜索和开发之间的平衡, 提升了算法的自适应动态搜索能力。

为改善 IMA 的全局搜索能力, 增加种群多样性, 提高搜索的自适应性, 本文提出多策略融合改进的自适应蜉蝣算法 (multi-strategy fusion improved adaptive mayfly algorithm, MIMA)。①使用 Sin 混沌映射初始化蜉蝣种群, 使蜉蝣种群在解空间中均匀分布, 提高前期全局搜索能力; ②引入 Tent 混沌映射和高斯变异对种群个体进行调节, 避免种群过于“集中”或“分散”, 提高种群的多样性, 增强局部最优逃逸能力; ③引入不完全伽马函数, 重构自适应动态调节的重力系数, 对全局搜索能力和局部开发能力建立更好的平衡; ④引入随机反向学习 (random opposition-based learning, ROBL) 策略, 增加种群的多样性, 补足全局搜索能力, 提高收敛速度并增强稳定性。通过经典测试函数集进行优化对比, 验证了 MIMA 的寻优精度、收敛速度、稳定性都取得较大提升。

1　改进蜉蝣算法原理

IMA 通过将最优雄性个体与最优雌性个体交配得到最优子代, 寻优过程为: 雄性蜉蝣的运动、雌性蜉蝣的运动、蜉蝣交配与变异。

1.1　雄性蜉蝣的运动

设雄性蜉蝣总数为 N, 每只雄性蜉蝣根据自己的经验和邻居的经验, 调整自身位置。假设 $\boldsymbol{x}_i^t$ 为第 i 个蜉蝣于时间步 t 在搜索空间 $\boldsymbol{U}$ 中的位置, 通过在当前位置上添加一个速度 $\boldsymbol{v}_i^{t+1}$ 来改变位置:

$$\boldsymbol{x}_i^{t+1}=\boldsymbol{x}_i^t+\boldsymbol{v}_i^{t+1} \tag{1}$$

式中: 雄性蜉蝣位置 $\boldsymbol{x}_i\in\boldsymbol{U}(\boldsymbol{x}_{\min},\boldsymbol{x}_{\max})$, 若添加速度 $\boldsymbol{v}_i^{t+1}$ 后超出 $\boldsymbol{U}$, 则将其限制回最近边界值。

雄性蜉蝣的速度 $\boldsymbol{v}_i^{t+1}$ 表示为

$$v_{i,j}^{t+1}=\begin{cases}gv_{i,j}^t+a_1\mathrm{e}^{-\beta r_p^2}(p_{\mathrm{best},i,j}-x_{i,j}^t)+\\ \quad a_2\mathrm{e}^{-\beta r_g^2}(g_{\mathrm{best},i}-x_{i,j}^t) & f(\boldsymbol{x}_i^t)>f_{\min}\\ gv_{i,j}^t+dr & f(\boldsymbol{x}_i^t)=f_{\min}\end{cases} \tag{2}$$

若改变后的速度超过范围 $[\boldsymbol{v}_{\min},\boldsymbol{v}_{\max}]$, 则将其限制回最近边界值。式中: $v_{i,j}^t$ 为第 i 个雄性蜉蝣于时间步 t 在维度 $j(j=1,\cdots,n)$ 上的速度; $x_{i,j}^t$ 为第 i 个蜉蝣于时间步 t 在维度 j 上的位置; g 为重力系数, 其迭代公式为

$$g^{t+1}=g^t g_{\mathrm{damp}} \tag{3}$$

其中: g^t 为时间步 t 时的重力系数; g_{damp} 为重力系数阻尼。

a_1 和 a_2 为正吸引常数, 分别用于缩放认知和社会部分的贡献, β 为固定的可见度系数, 用于限制蜉蝣的可见度, r_p、r_g 分别为 $\boldsymbol{x}_i$ 与 $\boldsymbol{p}_{\mathrm{best},i}$、$\boldsymbol{x}_i$ 与 $\boldsymbol{g}_{\mathrm{best}}$ 之间的笛卡儿距离, 计算公式为

$$\|\boldsymbol{x}_i-\boldsymbol{X}_i\|=\sqrt{\sum_{j=1}^{n}(x_{i,j}-X_{i,j})} \tag{4}$$

式中: $x_{i,j}$ 为蜉蝣 $\boldsymbol{x}_i$ 在维度 j 上的位置; $\boldsymbol{X}_i$ 对应 $\boldsymbol{p}_{\mathrm{best},i}$ 与 $\boldsymbol{g}_{\mathrm{best}}$。

f 表示适应度函数, 越小则函数越适应, $f_{\min}$ 即雄性蜉蝣中最小适应度函数值。最优位置雄性蜉蝣进行婚礼舞蹈, 改变速度, d 为婚礼舞蹈系数, r 为 [−1,1] 之间的随机数。d 的迭代公式为

$$d^{t+1}=d^t d_{\mathrm{damp}} \tag{5}$$

式中: d^t 为时间步 t 时的婚礼舞蹈系数; d_{damp} 为舞蹈系数阻尼。

1.2　雌性蜉蝣的运动

雌性蜉蝣最大的行为特征在于飞向雄性蜉蝣以繁殖。与雄性蜉蝣类似, 设雌蜉蝣总数为 N。

假设 $\boldsymbol{y}_i^t$ 为第 i 个雌性蜉蝣于时间步 t 在搜索空间 $\boldsymbol{U}$ 中的位置, 通过在当前位置上添加速度 $\boldsymbol{v}_i^{t+1}$ 来改变位置。雌雄蜉蝣位置范围为 $\boldsymbol{U}(\boldsymbol{x}_{\min},\boldsymbol{x}_{\max})$, 若添加速度 $\boldsymbol{v}_i^{t+1}$ 后超出 $\boldsymbol{U}$, 则将其限制回最近边界值。

吸引过程设定为最优雄性吸引最优雌性, 第二优雄性吸引第二优雌性。雌性蜉蝣速度为

$$v_{i,j}^{t+1}=\begin{cases}gv_{i,j}^t+a_3\mathrm{e}^{-\beta r_{\mathrm{mf}}^2}(x_{i,j}^t-y_{i,j}^t) & f(\boldsymbol{y}_i^t)>f(\boldsymbol{x}_i^t)\\ gv_{i,j}^t+lr & f(\boldsymbol{y}_i^t)\leqslant f(\boldsymbol{x}_i^t)\end{cases} \tag{6}$$

若改变后的速度超过范围 $[\boldsymbol{v}_{\min},\boldsymbol{v}_{\max}]$, 则将其限制回最近边界值。式中: $v_{i,j}^t$ 为第 i 个蜉蝣于时间步 t 在维度 $j(j=1,\cdots,n)$ 上的速度; $y_{i,j}^t$ 为第 i 个蜉蝣于时间步 t 在维度 j 上的位置; a_3 为正吸引常数; r_{mf} 为雌性蜉蝣与对应雄性蜉蝣之间的笛卡儿距离, 通过式 (4) 计算; l 为随机飞行系数, 当雌性蜉蝣没有被雄性蜉蝣吸引时使用, 此时雌性蜉蝣随机飞行, 其迭代公式为

$$l^{t+1}=l^t l_{\mathrm{damp}} \tag{7}$$

式中: l^t 为时间步 t 时的随机飞行系数; l_{damp} 为随机飞行系数阻尼。

1.3　蜉蝣交配及变异

交配过程以交叉算子表示, 从雄性蜉蝣中选择父本, 雌性蜉蝣中选取母本, 二者在本性别种群适应度排名相同。采用优胜劣汰机制, 将最优个体的雄性和雌性蜉蝣进行繁殖得到最优个体, 依次类推, 得到的 2 个子代表达式为

$$\begin{cases}\boldsymbol{c}_1=L\boldsymbol{m}+(1-L)\boldsymbol{f}+\sigma\boldsymbol{N}(0,1)\\ \boldsymbol{c}_2=L\boldsymbol{f}+(1-L)\boldsymbol{m}+\sigma\boldsymbol{N}(0,1)\end{cases} \tag{8}$$

式中：$\boldsymbol{c}_1$为雄性蜉蝣子代；$\boldsymbol{c}_2$为雌性蜉蝣子代；L为[−1,1]范围内服从高斯分布的随机数；$\boldsymbol{m}$为父本；$\boldsymbol{f}$为母本；$\sigma\boldsymbol{N}(0,1)$表示服从高斯分布的、均值为0、方差为1的随机数组成的$1\times n$矩阵。

2 本文算法

2.1 Sin混沌映射初始化种群

群体初始化对当前大多数智能优化算法的效率有很大影响，均匀分布的种群可以适度地扩大算法的搜索范围，从而提高收敛速度和求解精度。混沌映射具有随机性、遍历性和规律性特点，常用于优化搜索问题，以保持种群多样性及跳出局部最优解。本文以Sin混沌映射生成初始种群，定义如下：

$$y_{n+1}=\mu\sin(\pi y_n)\qquad 0\leqslant y_n\leqslant 1 \tag{9}$$

式中：y为迭代值；μ为[0,1]间的控制参数。

通过Sin混沌映射可以使生成的初始种群较为均匀地充斥整个解空间，提高初始种群质量，利于前期的全局搜索。

2.2 基于Tent混沌映射和高斯变异的种群调节

为增加种群多样性、提高求解进度，使算法不易在迭代末期陷入局部最优区域而无法脱出，需要在每个迭代环节末对种群进行调节。设f_i^{m}为蜉蝣适应度函数值，f_{a}为种群适应度函数值的平均值，按如下方式判断：

1）若$f_i^{\mathrm{m}}<f_{\mathrm{a}}$，即出现“聚集”现象，进行高斯变异，若新位置适应度函数值低于旧位置，则进行位置替换。

2）若$f_i^{\mathrm{m}}\geqslant f_{\mathrm{a}}$，即出现“发散”现象，进行Tent混沌映射，按同样原则进行位置替换。

2.2.1 Tent混沌映射

参考文献[8,10]，若利用混沌映射初始化种群后再对种群进行调节，使用多种混沌映射能避免多次陷入重复点，导致调节失效。Tent混沌映射定义如下：

$$y_{n+1}=\begin{cases}2y_n & y_n\in[0,0.5]\\ 0.5y_n & y_n\in(0.5,1]\end{cases} \tag{10}$$

但Tent混沌映射容易在小循环周期和不动点上出现问题，针对此，文献[11]在Tent混沌映射基础上引入随机变量，改进后表达式为

$$y_{n+1}=\begin{cases}2y_n+\mathrm{rand}(0,1)/M & y_n\in[0,0.5]\\ 0.5y_n+\mathrm{rand}(0,1)/M & y_n\in(0.5,1]\end{cases} \tag{11}$$

式中：M为混沌序列内粒子个数；rand(0,1)为[0,1]间的随机数。

2.2.2 高斯变异

高斯分布即正态分布，在自然界中，很多随机因素和影响都可以近似地用高斯分布来描述，许多概率分布可以也用高斯分布来近似或导出，其在统计学等诸多领域有重大的影响力。高斯变异[12]是在原有个体的状态上加一个服从高斯分布的随机向量，定义如下：

$$\mathrm{mutation}(\boldsymbol{x})=\boldsymbol{x}[1+\sigma\boldsymbol{N}(0,1)] \tag{12}$$

式中：$\boldsymbol{x}$为蜉蝣原本的位置；mutation($\boldsymbol{x}$)为变异后的蜉蝣位置。

由高斯分布特性可知，多数变异算子分布在原始位置周围，局部搜索能力强，能够提高优化算法的寻优精度，有利于跳出局部最优区域。同时，少数算子远离当前位置，提高了种群的多样性，有利于更好地搜索潜在区域，从而加快搜索速度，提高逃出局部最优解的概率。

2.3 不完全伽马函数的自适应重力系数

重力系数或称惯性权重，对算法的搜索能力和开发能力具有一定的指导作用，体现出蜉蝣能够借鉴一定先验行为的能力。文献[13]指出，较大的重力系数具有良好的全局搜索能力，较小的重力系数具有较好的局部开发能力。本文引入一种非线性递减的自适应重力系数，从而更好地平衡全局搜索和局部开发能力。为与IMA的重力系数g区别，设为$g(t)$，表达式为

$$g(t)=(1-t/T)^{\sqrt[q]{t/T}} \tag{13}$$

式中：t为当前时间步；T为最大迭代次数；α为重力系数的控制系数。

考虑到具有确定性数学表达式的重力系数按照固定规律搜索，很容易陷入局部最优，达不到真正的自适应机制。因此，引入不完全伽马函数[14]，重构自适应动态调节的重力系数$g'(t)$，其表达式如下：

$$g'(t)=(1-t/T)^{\sqrt[q]{t/T}}\Gamma(\lambda,1-t/T) \tag{14}$$

式中：Γ为不完全的伽马函数；λ为大于0的随机变量，取0.1。

雄性蜉蝣的速度表达式更新如下：

$$v_{i,j}^{t+1}=\begin{cases}g'(t)v_{i,j}^t+a_1\mathrm{e}^{-\beta r_p^2}(p_{\mathrm{best},i,j}-x_{i,j}^t)+\\ \quad a_2\mathrm{e}^{-\beta r_g^2}(g_{\mathrm{best}}-x_{i,j}^t) & f(\boldsymbol{x}_i^t)>f_{\min}\\ g'(t)v_{i,j}^t+dr & f(\boldsymbol{x}_i^t)=f_{\min}\end{cases} \tag{15}$$

若改变后的速度超过范围$[\boldsymbol{v}_{\min},\boldsymbol{v}_{\max}]$，则限制为边界值。

雌性蜉蝣的速度表达式更新如下：

$$v_{i,j}^{t+1}=\begin{cases}g'(t)v_{i,j}^t+a_3\mathrm{e}^{-\beta r_{\mathrm{mf}}^2}(x_{i,j}^t-y_{i,j}^t) & f(\boldsymbol{y}_i^t)>f(\boldsymbol{x}_i^t)\\ g'(t)v_{i,j}^t+lr & f(\boldsymbol{y}_i^t)\leqslant f(\boldsymbol{x}_i^t)\end{cases} \tag{16}$$

若改变后的速度超过范围[$\boldsymbol{v}_{\min}$,$\boldsymbol{v}_{\max}$], 则限制为边界值。

为确定合适的α值, 利用经典测试函数f_1对不同参数独立运行 50 次, 最大迭代 1 000 次, 设置维度为 20, 记录不同α值时的平均值、最优值和标准差。

由表 1 可知, 重力系数的控制系数α对算法的寻优能力有一定影响。当α = 1时, MIMA 可以表现出较好的搜索能力。

表 1　不同α值的搜索结果

Table 1　Search results with different α values

α	平均值	最优值	标准差
1	$1.427\,5\times10^{-40}$	$5.165\,1\times10^{-48}$	$4.424\,7\times10^{-40}$
1.5	$2.923\,9\times10^{-40}$	$3.755\,2\times10^{-47}$	$1.317\,3\times10^{-39}$
2	$1.043\,5\times10^{-39}$	$5.166\,7\times10^{-47}$	$4.285\,5\times10^{-39}$
2.5	$3.823\,0\times10^{-39}$	$8.348\,5\times10^{-50}$	$2.326\,0\times10^{-38}$
3	$1.739\,6\times10^{-38}$	$2.580\,9\times10^{-48}$	$6.760\,1\times10^{-38}$

2.4　随机反向学习策略

OBL 策略[15] 是一种于 2005 年提出的群智能领域中的改进策略, 其思想为: 在种群寻优的过程中, 以上下界之和减去当前解从而产生反向解, 比较当前解与反向解的适应度函数值, 择优进入下一次迭代。OBL 策略生成的反向解与当前解距离为定值, 缺乏随机性, 无法有效增强种群多样性。因此, Long 等[16] 提出了改进的 ROBL 策略, 加强全局搜索能力, 减少算法陷入局部最优解无法逃逸的概率。变异公式如下:

$$\text{mutation}(\boldsymbol{x}) = \boldsymbol{x}_{\min} + \boldsymbol{x}_{\max} - r\boldsymbol{x} \tag{17}$$

2.5　算法流程

本文提出的基于 Tent 混沌映射和高斯变异的种群调节及 ROBL 策略都采用贪心算法。简而言之, 即当且仅当变异后蜉蝣个体的适应度函数值减小, 才进行此次变异, 蜉蝣个体的位置才会因此发生改变。

综合引入的 4 项改进策略, 本文提出的 MIMA 流程如图 1 所示。

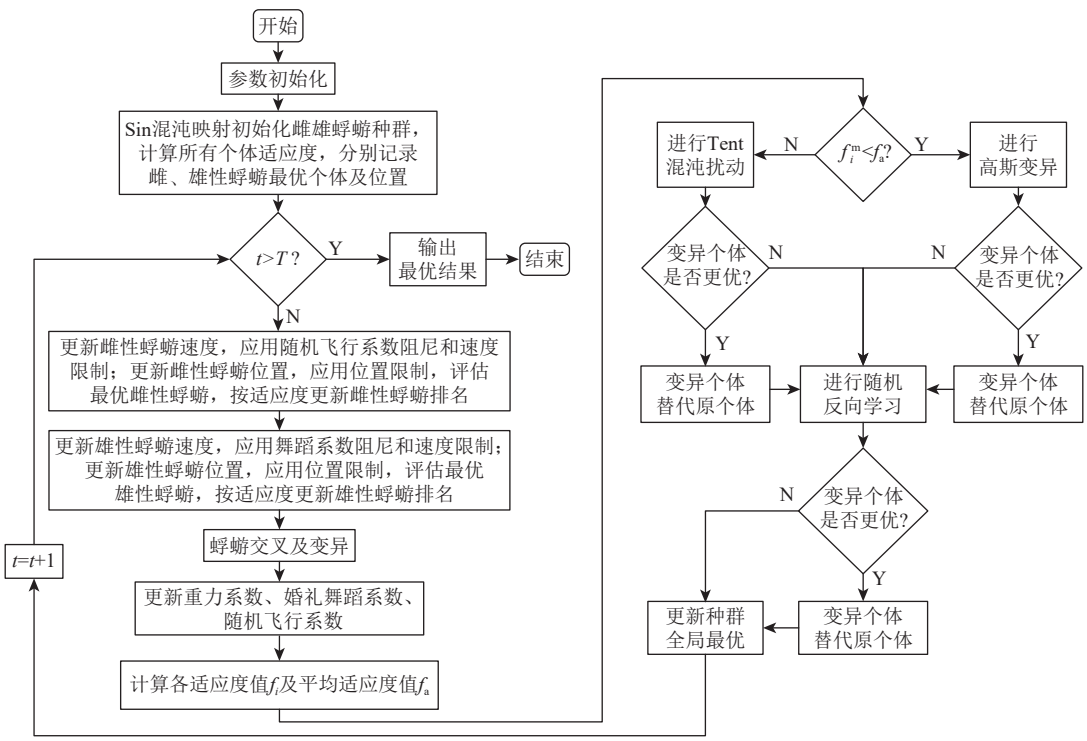

图 1　MIMA 算法流程

Fig. 1　MIMA Algorithm Flow Chart

3　算法性能测试

为验证 MIMA 的有效性和优异性, 本文以 IMA2[17]、TCMA[18] 2 种基于 IMA 的改进算法作为对比, 并引入 GWO 算法、正余弦优化算法 (sine cosine algorithm, SCA)、PSO 等经典优化算法进行复现比较。

3.1　参数设置

实验选取种群规模为 40(雄性蜉蝣 20, 雌性蜉蝣 20), 种群交叉个数为 20, 突变率为 0.01。各算法具体参数设置为: MIMA、IMA、IMA2 和 TCMA 中, α=1.0, l=1, l_{damp}=0.99, d=5, d_{damp}=0.8, a_1=1.0, a_2=1.5, a_3=1.5, β=2, g_0=0.8; PSO 算法中, c_1=1.494 45, c_2=1.494 45。

3.2 测试函数

实验利用14个经典基准测试函数进行测试，其中，f_1~f_6为不定维单峰函数、f_7~f_{11}为不定维多峰函数，f_{12}~f_{14}为固定维函数。表2为测试函数的具体信息。

3.3 结果分析

3.3.1 寻优精度和稳定性分析

在MATLAB R2018b环境下对所有测试函数进行仿真对比实验，不定维函数设置维度为20，实验均独立运行50次，最大迭代次数为1 000。记录各算法的平均值、最优值、标准差，测试结果如表3所示。

由表3可知，对于不定维单峰函数，除f_4外，MIMA相对其他所有算法具有显著性优势。针对f_4测试时，MIMA、IMA、TCMA均能找到理论最优解0，但三者小概率出现进入局部最优解而无法在1 000次迭代内跳出，从而导致平均值、标准差差别不大；而IMA2则可以稳定跳出该局部最优解。

对于不定维多峰函数，MIMA的各项测试指标除f_7、f_{11}外均显著优于其他算法。针对f_7测试时，MIMA逊于IMA2；针对f_{11}测试时，二者平均值较为接近，但MIMA方差更小，具有更强的稳定性和鲁棒性。

对于固定维函数，由于IMA本身实验效果已非常接近理论最优解，所有基于IMA的改进算法均难以取得大幅度的提升（仅IMA2在针对f_{13}进行测试时出现不能稳定获取最优解的情况），但MIMA在针对f_{12}、f_{13}测试时，稳定性相对更好。

综上所述，MIMA的优势直观上较为明显。

3.3.2 收敛曲线分析

为更直观地比较7种算法的收敛速度，分别用7种算法对测试函数进行实验，平均收敛曲线如图2所示。

由图2可知，对于图2(a)~图2(f)，MIMA寻优精度最高，除图2(d)外，MIMA收敛速度均优于其他算法。图2(d)中，起步阶段，MIMA、IMA、IMA2、

表2 基准测试函数

Table 2 Benchmark test function

函数公式	维度	搜索空间	$f_{\min}$
$f_1(x)=\sum_{i=1}^{n}x_i^2$	20	[−100,100]	0
$f_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	20	[−10,10]	0
$f_3(x)=\max\{\lvert x_i\rvert\}$	20	[−100,100]	0
$f_4(x)=\sum_{i=1}^{n}(\lvert x_i+0.5\rvert)^2$	20	[−100,100]	0
$f_5(x)=\sum_{i=1}^{n}x_i^4+\mathrm{random}[0,1)$	20	[−1.28.1.28]	0
$f_6(x)=\sum_{i=1}^{n}\lvert x_i\rvert^{i+1}$	20	[−1,1]	0
$f_7(x)=-\sum_{i=1}^{n}\left[x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)\right]$	20	[−500,500]	−8 379.66
$f_8(x)=\sum_{i=1}^{n}\left(x_i^2-10\cos(2\pi x_i)+10\right)$	20	[−5.12,5.12]	0
$f_9(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+e$	20	[−32,32]	0
$f_{10}(x)=\frac{1}{4\,000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	20	[−600,600]	0
$f_{11}(x)=\frac{\pi}{n}\left[10\sin^2(\pi y_i)+\sum_{i=1}^{n-1}(y_i-1)^2\left(1+10\sin^2(\pi y_{i+1})\right)+(y_n-1)^2\right]+\sum_{i=1}^{n}u(x_i,10,100,4)$	20	[−50,50]	0
$f_{12}(x)=\left[0.002+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}\left(x_i-a_{ij}\right)^6}\right]^{-1}$	2	[−65.53,65.53]	1
$f_{13}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1\left(b_i^2+b_ix_2\right)}{b_i^2+b_ix_3+x_4}\right]^2$	4	[−5,5]	3.075×10^{-4}
$f_{14}(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-11\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos x_1+10$	2	[−5,10]	0.398

表 3　基准测试函数结果对比

Table 3　Comparison of benchmark test function results

测试函数	平均值						
	MIMA	IMA	PSO	GWO	SCA	IMA2	TCMA
f_1	1.486×10^{-192}	6.023×10^{-39}	2.007	1.945×10^{-65}	2.717×10^{-8}	1.229×10^{-44}	2.310×10^{-44}
f_2	7.613×10^{-98}	2.106×10^{-16}	8.188×10^{-1}	2.540×10^{-38}	1.844×10^{-9}	1.576×10^{-24}	6.570×10^{-21}
f_3	9.818×10^{-64}	2.106×10^{-3}	8.653×10^{-1}	3.919×10^{-16}	5.631×10^{-1}	4.224×10^{-4}	2.976×10^{-4}
f_4	4.930×10^{-34}	1.972×10^{-33}	2.283	4.616×10^{-1}	2.048	0	2.465×10^{-33}
f_5	1.254×10^{-4}	5.163×10^{-3}	5.900×10^{-3}	6.322×10^{-4}	6.508×10^{-3}	3.237×10^{-3}	4.164×10^{-3}
f_6	1.864×10^{-294}	7.675×10^{-74}	4.774×10^{-8}	2.650×10^{-221}	4.803×10^{-9}	2.972×10^{-72}	2.413×10^{-73}
f_7	-7.445×10^{3}	-6.881×10^{3}	-5.169×10^{2}	-6.130×10^{3}	-3.234×10^{3}	-8.380×10^{3}	-6.557×10^{3}
f_8	0	2.815	1.909×10^{1}	4.887×10^{-1}	2.379	3.317×10^{-1}	4.345
f_9	8.882×10^{-16}	1.182	1.840	1.489×10^{-14}	8.627	2.599×10^{-14}	1.614
f_{10}	0	3.780×10^{-2}	1.764×10^{-1}	1.776×10^{-3}	6.454×10^{-2}	2.254×10^{-2}	2.146×10^{-16}
f_{11}	2.382×10^{-32}	3.421×10^{-2}	4.423×10^{-2}	3.328×10^{-2}	2.622×10^{-1}	2.408×10^{-32}	5.183×10^{-3}
f_{12}	9.980×10^{-1}	9.980×10^{-1}	1.267×10^{1}	4.131	9.982×10^{-1}	9.980×10^{-1}	9.980×10^{-1}
f_{13}	3.075×10^{-4}	3.075×10^{-4}	6.628×10^{-4}	4.376×10^{-3}	9.713×10^{-4}	1.645×10^{-3}	3.075×10^{-4}
f_{14}	3.979×10^{-1}	3.979×10^{-1}	3.979×10^{-1}	3.979×10^{-1}	3.987×10^{-1}	3.979×10^{-1}	3.979×10^{-1}
测试函数	最优值						
	MIMA	IMA	PSO	GWO	SCA	IMA2	TCMA
f_1	5.722×10^{-193}	8.786×10^{-49}	6.672×10^{-1}	8.155×10^{-68}	1.571×10^{-14}	1.468×10^{-51}	2.017×10^{-51}
f_2	1.450×10^{-99}	1.962×10^{-26}	5.817×10^{-1}	3.137×10^{-39}	6.787×10^{-13}	2.663×10^{-30}	5.440×10^{-27}
f_3	1.010×10^{-97}	5.682×10^{-5}	4.711×10^{-1}	8.026×10^{-18}	6.846×10^{-3}	4.710×10^{-5}	1.529×10^{-5}
f_4	0	0	1.175	1.410×10^{-5}	1.579	0	0
f_5	1.227×10^{-7}	1.762×10^{-3}	9.874×10^{-4}	1.539×10^{-4}	4.763×10^{-4}	1.334×10^{-3}	1.468×10^{-3}
f_6	0	3.927×10^{-93}	4.576×10^{-9}	9.344×10^{-231}	9.462×10^{-22}	1.576×10^{-93}	1.037×10^{-86}
f_7	-7.926×10^{3}	-7.787×10^{3}	-1.506×10^{3}	-8.067×10^{3}	-3.824×10^{3}	-8.380×10^{3}	-7.432×10^{3}
f_8	0	5.684×10^{-14}	8.689	0	5.684×10^{-14}	0	9.950×10^{-1}
f_9	8.882×10^{-16}	2.931×10^{-14}	1.015	7.994×10^{-15}	1.877×10^{-8}	7.994×10^{-15}	6.401×10^{-6}
f_{10}	0	0	4.317×10^{-2}	0	8.378×10^{-13}	0	0
f_{11}	2.356×10^{-32}	2.356×10^{-32}	7.751×10^{-3}	6.514×10^{-3}	1.546×10^{-1}	2.356×10^{-32}	2.356×10^{-32}
f_{12}	9.980×10^{-1}	9.980×10^{-1}	1.267×10^{1}	9.980×10^{-1}	9.980×10^{-1}	9.980×10^{-1}	9.980×10^{-1}
f_{13}	3.075×10^{-4}	3.075×10^{-4}	3.166×10^{-4}	3.075×10^{-4}	3.436×10^{-4}	3.075×10^{-4}	3.075×10^{-4}
f_{14}	3.979×10^{-1}	3.979×10^{-1}	3.979×10^{-1}	3.979×10^{-1}	3.979×10^{-1}	3.979×10^{-1}	3.979×10^{-1}
测试函数	标准差						
	MIMA	IMA	PSO	GWO	SCA	IMA2	TCMA
f_1	0	1.887×10^{-37}	7.375×10^{-1}	3.560×10^{-65}	1.025×10^{-7}	2.807×10^{-44}	7.722×10^{-44}
f_2	1.161×10^{-97}	1.043×10^{-14}	1.153×10^{-1}	2.434×10^{-38}	4.370×10^{-9}	6.994×10^{-24}	2.832×10^{-20}
f_3	4.279×10^{-63}	6.188×10^{-3}	1.758×10^{-1}	6.989×10^{-16}	1.159	7.688×10^{-4}	1.555×10^{-4}
f_4	2.415×10^{-33}	7.542×10^{-33}	7.023×10^{-1}	2.647×10^{-1}	2.190×10^{-1}	0	1.121×10^{-32}
f_5	1.318×10^{-4}	2.830×10^{-3}	3.497×10^{-3}	3.159×10^{-4}	5.658×10^{-3}	1.292×10^{-3}	2.459×10^{-3}
f_6	0	4.000×10^{-73}	3.824×10^{-8}	0	1.413×10^{-8}	1.590×10^{-71}	8.527×10^{-73}
f_7	2.102×10^{2}	3.168×10^{2}	2.053×10^{2}	6.476×10^{2}	2.169×10^{2}	1.012×10^{-11}	3.688×10^{2}
f_8	0	1.888	5.034	1.946	5.766	6.957×10^{-1}	1.441
f_9	0	7.940×10^{-1}	5.516×10^{-1}	2.602×10^{-15}	9.000	1.917×10^{-14}	7.649×10^{-1}
f_{10}	0	5.098×10^{-2}	6.514×10^{-2}	5.066×10^{-3}	1.382×10^{-1}	2.578×10^{-2}	3.280×10^{-16}
f_{11}	5.825×10^{-34}	7.800×10^{-2}	3.926×10^{-2}	1.549×10^{-2}	4.723×10^{-2}	1.187×10^{-33}	2.791×10^{-2}
f_{12}	7.022×10^{-17}	1.088×10^{-16}	2.139×10^{-10}	4.107	4.398×10^{-4}	9.065×10^{-17}	1.147×10^{-16}
f_{13}	3.048×10^{-19}	1.870×10^{-19}	2.497×10^{-4}	7.997×10^{-3}	4.014×10^{-4}	5.003×10^{-3}	3.278×10^{-19}
f_{14}	3.331×10^{-16}	3.331×10^{-16}	2.714×10^{-5}	4.969×10^{-5}	8.580×10^{-4}	3.331×10^{-16}	3.331×10^{-16}

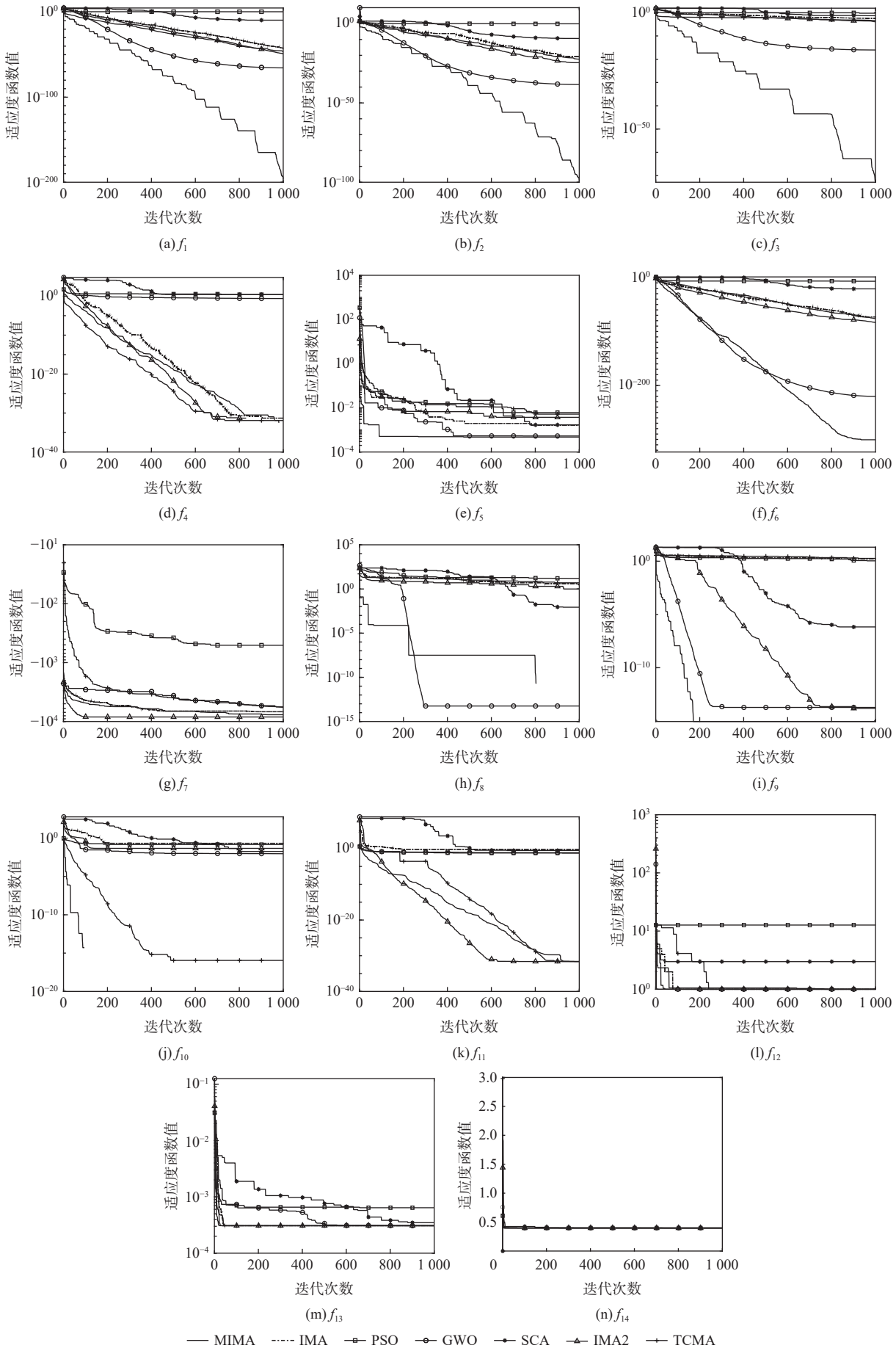

图2 测试函数平均收敛曲线

Fig. 2 Average convergence curves of test functions

TCMA 均能快速收敛, 大概率收敛至 0, 而 IMA2 更为稳定, 其次为 MIMA。图 2(e) 中, MIMA 起步即快速收敛, 但被 GWO 追上, 不过 MIMA 寻优精度优于 GWO。在其他测试函数中, MIMA 的收敛曲线处于所有曲线左下角, 寻优精度远超其他算法几十个数量级, 表现出优异的收敛速度和寻优精度。这与表 3 中数据相互印证。

对于不定维多峰函数(见图 2(g)~图 2(k)), 多数情况下, MIMA 收敛曲线明显远离其他算法, 处于左下角, 即各项指标皆远超其他算法。图 2(g) 中, 由于 SCA 效果太差而舍弃, IMA2 表现优异达到理论值, 而 MIMA 收敛曲线整体处于 IMA 下方, 寻优精度优于 IMA, 仅次于 IMA2。图 2(h) 中, MIMA 收敛曲线因陷入局部最优解而部分处于 GWO 上方, 但最终能跳出局部最优解收敛至理论最优解 0, 而 GWO 不能在迭代中跳出局部最优。图 2(k) 中, MIMA 在迭代初期收敛速度最快, 中期被 IMA2 赶上, 但 IMA2 迭代中后期陷入停滞, 最终 MIMA、IMA2、TCMA 三者收敛精度相近。

对于固定维函数(见图 2(l)~图 2(n)), 5 种算法收敛曲线较为接近, 但仍能看出 MIMA 的起步收敛速度较快, 一般处于所有曲线的左下方。

整体而言, 所有测试函数平均收敛曲线均能明确显示 MIMA 收敛曲线的断层现象, 表现出极强的局部最优逃逸能力; 而另外 2 种基于 IMA 的改进算法由于变异手段较为简单, 未能如 MIMA 一样多种变异手法互相补足, 导致其陷入局部最优解后往往最多只能跳出三四次; 其他经典算法的收敛曲线则基本上只能较为平滑地下降。此外, MIMA 的收敛曲线起点一般较低, 即初始解质量较高, 在图 2(b)、图 2(d)、图 2(h)、图 2(i) 中表现尤为明显, 这是加入 Sin 混沌映射初始化的结果; 在图 2(j) 中, 使用 Tent 混沌映射初始化的 TCMA 与 MIMA 收敛曲线起点接近, 并且明显远离其他算法的起点, 说明混沌映射初始化能够较为有效地提高初始解质量。

综上所述, MIMA 在 14 个基准测试函数上的收敛速度、寻优精度、全局搜索能力、局部最优逃逸能力、稳定性等 5 个指标上, 综合强于常用经典优化算法和其他基于 IMA 的改进算法。由此可见, 本文算法具有更强的寻优能力。

3.3.3　Wilcoxon 秩和检验分析

仅根据算法运行 50 次后的平均值、标准差等指标对算法性能进行评价往往不够准确, 为体现算法改进的鲁棒性和公平性, 利用统计检验来评估本文算法相比于其他算法的优越性。因此, 在 5% 的显著水平下进行 Wilcoxon 秩和检验。表 4 给出了 14 个测试函数的 MIMA 运行结果与其他算法的 Wilcoxon 秩和检验中计算的 p 值。当 $p < 0.05$ 时, 即可被认为是拒绝零假设的有力验证[19]。由表 4 数据可知, 绝大部分的秩和检验 p 值小于 0.05, 说明 MIMA 的性能较之其他算法而言具有显著优异的收敛性能。

3.3.4　消融实验分析

由于引入 4 项改进机制, MIMA 的寻优性能较为优异。为明确各改进机制的作用, 对 MIMA 进行消融实验, 对不同策略进行实验测试。本文引入 2 个实验测试。基础的改进蜉蝣算法记为 IMA, 在 IMA 基础上引入 Sin 混沌映射初始化蜉蝣种群、Tent 混沌映射和高斯变异对种群个体进行调节策略的算法记为 CMA, 在 IMA 基础上引入自适应

表 4　Wilcoxon 秩和检验 p 值

Table 4　p-value for Wilcoxon's rank-sum results

测试函数	IMA	PSO	GWO	SCA	IMA2	TCMA
f_1	$3.019\,9\times10^{-11}$	$3.019\,9\times10^{-11}$	$3.019\,9\times10^{-11}$	$3.019\,9\times10^{-11}$	$3.019\,9\times10^{-11}$	$3.019\,9\times10^{-11}$
f_2	$3.019\,9\times10^{-11}$	$3.019\,9\times10^{-11}$	$3.019\,9\times10^{-11}$	$3.019\,9\times10^{-11}$	$3.019\,9\times10^{-11}$	$3.019\,9\times10^{-11}$
f_3	$3.019\,9\times10^{-11}$	$3.019\,9\times10^{-11}$	$3.019\,9\times10^{-11}$	$3.019\,9\times10^{-11}$	$3.019\,9\times10^{-11}$	$3.019\,9\times10^{-11}$
f_4	$7.457\,7\times10^{-9}$	$2.363\,8\times10^{-12}$	$2.363\,8\times10^{-12}$	$2.363\,8\times10^{-12}$	$1.607\,4\times10^{-1}$	$3.019\,9\times10^{-11}$
f_5	$3.019\,9\times10^{-11}$	$3.019\,9\times10^{-11}$	$7.389\,1\times10^{-11}$	$3.019\,9\times10^{-11}$	$3.019\,9\times10^{-11}$	$3.019\,9\times10^{-11}$
f_6	$6.478\,9\times10^{-12}$	$6.478\,9\times10^{-12}$	$6.478\,9\times10^{-12}$	$6.478\,9\times10^{-12}$	$6.478\,9\times10^{-12}$	$6.478\,9\times10^{-12}$
f_7	$2.530\,6\times10^{-4}$	$3.019\,9\times10^{-11}$	$9.063\,2\times10^{-8}$	$3.019\,9\times10^{-11}$	$2.430\,6\times10^{-11}$	$2.194\,7\times10^{-8}$
f_8	$2.530\,6\times10^{-4}$	$3.019\,9\times10^{-11}$	$9.063\,2\times10^{-8}$	$3.019\,9\times10^{-11}$	$4.527\,0\times10^{-12}$	$1.211\,8\times10^{-12}$
f_9	$1.210\,8\times10^{-12}$	$1.211\,8\times10^{-12}$	$2.034\,1\times10^{-13}$	$1.211\,8\times10^{-12}$	$1.060\,9\times10^{-12}$	$1.211\,8\times10^{-12}$
f_{10}	$5.772\,0\times10^{-11}$	$1.211\,8\times10^{-12}$	$8.152\,3\times10^{-2}$	$1.211\,8\times10^{-12}$	$1.656\,0\times10^{-11}$	$2.911\,2\times10^{-2}$
f_{11}	$1.152\,9\times10^{-9}$	$2.972\,9\times10^{-11}$	$2.972\,9\times10^{-11}$	$2.972\,9\times10^{-11}$	$3.195\,8\times10^{-7}$	$1.473\,5\times10^{-7}$
f_{12}	$3.049\,0\times10^{-1}$	$3.157\,8\times10^{-12}$	$1.988\,1\times10^{-11}$	$4.390\,9\times10^{-12}$	$7.805\,6\times10^{-1}$	$6.324\,1\times10^{-1}$
f_{13}	NaN	$1.211\,8\times10^{-12}$	$1.211\,8\times10^{-12}$	$1.211\,8\times10^{-12}$	NaN	$8.140\,4\times10^{-2}$
f_{14}	NaN	$1.211\,8\times10^{-12}$	$1.211\,8\times10^{-12}$	$1.211\,8\times10^{-12}$	NaN	NaN

注: NaN表示算法性能相当。

动态调节的重力系数和ROBL策略的算法记为GMA。分别以IMA、CMA、GMA、MIMA在测试函数上作对比实验,结果如表5所示。

从表5中可以看出,在不定维单峰函数f_1~f_6

表5 消融实验结果

Table 5 Experimental results of ablation

测试函数	平均值			
	IMA	CMA	GMA	MIMA
f_1	6.023×10^{-39}	4.459×10^{-160}	8.516×10^{-156}	1.486×10^{-192}
f_2	2.106×10^{-16}	1.159×10^{-94}	1.604×10^{-82}	7.613×10^{-98}
f_3	2.106×10^{-3}	2.058×10^{-6}	2.844×10^{-51}	9.818×10^{-64}
f_4	1.972×10^{-33}	7.396×10^{-34}	1.761×10^{-31}	4.930×10^{-34}
f_5	5.163×10^{-3}	2.098×10^{-3}	1.379×10^{-4}	1.254×10^{-4}
f_6	7.675×10^{-74}	6.998×10^{-222}	7.204×10^{-217}	1.864×10^{-294}
f_7	-6.881×10^{3}	-7.233×10^{3}	-7.105×10^{3}	-7.445×10^{3}
f_8	2.815	1.678	0	0
f_9	1.182	4.370×10^{-15}	8.882×10^{-16}	8.882×10^{-16}
f_{10}	3.780×10^{-2}	0	0	0
f_{11}	3.421×10^{-2}	1.662×10^{-2}	1.960×10^{-31}	2.382×10^{-32}
f_{12}	9.980×10^{-1}	9.980×10^{-1}	9.980×10^{-1}	9.980×10^{-1}
f_{13}	3.075×10^{-4}	1.310×10^{-3}	3.075×10^{-4}	3.075×10^{-4}
f_{14}	3.979×10^{-1}	3.979×10^{-1}	3.979×10^{-1}	3.979×10^{-1}
测试函数	最优值			
	IMA	CMA	GMA	MIMA
f_1	8.786×10^{-49}	4.162×10^{-173}	4.062×10^{-196}	5.722×10^{-193}
f_2	1.962×10^{-26}	3.206×10^{-95}	4.226×10^{-98}	1.450×10^{-99}
f_3	5.682×10^{-5}	1.443×10^{-24}	9.776×10^{-98}	1.010×10^{-97}
f_4	0	0	0	0
f_5	1.762×10^{-3}	3.955×10^{-4}	5.288×10^{-6}	1.227×10^{-7}
f_6	3.927×10^{-93}	1.580×10^{-232}	7.363×10^{-229}	0
f_7	-7.787×10^{3}	-7.906×10^{3}	-7.906×10^{3}	-7.926×10^{3}
f_8	5.684×10^{-14}	0	0	0
f_9	2.931×10^{-14}	8.882×10^{-16}	8.882×10^{-16}	8.882×10^{-16}
f_{10}	0	0	0	0
f_{11}	2.356×10^{-32}	2.356×10^{-32}	2.404×10^{-32}	2.356×10^{-32}
f_{12}	9.980×10^{-1}	9.980×10^{-1}	9.980×10^{-1}	9.980×10^{-1}
f_{13}	3.075×10^{-4}	3.075×10^{-4}	3.075×10^{-4}	3.075×10^{-4}
f_{14}	3.979×10^{-1}	3.979×10^{-1}	3.979×10^{-1}	3.979×10^{-1}
测试函数	标准差			
	IMA	CMA	GMA	MIMA
f_1	1.887×10^{-37}	2.425×10^{-159}	5.945×10^{-155}	0
f_2	1.043×10^{-14}	1.268×10^{-94}	1.123×10^{-81}	1.161×10^{-97}
f_3	6.188×10^{-3}	9.700×10^{-6}	1.240×10^{-50}	4.279×10^{-63}
f_4	7.542×10^{-33}	2.927×10^{-33}	5.428×10^{-31}	2.415×10^{-33}
f_5	2.830×10^{-3}	1.350×10^{-3}	1.045×10^{-4}	1.318×10^{-4}
f_6	4.000×10^{-73}	0	0	0
f_7	3.168×10^{2}	2.789×10^{2}	4.038×10^{2}	2.102×10^{2}
f_8	1.888	2.873	0	0
f_9	7.940×10^{-1}	4.974×10^{-16}	0	0
f_{10}	5.098×10^{-2}	0	0	0
f_{11}	7.800×10^{-2}	5.627×10^{-2}	5.236×10^{-31}	5.825×10^{-34}
f_{12}	1.088×10^{-16}	5.439×10^{-17}	2.259×10^{-10}	7.022×10^{-17}
f_{13}	1.870×10^{-19}	4.371×10^{-3}	2.341×10^{-19}	3.048×10^{-19}
f_{14}	3.331×10^{-16}	3.331×10^{-16}	0	3.331×10^{-16}

中, 除 f_4、f_5外, CMA、GMA、MIMA 结果均显著优于 IMA, CMA、GMA 均使寻优性能有一定提升; 在 f_4中, GMA 略逊于 IMA, 但 CMA、GMA 结合后, MIMA 的平均值最好, 稳定性最高; 在 f_5中, CMA、GMA 平均值、均方差略优于 IMA, 因此, MIMA 相对于 IMA 的总体提升幅度不大。

在不定维多峰函数 f_7~f_{11}中, CMA、GMA 对 MIMA 的寻优能力提升显著, 仅 f_7因为理论最优解约为−8 379.66 而使得显示结果不够明显; 在 f_8、f_{11}中, CMA 不能稳定找到理论最优解 0, 对于平均值提升幅度不大, 但 GMA 可以稳定收敛至理论最优解 0, 对 MIMA 影响更大。

在固定维函数 f_{12}~f_{14}中, 由于每种算法都无限接近理论最优解, MIMA 的性能提升幅度不明显, 但 CMA、GMA、MIMA 的结果在 16 位小数显示状态下可以看到均不逊于 IMA。

综上所述, 通过模型消融实验证明本文所提出的每一种策略都对 MIMA 的寻优性能有所帮助。

4　结　论

本文综合 4 种改进策略, 基于改进蜉蝣算法, 提出了一种多策略融合改进的自适应蜉蝣算法。

1) 本文算法在 14 种经典基准测试函数上, 相较于 IMA2、TCMA 2 种基于 IMA 的改进算法和 GWO、SCA、PSO 等经典优化算法, 有更加出色的寻优能力和稳定性。

2) 通过 Wilcoxon 秩和检验分析, 验证了本文算法相比于其他算法的优越性, 通过消融实验验证了所采用改进策略的有效性。

综上, 本文算法相比其他算法, 在寻优精度、收敛速度、稳定性上都取得了较大提升, 但其使用了诸多映射方法, 若将算法应用于复杂约束问题, 会存在困难。因此, 对于多目标复杂问题的求解, 并将其应用复杂工程设计问题中, 是未来的研究重点。

参考文献 (References)

[1] RICCI F, ROKACH L, SHAPIRA B. Recommender systems handbook[M]. Berlin: Springer, 2011.

[2] XUE J K, SHEN B. A novel swarm intelligence optimization approach: Sparrow search algorithm[J]. Systems Science & Control Engineering, 2020, 8(1): 22-34.

[3] PANDA M, DAS B, PATI B B. Grey wolf optimization for global path planning of autonomous underwater vehicle[C]//Proceedings of the 3rd International Conference on Advanced Informatics for Computing Research. New York: ACM, 2019: 1-6.

[4] YANG X S. Firefly algorithms for multimodal optimization[C]// Proceedings of the International Symposium on Stochastic Algorithms. Berlin: Springer, 2009: 1-10.

[5] ZERVOUDAKIS K, TSAFARAKIS S. A mayfly optimization algorithm[J]. Computers & Industrial Engineering, 2020, 145: 106559.

[6] 张达敏, 徐航, 王依柔, 等. 嵌入 Circle 映射和逐维小孔成像反向学习的鲸鱼优化算法[J]. 控制与决策, 2021, 36(5): 1173-1180.
ZHANG D M, XU H, WANG Y R, et al. Whale optimization algorithm for embedded Circle mapping and one-dimensional oppositional learning based small hole imaging[J]. Control and Decision, 2021, 36(5): 1173-1180(in Chinese).

[7] 胡健, 刘祥敏, 毛伊敏, 等. 基于 KD 树和混沌蜉蝣优化的并行谱聚类算法[J]. 计算机集成制造系统, 2023, 29(12): 4001-4020.
HU J, LIU X M, MAO Y M, et al. Parallel spectral clustering algorithm using KD tree and chaotic mayfly optimization algorithm[J]. Computer Integrated Manufacturing Systems, 2023, 29(12): 4001-4020(in Chinese).

[8] 唐延强, 李成海, 宋亚飞, 等. 自适应变异麻雀搜索优化算法[J]. 北京航空航天大学学报, 2023, 49(3):681-692.
TANG Y Q, LI C H, SONG Y F, et al. Adaptive mutation sparrow search optimization algorithm[J]. Journal of Beijing University of Aeronautics and Astronautics, 2023, 49(3):681-692(in Chinese).

[9] 王义, 张达敏, 邹诚诚. 增强全局搜索和自适应蜉蝣算法[J]. 哈尔滨工业大学学报, 2022, 54(11): 137-150.
WANG Y, ZHANG D M, ZOU C C. Enhance global search and adaptive mayfly algorithm[J]. Journal of Harbin Institute of Technology, 2022, 54(11): 137-150(in Chinese).

[10] 吕鑫, 慕晓冬, 张钧, 等. 混沌麻雀搜索优化算法[J]. 北京航空航天大学学, 2021, 47(8): 1712-1720.
LU X, MU X D, ZHANG J, et al. Chaos sparrow search optimization algorithm[J]. Journal of Beijing University of Aeronautics and Astronautics, 2021, 47(8): 1712-1720(in Chinese).

[11] 张娜, 赵泽丹, 包晓安, 等. 基于改进的 Tent 混沌万有引力搜索算法[J]. 控制与决策, 2020, 35(4): 893-900.
ZHANG N, ZHAO Z D, BAO X A, et al. Gravitational search algorithm based on improved Tent chaos[J]. Control and Decision, 2020, 35(4): 893-900(in Chinese).

[12] WU K L, YANG M S. Alternative c-means clustering algorithms[J]. Pattern Recognition, 2002, 35(10): 2267-2278.

[13] SHI Y. A modified particle swarm optimizer[C]//Proceedings of the IEEE International Conference on Evolutionary Computation. Piscataway: IEEE Press, 1998: 69-73.

[14] 王梦娜, 王秋萍, 王晓峰. 基于 Iterative 映射和单纯形法的改进灰狼优化算法[J]. 计算机应用, 2018, 38(S2): 16-20.
WANG M N, WANG Q P, WANG X F. Improved grey wolf optimization algorithm based on Iterative mapping and simplex method[J]. Journal of Computer Applications, 2018, 38(S2): 16-20 (in Chinese).

[15] TIZHOOSH H R. Opposition-based learning: A new scheme for machine intelligence[C]//Proceedings of the International Conference on International Conference on Computational Intelligence for Modelling, Control & Automation. Piscataway: IEEE Press, 2005: 695-701.

[16] LONG W, JIAO J, LIANG X, et al. A random opposition-based learning grey wolf optimizer[J]. IEEE Access, 2019, 7: 113810-113825.

[17] 邵瑞凝, 杨博, 束洪春, 等. 基于改进蜉蝣算法的光伏阵列最优重构方法[J]. 电力系统自动化, 2022, 14(11): 142-150.

SHAO R N, YANG B, SHU H C, et al. Improved mayfly algorithm for optimal PV array reconfiguration[J]. Automation of Electric Power Systems, 2022, 14(11): 142-150(in Chinese).

[18] 徐焕增, 徐文倩, 孔政敏. 基于 Tent 混沌序列改进的蜉蝣算法及其应用[J]. 控制工程, 2022, 29(3): 435-440.
XU H Z, XU W Q, KONG Z M. Mayfly algorithm based on Tent chaotic sequence and its application[J]. Control Engineering of China, 2022, 29(3): 435-440(in Chinese).

[19] DERRAC J, GARCIA S, MOLINA D, et al. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms[J]. Swarm & Evolutionary Computation, 2011, 1(1): 3-18.

Multi-strategy fusion improved adaptive mayfly algorithm

JIANG Yufei，XU Xianze*，XU Fengqiu，GAO Bo

(School of Electronic Information，Wuhan University，Wuhan 430072，China)

Abstract：This paper proposes the multi-strategy fusion improved adaptive mayfly algorithm (MIMA), which addresses the shortcomings of the improved mayfly algorithm, including its low adaptive ability, minimal population diversity, and poor global search performance. Firstly, Sin chaos mapping was used to initialize the mayfly population so that the population could be uniformly distributed in the solution space, which improved the initial population quality and enhanced the global search ability. Second, in order to improve the local optimal escape ability, control population density, and boost population diversity, individuals in the population were exposed to Gaussian variation and Tent chaos mapping. Then, the incomplete gamma function was introduced to reconstruct the adaptive dynamic adjustment of gravity coefficients to establish a better balance between global search and local exploitation ability, which in turn improved the convergence accuracy of the algorithm and facilitated the potential of global search to find the optimal solution. Finally, the random opposition-based learning (ROBL) strategy was adopted to enhance the global search ability, improve the convergence speed and enhance the stability. To demonstrate the efficacy and dependability of the four improvement measures, the algorithms were compared using the classical test function set and their optimization effect was examined using the Wilcoxon rank sum test. The experimental results show that compared with other algorithms, the MIMA has better searching accuracy, convergence speed, and stability.

Keywords：mayfly algorithm；chaotic mapping；Gaussian mutation；adaptive dynamic adjustment；random opposition-based learning

Received：2022-06-16；**Accepted**：2022-09-23；**Published Online**：2022-10-10 10：36

URL：link.cnki.net/urlid/11.2625.V.20221009.1050.010

Foundation item：National Natural Science Foundation of China (51975422)

* **Corresponding author**. E-mail：xxz@whu.edu.cn