混合柯西变异和均匀分布的蝗虫优化算法

何 庆[1,2†], 林 杰[1,2], 徐 航[1]

(1. 贵州大学 1 大数据与信息工程学院;2. 贵州省公共大数据重点实验室, 贵阳 550025)

摘 要: 由于位置更新公式存在的局部开发能力较强而全局探索能力较弱的缺陷, 导致蝗虫优化算法 (GOA) 易陷入局部最优, 易早熟收敛. 因此提出混合柯西变异和均匀分布的蝗虫优化算法 (HCUGOA), 受柯西算子和粒子群算法的启发, 提出具有分段思想的位置更新方式以增加种群多样性, 增强全局探索能力; 将柯西变异算子和反向学习策略相融合, 对最优位置即目标值进行变异更新, 提高算法跳出局部最优的能力; 为了更好的平衡全局探索和局部开发, 将均匀分布函数引入到非线性控制参数 c, 构建新的随机调整策略. 通过对 12 个基准函数和 CEC2014 函数进行仿真实验以及 Wilcoxon 秩和检验的方法来评估改进算法的寻优能力, 实验结果表明 HCUGOA 算法在收敛精度和收敛速度等方面得到了极大的改进.

关键词: 蝗虫优化算法; 粒子群算法; 柯西变异; 均匀分布; 反向学习; 时间复杂度; 统计检验

中图分类号: TP301 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2019.1609

Hybrid Cauchy Mutation and Uniform Distribution of Grasshopper Optimization Algorithm

HE Qing[1,2†], *LIN Jie*[1,2], *XU Hang*[1]

(1. College of Big Data & Information Engineering;2. Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China)

Abstract: Due to the strong local exploitation ability and the weak global exploration ability of the location update formula, the grasshopper optimization algorithm (GOA) was easy to fall into local optimum and easy to prematurely converge. Therefore, this paper proposed a hybrid Cauchy mutation and uniform distribution of grasshopper optimization algorithm (HCUGOA). Firstly, inspired by the cauchy operator and particle swarm optimization algorithm, a location update method with segmentation idea was proposed to increase the diversity of the population and to enhance the global exploration ability. Secondly, the fusion of cauchy mutation and opposition-based learning, the variation of the optimal position, which was, the target value, improved the ability of the algorithm to jump out of the local optimum; Finally, in order to better balance the global exploration and local exploitation, the uniform distribution function was introduced into the nonlinear control parameter c, so that could build a new random adjustment strategy. The optimization performance of the improved algorithm is evaluated by a sets of simulation experiments and Wilcoxon’s test on 12 benchmark functions and modern CEC 2014 functions. The experimental results show that the HCUGOA algorithm has been greatly improved in terms of convergence accuracy and convergence speed.

Keywords: grasshopper optimization algorithm;particle swarm optimization algorithm;Cauchy mutation;uniform distribution;opposition-based learning;time complexity;statistical test

0 引 言

蝗虫优化算法 (GOA)[1] 是由 Mirjalili 教授等人于 2017 年提出的新型群智能优化算法. 虽 GOA 提出时间较短, 但 GOA 不仅与 GA[2]、ACO[3]、PSO[4] 等传统的优化算法相比具有竞争性, 与 WOA[5]、ALO[6]、SSA[7]、SCA[8] 等新型群智能算法相比也具有一定竞争性. 而且 GOA 已成功应用于多个工程领域 [9-12].

GOA 具有原理简单, 易于实现等优点, 但该算法仍存在易陷入局部最优和易早熟收敛等缺点, 因而众多学者提出了许多改进优化算法. 文献[13] 利用动态权重机制促进算法开发, 并引入随机跳跃策略避免算法陷入局部最优; 文献[14] 采用高斯变异和 Levy-

收稿日期: 2019-11-18；**修回日期:** 2020-03-15.

基金项目: 贵州省科技计划项目重大专项项目 (黔科合重大专项字 [2018]3002，黔科合重大专项字 [2016]3022)；贵州省公共大数据重点实验室开放课题 (2017BDKFJJ004)；贵州省教育厅青年科技人才成长项目 (黔科合 KY 字 [2016]124；贵州大学培育项目 (黔科合平台人才 [2017]5788).

†通讯作者. E-mail: qhe@gzu.edu.cn.

fight 策略来增强种群多样性; 文献[15] 提出 MGOA, 通过变异更新和随机突变来增大搜索区域以避免陷入局部值; 文献[16] 引入自适应策略而提出的 SA-CGOA 具有更好的求解质量和收敛速度.

针对 GOA 存在的问题, 本文提出混合柯西变异和均匀分布的蝗虫优化算法 (HCUGOA), 并通过 5 组仿真实验证明本文提出算法的有效性和优越性, 实验结果表明 HCUGOA 能显著提高算法寻优性能.

1 蝗虫优化算法

蝗虫优化算法 (GOA) 仿生原理: 根据蝗虫成虫大范围搜索和幼虫小范围移动, 并整体向着食物源 (即目标) 移动的方式, 实现蝗虫位置优化过程, 其中将蝗虫个体映射为搜索空间的点; 用数学模型来模拟个体间相互作用; 寻找食物源为寻优过程; 蝗虫位置映射为解, 位置的优劣映射为适应度; 位置不断更新映射为优化过程中最优解的选取过程.

GOA 的具体实现步骤如下:

1) 初始化种群大小 N、空间维度 dim、最大迭代次数 $T_{\max}$ 及参数 $c_{\max}$ 和 $c_{\min}$, 初始化种群位置 X_i,$i=(1,2,3,\ldots,N)^{\dim}$;

2) 计算个体适应度, 并将最优值存到目标值 $\hat{T}_d$;

3) 通过公式 (1) 更新参数 c:

$$c=c_{\max}-t\frac{c_{\max}-c_{\min}}{T_{\max}} \tag{1}$$

其中 c_{max},$c_{\min}$ 是参数 c 的最大值和最小值,$T_{\max}$ 是最大迭代次数,t 是当前迭代次数;

4) 通过式 (2) 更新蝗虫位置:

$$x_i^d=c\left(\sum_{\substack{j=1\\ j\neq i}}^{N} c\frac{ub_d-lb_d}{2}s\left(\left|x_j^d-x_i^d\right|\right)\frac{x_j-x_i}{d_{ij}}\right)+\hat{T}_d \tag{2}$$

式 (2) 中 x_i^d 表第 i 只蝗虫在第 d 维的位置,$d=(1,2,,\ldots,\dim)$,ubd、lbd 是第 i 只蝗虫在第 d 维的上下界,$\hat{T}_d$ 为蝗虫群的目标位置,$d_{ij}=\left|x_j^d-x_i^d\right|$ 表第 i 只与第 j 只蝗虫在第 d 维的距离.$(x_j-x_i)/d_{ij}$ 表第 i 只蝗虫与第 j 只蝗虫的单位向量.s 是个体之间的相互影响函数:

$$s(r)=f\mathrm{e}^{\frac{-r}{l}}-\mathrm{e}^{-r} \tag{3}$$

式中,f、l 分别为吸引强度参数与吸引尺度参数, 本文取值 $f=0.5$,$l=1.5$;

5) 计算每个个体的适应度, 并择优更新目标值;

6) 判断是否到达最大迭代次数, 若是, 则停止循环进行步骤 (7), 否则, 重复 (3)~(5);

7) 输出目标位置 $\hat{T}_d$ 和全局最优值.

2 HCUGOA 算法

2.1 服从均匀分布随机调整策略

GOA 的控制参数 c 是协调探索和开发的关键, 由式 (1) 可知,c 是随着迭代次数动态变化的, 且是线性递减函数, 在搜索过程中难以适应优化的实际情况: 在迭代前期参数 c 下降过快易导致蝗虫不能遍历更多搜索空间, 最终导致全局探索不足; 在迭代后期 c 下降过慢, 会导致局部开发受限, 收敛速度慢. 为解决这一问题, 并重新构建新函数 $c(t)$:

$$c(t)=\delta(\lg(\frac{c_{\max}}{T_{\max}})^{-20t}+\exp(\frac{t}{T_{\max}})^{-20(c_{\max}-c_{\min})}) \tag{4}$$

其中 δ 是服从 $[0,T_{\max}]$ 之间均匀分布的随机数, 且 δ 的取值范围介于 0 到 1 之间.

由式 (4) 可知, $c(t)$ 是非线性递减函数,$c(t)$ 随迭代次数增加在迭代期间呈非线性状态减小, 具体表现为在前期函数下降缓慢, 给全局探索充分时间, 后期函数迅速下降收敛, 加快算法收敛速度; 同时, 引入服从均匀分布随机数 δ, 前期让参数 c 有可能继续保持较大值, 一定程度上加强全局搜索; 后期参数 c 随着迭代次数增加而减小, 且变化比较缓慢, 此时服从均匀分布的随机数可能产生较大的值让参数 c 实现动态变化, 从而加快算法开发和收敛速度.

2.2 结合柯西算子和分段思想的更新策略

由式 (2) 可知, 基本 GOA 位置更新由自身位置与其他所有蝗虫间交互力及目标位置决定, 因此 GOA 具有很强的局部开发能力而全局探索能力不足. 为更好地提升算法性能, 本节对算法进行分段处理.

在算法搜索前半段, 受 PSO 对每代最优解进行记忆保存的思想启发, 对式 (2) 进行改进:

$$\begin{aligned}X_{new}(t+1)=&b1*W+b2*r1(X_{best}(t)-X_i(t))\\&+b2*r2(X_j(t)-X_k(t))\end{aligned} \tag{5}$$

其中 $X_{new}(t+1)$ 是第 i 只蝗虫在第 $t+1$ 代的位置,$X_{best}(t)$ 是第 t 代最优位置,$X_i(t)$ 是第 i 只蝗虫在第 t 代的位置,$X_j(t)$ 和 $X_k(t)$ 是第 t 代两个随机位置,$r1$ 和 $r2$ 分别为区间 $[0,1]$ 内产生的随机数,$b1$ 是记忆系数,$b2$ 是信息交流系数:

$$b1=\exp(c(t)-30\frac{t}{T_{\max}})^{-t} \tag{6}$$

$$b2=(\frac{T_{\max}-t}{T_{\max}})^{N} \tag{7}$$

其中 W 是与所有蝗虫相互联系决定位置更新的步

长函数, 其余参数已在第一章阐述:

$$W=\sum_{\substack{j=1\\ j\neq i}}^{N} c(t)\frac{ub_d-lb_d}{2}s\left(\left|x_j^d-x_i^d\right|\right)\frac{x_j-x_i}{d_j} \tag{8}$$

与式 (2) 相比, 式 (5) 分为 3 个部分:1) 利用除最优解外的全部位置信息, 有助于个体间进行信息交互, 增强全局探索能力;2) 由自身位置和最优位置决定, 加强算法开发能力和收敛速度;3) 通过随机选择两个个体引导位置更新, 增强全局探索能力.

在算法搜索后半段, 引入柯西算子作为变异步长, 当个体陷入局部最优时, 较大的步长可帮助个体跳出局部极值; 当个体接近收敛, 并在搜索最优解时, 较小步长可加速个体的收敛, 公式如下:

$$X_i{}^d(t+1)=X_i{}^d(t)+Cauchy\oplus(X_{best}{}^d(t)-X_k{}^d(t)) \tag{9}$$

其中 X_k^d 是随机选择的第 k 只蝗虫在第 d 维的位置,$Cauchy$ 是柯西算子, 一维标准的柯西分布的概率密度函数表达式如下:

$$f(x)=\frac{1}{\pi}(\frac{1}{x^2+1}),-\infty<x<\infty \tag{10}$$

由数学知识可知, 柯西算子具有较长的“尾巴”, 两端较长的分布可使个体具有更高概率跳到更好位置, 使个体逃离局部最优, 同时, 中心点较小的峰值表明柯西算子搜索领域空间的时间花费更少.

判断算法是进行前半段探索还是后半段开发由控制概率 P_c 决定,P_c 的表示为:

$$P_c=\frac{1}{2}\mathrm{T}_{\max} \tag{11}$$

算法前期需进行大范围全局探索, 后期需进行小范围的局部开发并避免算法过早收敛, 所以对位置更新引入分段思想, 前期对最优解进行记忆保存, 加强全局寻优能力, 并引入记忆系数和交流系数, 通过个体自身位置、最优位置及两个随机位置共同决定新位置; 后期引入柯西算子, 一定程度上降低算法陷入局部最优概率, 并加快算法的收敛速度.

算法 1 结合柯西算子和分段思想的策略

1) if rand$<P_c$

2) 根据式 (5)~(8) 更新个体位置;

3) else

4) 根据式 (9) 和 (10) 更新个体位置;

5) end

2.3 融合柯西变异和反向学习策略

基本 GOA 中, 目标位置更新依赖于每次迭代后的位置更新, 重新计算适应度, 选择最优位置对目标位置进行取代, 未对目标位置进行主动的扰动更新, 导致算法易陷入局部最优. 所以, 本节融合柯西变异和反向学习策略, 依概率对目标位置进行随机扰动更新, 避免算法陷入局部最优.

反向学习是 Tizhoosh[17] 在 05 年提出的新技术, 目的是基于当前解, 寻到其对应反向解, 通过评估选择保存更好的解. 为更好的引导个体寻到最优解, 将反向学习融入 GOA 中, 数学描述如下:

$$X^*{}_{best}(t)=ub+r\oplus(lb-X_{best}(t)) \tag{12}$$

$$X_{new}(t+1)=b3\oplus(X_{best}(t)-X^*{}_{best}(t)) \tag{13}$$

其中 $X_{new}^*(t)$ 是第 t 代目标解 X_{best} 的反向解,$X_{new}(t+1)$ 是第 $t+1$ 代目标解,ub 和 lb 是上下界,r 是服从 $(0,1)$ 标准均匀分布 $1\times dim$ 的随机数矩阵 (其中 dim 表算法搜索空间维数),$b3$ 是伪信息交流系数, 表达式如下:

$$b3=(\frac{T_{\max}-t}{T_{\max}})^t \tag{14}$$

将柯西算子引入目标位置更新, 发挥柯西算子的调节能力, 增强算法跳出局部最优的能力:

$$X_{new}(t+1)=cauchy\oplus X_{best}(t) \tag{15}$$

为提高算法寻优性能, 将反向学习策略和柯西算子扰动策略在一定概率下交替执行, 动态的随机更新目标位置. 反向学习策略中, 通过一般反向学习策略得到反向解, 增大算法搜索范围, 同时, 式 (12) 中的上下界 ub 和 lb 是动态变化的, 相对于固定边界的策略更利于算法优化; 柯西变异策略中运用变异算子对最优位置进行变异产生新解, 一定程度上改善算法易陷入局部最优的缺陷. 决定选择何种策略进行更新的选择概率 P_s 定义如下:

$$P_s=-\exp(1-\frac{t}{T_{\max}})^{20}+\theta \tag{16}$$

其中 θ 为调节因子, 经多次实验,θ 取值为 0.05 时函数优化结果最优.

算法 2 融合柯西变异和反向学习策略

1) if rand$<P_s$

2) 根据式 (12)~(14) 反向学习策略更新目标位置;

3) else

4) 根据公式 (15) 柯西变异策略更新目标位置;

5) end

对目标位置进行扰动更新虽能让算法跳出局部最优, 但不能保证新位置优于原目标位置, 因此在扰动更新操作后加入贪婪机制, 通过比较新旧目标位置的适应度后再决定是否更新目标位置.

算法 3 贪婪算法

1) if $f(X_{new})<f(X_{best})$

2) $X_{best} = X_{new}$;

3) else

4) $X_{best} = X_{best}$

5) end

基于贪婪选择策略促进算法向着期望寻找到的目标位置方向进化, 让目标位置能够充分发挥引导作用, 使算法获得更好的收敛速度和精度.

2.4 HCUGOA 算法的实现步骤

综上所述, 本文提出的改进算法的流程图如图 1 所示:

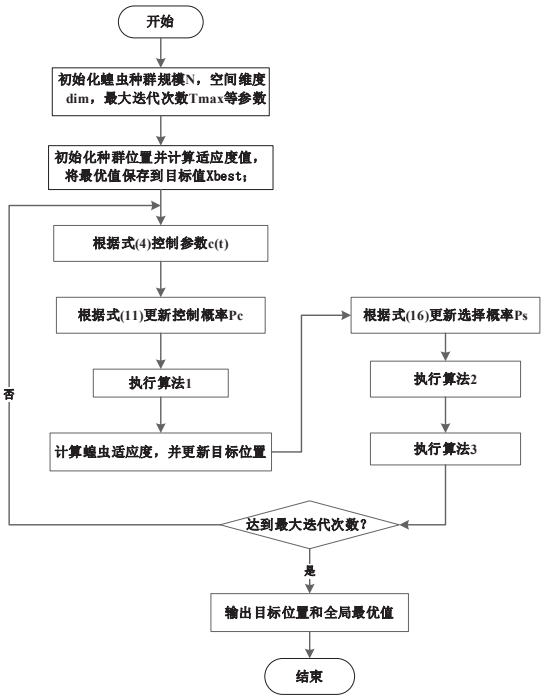


图 1 HCUGOA 算法流程图

2.5 算法时间复杂度分析

时间复杂度间接反映算法的收敛速度. 在 GOA 中, 假设参数初始化 (种群规模 N, 空间维度 n 等参数) 所需执行时间为 x_1, 产生均匀分布随机数时间为 x_2, 求给定适应度函数时间为 $f(n)$, 则 GOA 种群初始阶段的时间复杂度为:

$$O(x_1 + N(nx_2 + f(n))) = O(n + f(n)) \tag{17}$$

设更新函数 c 的时间为 x_3, 每一维按公式 (2) 更新位置所需时间为 x_4, 比较位置优劣和择优更新目标位置的时间分别为 x_5 和 x_6, 该阶段的时间复杂度为:

$$O(N(x_3+nx_4+f(n)+x_5)+x_6)=O(n+f(n)) \tag{18}$$

所以基本 GOA 求解最优解的总时间复杂度为:

$$T(n)=O(n+f(n))+O(n+f(n))=O(n+f(n)) \tag{19}$$

在 HCUGOA 中, 初始化参数所需时间与基本 GOA 相同, 因此 HCUGOA 在初始化阶段的时间复杂度与式 (17) 一致. 在算法循环部分, 设更新函数 $c(t)$ 时间为 w_1, 更新 P_c 时间为 w_2, 执行算法 1 时间为 w_3, 更新 P_s 时间为 w_4, 执行算法 2 和算法 3 时间为 w_5 和 w_6, 则循环部分的时间复杂度为:

$$\begin{aligned} &O(w_1+w_2+N(nw_3+f(n)+x_5)+x_6+w_4+N(w_5+w_6)) \\ &= O(n + f(n)) \end{aligned} \tag{20}$$

由上分析可得 HCUGOA 求解每一代最优解的总时间复杂度为:

$$T(n)=O(n+f(n))+O(n+f(n))=O(n+f(n)) \tag{21}$$

综上所述,HCUGOA 和基本 GOA 相比, 时间复杂度一致, 本文针对 GOA 缺陷所提出的改进策略没有增加算法时间复杂度.

3 仿真实验与结果分析

为全面验证本文算法性能, 同时验证每种改进策略的有效性, 本文的仿真实验分为五个部分进行:

a) 将 HCUGOA 与基本 GOA 比较, 通过实验数据对比分析本文改进算法的有效性和可行性;

b) 将 HCUGOA 与改进控制参数 c 的 GOA1、结合柯西算子和分段思想的更新策略的 GOA2、融合柯西变异和反向学习策略的 GOA3 进行对比, 验证不同改进策略的有效性;

c) 将 HCUGOA 与混合策略改进鲸鱼算法 (MS-WOA)[5]、正弦余弦改进算法 (m-SCA)[8] 以及最新的蝗虫改进算法 IGOA[14] 和 MGOA[15] 进行比较, 验证本文改进算法的优越性和竞争性;

d) 通过 Wilcoxon 秩和检验验证本文算法与其他对比算法之间的显著性差异;

f) 在 CEC2014 基准函数中选取部分单峰、多峰、混合和复合类型的函数进行优化测试, 验证本文算法的有效性和鲁棒性.

实验引入 12 个具有不同寻优特征的经典测试函数, 如表 1 所示, 其中 F1~F6 是连续单峰函数,F7~F11 是复杂非线性多峰函数,F12 是固定低维函数. 单峰函数局部最优即为全局最优, 用来测试算法收敛速度和收敛精度; 多峰函数具有对个局部极值, 常用来测试算法跳出局部最优的能力和全局搜索能力.

为公平起见, 算法基本参数设置相同: 种群规模 N=30, 空间维数 dim=10/30/200, 最大迭代次数 $T_{\max}$=500. 算法具体参数设置: 对于 GOA, $c_{\max}$=1, $c_{\min}$=0.00004; 对于 MS-WOA, $a_{\max}$=2, $a_{\min}$=0; 对于 m-SCA,a=2; 对于 GOA3 和 HCUGOA, θ =0.05, 其他改

进 GOA 与基本 GOA 设置一致.

表 1 测试函数

编号	函数名	维度	定义域	理论值
F1	Sphere	10/30/200	[-100,100]	0
F2	Schwefel 2.22	10/30/200	[-10,10]	0
F3	Schwefel 1.2	10/30/200	[-100,100]	0
F4	Schwefel 2.21	10/30/200	[-100,100]	0
F5	Rosenbrock	10/30/200	[-30,30]	0
F6	Quartic	10/30/200	[-1.28,1.28]	0
F7	Schwefel2.26	10/30/200	[-500,500]	-12569.5
F8	Rastrigin	10/30/200	[-5.12,5.12]	0
F9	Ackley	10/30/200	[-32,32]	0
F10	Griewank	10/30/200	[-600,600]	0
F11	Penalized	10/30/200	[-50,50]	0
F12	Kowalik	4	[-5,5]	0.0003

3.1 与基本 GOA 算法比较

将基本 GOA 与 HCUGOA 在空间维数 dim=30 的条件下对 10 个测试函数进行求解, 并通过 5 个性能评估仿真结果, 算法独立运行 30 次的数据如表 2 所示.

表 2 HCUGOA 与 GOA 结果比较

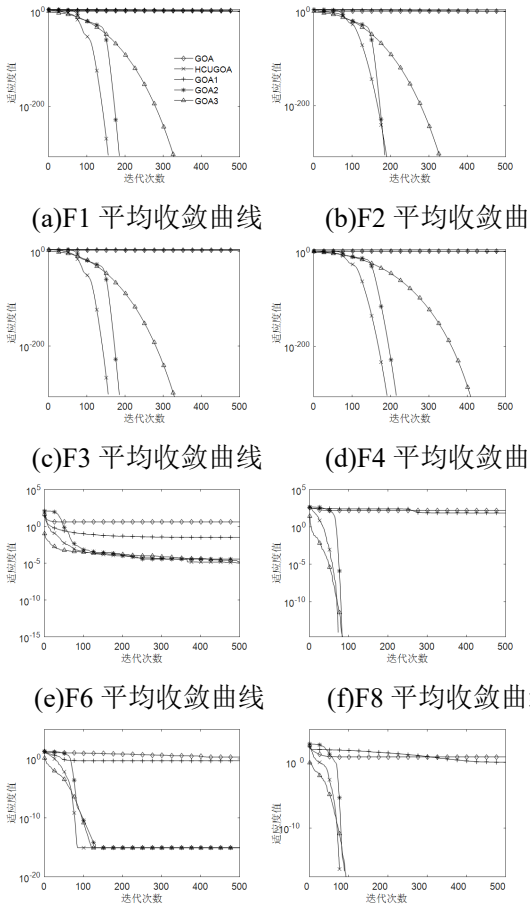
	算法	最优值	最差值	均值	标准差	耗时/s
F1	GOA	2.77E+02	9.83E+02	7.02E+02	2.03E+02	53.7818
	HCUGOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	28.7335
F2	GOA	1.12E+01	2.86E+01	2.05E+01	4.58E+00	55.3564
	HCUGOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	29.2753
F3	GOA	1.99E+03	4.50E+03	3.18E+03	7.73E+02	53.5119
	HCUGOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	28.4700
F4	GOA	2.05E+01	3.71E+01	2.50E+01	4.00E+00	52.6925
	HCUGOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	27.7006
F5	GOA	1.71E+04	1.17E+05	7.05E+04	3.00E+04	52.1765
	HCUGOA	1.78E-03	2.12E+01	6.89E+00	7.26E+00	27.9789
F6	GOA	3.00E+00	5.68E+00	3.96E+00	7.06E-01	52.8676
	HCUGOA	4.96E-07	2.31E-05	1.08E-05	6.64E-06	28.1102
F7	GOA	-6.74E+03	-5.11E+03	-6.29E+03	4.39E+02	52.8227
	HCUGOA	-9.99E+122	-3.44E+92	-4.37E+121	-1.85E+122	28.1537
F8	GOA	9.83E+01	1.87E+02	1.57E+02	2.19E+01	52.8676
	HCUGOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	28.1102
F9	GOA	1.16E+00	3.40E+00	2.09E+00	4.88E-01	53.0531
	HCUGOA	8.88E-16	8.88E-16	8.88E-16	1.00E-31	28.0339
F10	GOA	1.67E+00	9.61E+00	6.60E+00	2.27E+00	53.0192
	HCUGOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	28.2026

最优值和最差值反映算法寻优结果的质量. 从表 2 数据可知, 对于函数 F1、F2、F3、F4、F8、F10,HCUGOA 均寻到理论最优值 0; 对于函数 F5 和 F6,HCUGOA 虽没有寻到理论值, 但是与 GOA 相比最优值均相差 7 个数量级相; 对于函数 F9, 其理论值为 0,HCUGOA 没有寻到理论值, 但是与基本 GOA 相比, 具有更高的收敛精度; 对于函数 F7, 与基本 GOA 相比,HCUGOA 能够得到更优的结果.

平均值反映算法收敛速度, 标准差反映算法稳定性和鲁棒性. 对于函数 F6, 虽未寻到理论值, 但从平均值和标准差知 HCUGOA 具有更高的寻优精度且寻优结果稳定. 对于 F9, 与基本 GOA 相比, 标准差相差 32 个数量级, 说明 HCUGOA 对 F9 求解时, 得到的高精度结果并非偶然. 总之, 对于 10 个测试函数, 与 GOA 相比,HCUGOA 能够取得更优的结果, 具有更强的有效性和可行性. 另外从平均耗时可看出, 本文改进算法的平均耗时比原始算法少, 证明本文提出的改进策略没有增加算法时间复杂度.

3.2 与不同改进策略对比

将 HCUGOA 与改进控制参数 $c(t)$ 的 GOA1、结合柯西算子和分段思想的更新策略算法 GOA2、融合柯西变异和反向学习策略算法 GOA3 进行对比, 进一步验证不同改进策略的有效性. 算法参数与 3.1 节相同, 算法独立运行 30 次的仿真结果如表 3 所示.



(a)F1 平均收敛曲线 (b)F2 平均收敛曲线

(c)F3 平均收敛曲线 (d)F4 平均收敛曲线

(e)F6 平均收敛曲线 (f)F8 平均收敛曲线

(g)F9 平均收敛曲线 (h)F10 平均收敛曲线

图 2 不同策略算法的平均收敛曲线

由表 3 可知,HCUGOA 对单峰函数寻优时, 前四个函数的四个性能指标均达到理论值, 而对于 F5 和 F6,HCUGOA 的收敛精度也得到了一定程度提

升. 对于多峰函数,HCUGOA 对 F8 和 F10 寻到理论值 0,F7 和 F9 虽未寻到最优, 但相对于其他改进算法,HCUGOA 有更高的求解精度和稳定性;HCUGOA 对 F11 求解时, 标准差没有 GOA3 的显著, 说明前两种改进策略相对抑制了 HCUGOA 对 F11 的寻优.

具体来说,GOA2 和 GOA3 对 F1~F4、F8 和 F10 的有效性显著, 从前 4 个评价指标可看出结合柯西算子和分段思想的更新策略和融合柯西变异和反向学习策略可有效提高算法寻优能力.GOA1 对函数的寻优结果是 3 种改进策略算法中最差的, 但其收敛精度相对 GOA 也得到了一定提升, 尤其是 F6 和 F9, 得到寻优精度更高的结果. 表 3 的评价指标证明了在单峰和多峰函数中, 融合 3 个改进策略的 HCUGOA 对于增加种群多样性、有效提高算法收敛精度的有效性.

另外, 从运行一次算法的平均耗时来看, 相同维度下, 结合柯西算子和分段思想的算法 GOA2 耗时最短, 说明在 GOA 中加入策略 2 之后, 在不同的时期采用不同的更新公式有效增强了算法的全局探索能力; 加入柯西变异和反向学习的算法 GOA3 和加入均匀分布的策略算法 GOA1 的耗时无显著性差别, 但均比基本 GOA 耗时短, 证明了另外两种改进策略的有效性, 另外,HCUGOA 耗时要比 GOA2 稍长, 这是因为在 GOA 中加入了三种改进策略之后, 找到的解更多, 导致寻优时间变长, 即耗时变长.

表 3 不同改进策略对比结果

	算法	最优值	最差值	均值	标准差	耗时/s
	GOA	2.77E+02	9.83E+02	7.02E+02	2.03E+02	53.7818
	GOA1	4.49E+00	3.00E+01	1.96E+01	5.85E+00	52.3237
F1	GOA2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	**26.8390**
	GOA3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	50.3823
	HCUGOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	28.7335
	GOA	1.12E+01	2.86E+01	7.02E+02	4.58E+00	55.3564
	GOA1	2.49E+00	8.42E+00	1.96E+01	1.84E+00	53.4809
F2	GOA2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	**27.0981**
	GOA3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	52.7359
	HCUGOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	29.2753
	GOA	1.99E+03	4.50E+03	7.02E+02	7.73E+02	53.5119
	GOA1	1.10E+03	2.50E+03	1.96E+01	3.82E+02	50.7517
F3	GOA2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	**26.4331**
	GOA3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	50.9500
	HCUGOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	28.4700
	GOA	2.05E+01	3.71E+01	7.02E+02	4.00E+00	52.3205
	GOA1	4.55E+00	1.23E+01	1.96E+01	2.12E+00	50.5881
F4	GOA2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	**25.9882**
	GOA3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	50.4795
	HCUGOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	28.0496
	GOA	1.71E+04	1.17E+05	7.02E+02	3.00E+04	52.6925
	GOA1	2.03E+02	2.46E+03	1.96E+01	7.03E+02	50.0991
F5	GOA2	2.89E+01	2.90E+01	0.00E+00	**2.12E-02**	**26.0284**
	GOA3	2.84E+01	2.87E+01	0.00E+00	5.04E-02	50.3249
	HCUGOA	**1.78E-03**	**2.12E+01**	**0.00E+00**	7.26E+00	27.7006
	GOA	3.00E+00	5.68E+00	7.02E+02	7.06E-01	52.1765
	GOA1	1.79E-02	3.99E-02	1.96E+01	6.50E-03	50.9860
F6	GOA2	2.83E-06	8.96E-05	0.00E+00	2.62E-05	**26.1002**
	GOA3	9.64E-07	3.29E-05	0.00E+00	8.63E-06	50.4322
	HCUGOA	**4.96E-07**	**2.31E-05**	**0.00E+00**	**6.64E-06**	27.9789
	GOA	-6.74E+03	-5.11E+03	7.02E+02	4.39E+02	52.8227
	GOA1	-8.38E+03	-7.58E+03	1.96E+01	2.23E+02	50.7339
F7	GOA2	-1.08E+04	-6.71E+03	0.00E+00	9.37E+02	**26.0998**
	GOA3	-1.84E+86	-2.74E+50	0.00E+00	3.35E+85	50.6104
	HCUGOA	**-9.99E+122**	**-3.44E+92**	**0.00E+00**	**1.85E+122**	28.1537
	GOA	9.83E+01	1.87E+02	7.02E+02	2.19E+01	52.8676
	GOA1	3.58E+01	9.23E+01	1.96E+01	1.53E+01	51.1436
F8	GOA2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	**26.4792**
	GOA3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	50.9370
	HCUGOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	28.1102
	GOA	1.16E+00	3.40E+00	7.02E+02	4.88E-01	53.0531
	GOA1	7.99E-15	1.16E+00	1.96E+01	5.32E-01	50.9831
F9	GOA2	8.88E-16	8.88E-16	0.00E+00	1.00E-31	**26.2314**
	GOA3	8.88E-16	8.88E-16	0.00E+00	1.00E-31	50.8447
	HCUGOA	**8.88E-16**	**8.88E-16**	**0.00E+00**	**1.00E-31**	28.0339
	GOA	1.67E+00	9.61E+00	7.02E+02	2.27E+00	53.0192
	GOA1	8.86E-01	1.10E+00	1.96E+01	4.40E-02	50.9324
F10	GOA2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	**26.2385**
	GOA3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	50.8215
	HCUGOA	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	28.2026
	GOA	6.11E+00	1.73E+01	7.02E+02	3.44E+00	43.1112
	GOA1	2.36E+00	7.70E+00	1.96E+01	1.63E+00	41.5197
F11	GOA2	3.80E-01	9.97E-01	0.00E+00	1.55E-01	**21.2062**
	GOA3	9.36E-05	**3.51E-04**	0.00E+00	**6.16E-05**	41.8243
	HCUGOA	**2.66E-06**	4.50E-03	**0.00E+00**	1.51E-03	23.8091

为从多样性的角度分析改进策略的有效性, 图 2 给出了 GOA 和改进策略算法以及 HCUGOA 的部分函数独立运行 30 次的平均收敛曲线分布图, 各算法图例如图 2(a) 所示. 从图 2 可明显看出, 无论是对单峰函数还是多峰函数,HCUGOA 都具有更高的收敛速度和收敛精度. 具体来说, 图 2(a)~(e) 为单峰函数平均收敛曲线, 从图中可看出 GOA2 和 GOA3 具有更高的收敛精度和更快的寻优速度,GOA1 是 3 个改进策略里寻优效果较差的, 但其寻优精度也比 GOA 的高; 图 (f)~(h) 为多峰函数平均收敛曲线, 从图可知,3 种改进策略算法的平均收敛曲线均位于 GOA 收敛曲线下方, 且下降速度快, 解的分布范围广, 使改进策略算法在相同的迭代次数下具有更高的收敛精度, 在相同的收敛精度下具有更快的收敛速度.

综上所述, 分析表 3 和图 2 的结论一致证明了三种改进策略的有效性.

表 4　与四种群智能算法在不同维度下的结果比较

函数	算法	10dim			30dim			200dim		
		平均值	标准差	耗时/s	平均值	标准差	耗时/s	平均值	标准差	耗时/s
F1	MS-WOA[5]	0.00E+00	0.00E+00	——	0.00E+00	0.00E+00	——	0.00E+00	0.00E+00	——
	m-SCA[8]	——	——	——	5.70E-03	2.63E-02	——	——	——	——
	IGOA	5.23E-17	5.76E-11	26.1423	1.68E-17	3.28E-17	55.7765	2.18E+01	8.04E+02	130.917
	MGOA	2.05E-92	4.85E-77	25.6246	1.06E-12	2.15E-12	55.3427	4.48E-84	3.10E-74	129.391
	HCUGOA	**0.00E+00**	**0.00E+00**	**14.7679**	**0.00E+00**	**0.00E+00**	**28.7335**	**0.00E+00**	**0.00E+00**	**69.2165**
F2	MS-WOA[5]	0.00E+00	0.00E+00	——	1.31E-180	0.00E+00	——	6.59E-175	0.00E+00	——
	m-SCA[8]	——	——	——	9.11E-04	1.90E-03	——	——	——	——
	IGOA	1.20E-13	2.21E-09	26.6332	5.32E-10	6.84E-10	55.8266	7.46E-03	6.05E-01	139.7579
	MGOA	5.66E-58	1.69E-52	25.9628	2.76E-07	2.80E-07	55.2711	1.34E-56	2.39E-49	137.9058
	HCUGOA	**0.00E+00**	**0.00E+00**	**14.6698**	**0.00E+00**	**0.00E+00**	**29.2753**	**0.00E+00**	**0.00E+00**	**73.8671**
F3	MS-WOA[5]	0.00E+00	0.00E+00	——	5.63E-322	0.00E+00	——	4.74E-312	0.00E+00	——
	m-SCA[8]	——	——	——	8.48E+02	5.49E+02	——	——	——	——
	IGOA	5.84E-08	7.97E-02	27.3995	1.98E-16	5.11E-16	45.8111	1.97E+04	1.52E+04	142.4474
	MGOA	3.18E-02	3.20E+02	26.2447	1.80E-12	3.66E-12	45.1329	1.31E+05	4.18E+04	140.4843
	HCUGOA	**0.00E+00**	**0.00E+00**	**15.0764**	**0.00E+00**	**0.00E+00**	**28.4700**	**0.00E+00**	**0.00E+00**	**75.8143**
F4	MS-WOA[5]	0.00E+00	0.00E+00	——	9.64E-170	0.00E+00	——	1.12E-166	0.00E+00	——
	m-SCA[8]	——	——	——	7.07E-01	5.37E-01	——	——	——	——
	IGOA	1.60E-06	1.02E-03	16.9476	2.85E-09	3.53E-09	45.2030	5.66E+01	6.59E+00	140.0804
	MGOA	2.02E-01	8.53E+00	16.8291	5.88E-07	8.29E-07	44.1018	1.20E-03	2.82E+01	138.7262
	HCUGOA	**0.00E+00**	**0.00E+00**	**11.7371**	**0.00E+00**	**0.00E+00**	**28.0496**	**0.00E+00**	**0.00E+00**	**73.3962**
F5	MS-WOA[5]	——	——	——	2.73E+01	3.64E-01	——	——	——	——
	m-SCA[8]	——	——	——	2.95E+01	2.43E+00	——	——	——	——
	IGOA	6.78E+00	4.39E-01	17.6294	7.57E+00	3.33E-01	47.2460	2.09E+05	7.93E+06	138.2211
	MGOA	5.86E+00	4.16E-01	16.4653	6.82E+00	1.31E+00	47.9160	4.74E+01	3.56E-01	139.0233
	HCUGOA	**2.04E-02**	**1.97E+00**	**14.6491**	**1.78E-03**	**7.26E+00**	**27.7006**	**9.71E-04**	**8.65E+00**	**75.5475**
F6	MS-WOA[5]	4.77E-05	512E-05	——	1.05E-04	7.57E-05	——	1.58E-04	1.44E-04	——
	m-SCA[8]	——	——	——	1.95E-02	6.87E-03	——	——	——	——
	IGOA	3.08E-04	1.62E-03	18.1426	9.06E-05	9.35E-05	45.9090	4.39E-01	4.36E+00	141.5622
	MGOA	6.94E-06	2.95E-03	18.9377	9.16E-05	9.14E-05	45.6239	1.06E-04	4.56E-03	139.5832
	HCUGOA	**5.49E-08**	**9.74E-06**	**14.6742**	**4.96E-07**	**6.64E-06**	**28.0339**	**3.42E-07**	**7.22E-06**	**73.3773**
F7	MS-WOA[5]	——	——	——	-1.26E+04	1.55E+01	——	——	——	——
	m-SCA[8]	——	——	——	-4.26E+02	2.88E+02	——	——	——	——
	IGOA	-2.48E+03	1.63E+02	26.8037	-2.31E+03	2.17E+02	51.5862	-6.11E+03	4.14E+02	136.1719
	MGOA	-4.19E+03	6.35E+02	26.6435	-7.98E+03	2.23E+03	51.5585	-2.09E+04	3.00E+03	137.3575
	HCUGOA	**-1.12E+128**	**2.24E+127**	**14.8113**	**-9.99E+122**	**1.85E+122**	**27.9789**	**-5.76E+125**	**1.05E+125**	**73.6172**
F8	MS-WOA[5]	0.00E+00	0.00E+00	——	0.00E+00	0.00E+00	——	0.00E+00	0.00E+00	——
	m-SCA[8]	——	——	——	7.81E+01	5.21E+01	——	——	——	——
	IGOA	0.00E+00	5.20E+00	18.0718	0.00E+00	0.00E+00	52.8868	6.23E+00	6.10E+01	139.8944
	MGOA	0.00E+00	0.00E+00	19.0422	8.57E-14	1.08E-13	52.3478	0.00E+00	0.00E+00	138.3891
	HCUGOA	**0.00E+00**	**0.00E+00**	**14.8238**	**0.00E+00**	**0.00E+00**	**28.1102**	**0.00E+00**	**0.00E+00**	**73.4033**
F9	MS-WOA[5]	8.88E-16	0.00E+00	——	8.88E-16	0.00E+00	——	8.88E-16	0.00E+00	——
	m-SCA[8]	——	——	——	——	——	——	——	——	——
	IGOA	1.76E-10	1.21E+00	26.6218	2.19E-09	4.83E-09	47.8905	5.88E-04	9.18E+00	141.2315
	MGOA	8.88E-16	2.65E-15	26.2053	2.94E-07	2.09E-07	47.4453	8.88E-16	2.87E-15	139.5626
	HCUGOA	**8.88E-16**	**1.00E-31**	**16.4119**	**8.88E-16**	**1.00E-31**	**28.0339**	**8.88E-16**	**1.00E-31**	**74.7682**
F10	MS-WOA[5]	0.00E+00	0.00E+00	——	0.00E+00	0.00E+00	——	0.00E+00	0.00E+00	——
	m-SCA[8]	——	——	——	3.84E-02	7.15E-02	——	——	——	——
	IGOA	1.45E-12	9.50E-02	27.9889	0.00E+00	0.00E+00	56.1211	1.01E+00	5.34E+00	140.6504
	MGOA	0.00E+00	2.03E-01	26.3883	1.12E-11	2.22E-11	55.4097	0.00E+00	5.19E-02	138.6253
	HCUGOA	**0.00E+00**	**0.00E+00**	**14.8834**	**0.00E+00**	**0.00E+00**	**28.2026**	**0.00E+00**	**0.00E+00**	**73.9957**
F11	MS-WOA[5]	——	——	——	1.15E-02	5.43E-02	——	5.84E+00	4.29E+00	——
	m-SCA[8]	——	——	——	1.45E-01	8.19E-02	——	——	——	——
	IGOA	4.95E-02	5.41E-02	26.9681	7.05E-03	2.10E-03	47.0489	5.99E+00	1.48E+07	102.6983
	MGOA	2.41E-03	3.06E-01	26.1668	9.99E-06	4.65E-06	46.7195	1.25E-02	1.25E-02	101.2205
	HCUGOA	**1.73E-06**	**6.52E-06**	**15.1757**	**2.66E-06**	**1.51E-03**	**23.8091**	**3.57E-07**	**7.00E-04**	**55.0631**

3.3 与最新的群智能算法比较

将混合策略改进鲸鱼算法 MS-WOA[5]、正弦余弦改进算法 m-SCA[8] 和最新的蝗虫改进算法 IGOA[14] 及 MGOA[15] 与 HCUGOA 进行对比, 分别在空间维度为 10 维、30 维、200 维的情况下对 12 个测试函数进行寻优 (“——”表示参考文献未给出相应数据), 独立运行 30 次后结果如表 4 和表 5 所示.

表 5 与四种群智能算法在固定低维下的结果比较 (dim=4)

函数	算法	平均值	标准差	耗时/s
F12	MS-WOA[5]	4.06E-04	1.05E-04	——
	m-SCA[8]	5.10E-04	1.00E-04	——
	IGOA	3.21E-04	5.19E-06	10.8759
	MGOA	3.08E-04	6.07E-07	10.2513
	HCUGOA	**3.07E-04**	**5.86E-07**	**7.2332**

由表 4 和表 5 可知, 总体上 m-SCA 寻优结果相对较差,IGOA 和 MGOA 寻优精度相差不大,MS-WOA 寻优精度是 4 种对比算法中最高的, 但 HCUGOA 收敛精度和稳定性明显优于 4 种对比算法且耗时最短.

纵向观察数据, MS-WOA 只在对 F1、F8 和 F10 求解时, 才寻到理论值, 但由标准差可知 MS-WOA 寻到理论值不具稳定性. 对于 F5~F7 和 F9~F12, 在 4 种对比算法求解精度低或无法求解情况下,HCUGOA 仍具较高寻优精度和稳定性. 其中 MGOA 寻优结果是 3 种算法中较差的,IGOA 次之, 且 IGOA 在对 F8 和 F10 求解时, 寻到理论最优值 0,HCUGOA 对 12 个测试函数均取得较好结果, 尤其是 F1~F4、F8 和 F10, 寻到理论值. 横向观察数据, 当维度从 10 维到 30 维再到 200 维, 算法对求解精度和鲁棒性均有所下降, 这是因维度增加使函数更复杂, 寻优时需做更多调整. 但相对 4 种对比算法,HCUGOA 求解精度仍最高, 从而验证了 HCUGOA 在低维和高维情况下均有更好的求解能力. 另外, 从平均耗时来看, 相同维度下,HCUGOA 相对于其他两种 GOA 算法的平均耗时最短, 不同维度下, 随维度的增加,3 种算法的平均运行时间都变长, 这是因从低维到高维, 搜索空间增大, 寻优难度也呈指数增加.

3.4 Wilcoxon 秩和检验

上述仿真中, 均未对每次结果进行独立比较. 为体现算法的稳定性和公平性, 本文采用 Wilcoxon 秩和检验来验证本文改进算法每次运行结果是否在统计上与其他算法存在显著性差异. 秩和检验在 5% 的显著性水平下进行,p 值小于 5% 时, 拒绝 H0 假设, 说明两种对比算法具显著性差异; 否则接受 H0 假设, 说明两种算法寻优结果在整体上是相同的[18].

表 6 给出了测试函数下 GOA、IGOA、MGOA、GOA1、GOA2 和 GOA3 共 6 种算法与 HCUGOA 的秩和检验中计算的 p 值, 因最佳算法不能与自身进行比较, 所以表中标记为“Na”表不适用, 即无法进行显著性判断, “R”为显著性判断结果, “+”, “-”, “=”分别表 HCUGOA 性能优于, 劣于和相当于对比算法. 从表 6 可看出大部分 p 值远小于 5%, “R”为“+”, 拒绝零假设. 因此总体上 HCUGOA 与其他 6 种算法之间具显著性差异, 且 HCUGOA 显著更优.

3.5 在 CEC2014 基准函数上进行测试

为了进一步验证 HCUGOA 的有效性和鲁棒性, 本文在 CEC2014 单目标优化函数中选取部分单峰、多峰、混合和复合类型的函数进行优化求解, 对选取的部分函数的总结如表 7 所示, 本实验种群规模为 30, 最大迭代次数为 1000, 维度 dim= 30.

表 6 Wilcoxon 秩和检验 p 值

函数	GOA		IGOA		MGOA		GOA1		GOA2		GOA3	
	P	R	P	R	P	R	P	R	P	R	P	R
F1	1.21E-12	+	1.21E-12	+	1.21E-12	+	1.21E-12	+	Na	=	Na	=
F2	1.21E-12	+	1.21E-12	+	1.21E-12	+	1.21E-12	+	Na	=	Na	=
F3	1.21E-12	+	1.21E-12	+	1.21E-12	+	1.21E-12	+	Na	=	Na	=
F4	1.21E-12	+	1.21E-12	+	1.21E-12	+	1.21E-12	+	Na	=	Na	=
F5	3.02E-11	+	3.02E-11	+	3.02E-11	+	3.02E-11	+	3.02E-11	+	3.02E-11	+
F6	3.02E-11	+	3.02E-11	+	3.02E-11	+	3.02E-11	+	8.29E-06	+	1.15E-01	-
F7	3.02E-11	+	3.02E-11	+	3.02E-11	+	3.02E-11	+	3.02E-11	+	3.02E-11	+
F8	1.21E-12	+	1.61E-01	-	1.21E-12	+	1.21E-12	+	Na	=	Na	=
F9	1.21E-12	+	1.16E-08	+	1.21E-12	+	1.16E-12	+	Na	=	Na	=
F10	1.21E-12	+	1.61E-01	-	1.21E-12	+	1.21E-12	+	Na	=	Na	=
F11	3.02E-11	+	3.02E-11	+	3.02E-11	+	3.02E-11	+	3.02E-11	+	9.21E-05	+

表 7 CEC2014 函数 (部分)

函数	维度	特征	定义域	最佳值
CEC01	30	UN	[-100,100]	100
CEC05	30	MN	[-100,100]	500
CEC17	30	HF	[-100,100]	1700
CEC18	30	HF	[-100,100]	1800
CEC24	30	CF	[-100,100]	2400
CEC25	30	CF	[-100,100]	2500

表 8 CEC2014 优化结果比较

算法	指标	CEC01	CEC05	CEC17	CEC18	CEC24	CEC25
HCUGOA	Mean	1.2142E+06	5.2088E+02	5.2088E+02	2.8480E+03	2.6002E+03	2.7071E+03
	Std	4.9898E+05	5.2021E+02	5.2021E+02	3.7464E+02	0.0000E+00	0.0000E+00
GOA1	Mean	5.8819E+08	5.2284E+02	5.2284E+02	9.9000E+05	2.1325E+03	2.7426E+03
	Std	8.3437E+07	5.2192E+02	5.2192E+02	6.8374E+04	6.2000E+00	2.6301E+00
GOA2	Mean	3.5022E+06	5.2084E+02	5.2084E+02	8.1304E+03	2.6156E+03	2.7201E+03
	Std	6.9549E+05	5.2092E+02	5.2092E+02	3.7476E+03	0.0000E+00	0.0000E+00
GOA3	Mean	2.2166E+07	5.2105E+02	5.2105E+02	9.1320E+04	2.4159E+03	2.6085E+03
	Std	2.5377E+06	5.2089E+02	5.2089E+02	7.0098E+03	7.6160E+00	0.0000E+00
IGOA	Mean	4.7150E+08	5.2234E+02	5.2234E+02	2.0705E+08	2.9100E+03	2.7421E+03
	Std	3.8784E+08	5.2183E+02	5.2183E+02	4.3194E+08	6.6156E+00	5.6245E+01
MGOA	Mean	7.8319E+06	5.2101E+02	5.2101E+02	3.0360E+07	2.5231E+03	2.7260E+03
	Std	5.6314E+06	5.2069E+02	5.2069E+02	2.1886E+07	4.6162E+00	2.6404E+00
PSO[4]	Mean	6.69E+06	2.09E+01	2.09E+01	3.97E+03	2.30E+02	2.09E+02
	Std	8.99E+06	8.52E-02	8.52E-02	4.62E+03	6.64E+00	1.64E+00
L-SHADE[19]	Mean	1.42E-14	2.01E+01	2.01E+01	6.58E+00	2.24E+02	2.03E+02
	Std	3.61E-15	1.70E-02	1.70E-02	2.74E+00	1.15E+00	4.97E+02

CEC2014 函数因具有复杂的特征, 常用来验证算法的鲁棒性. 将本文算法与 3 个改进策略算法、2 个 GOA 变体以及标准 PSO[4] 和典型 DE 算法进行比较, 其中因 L-SHADE[19] 在 CEC2014 函数中的出色表现, 常作为比较算法, 另外,PSO 和 L-SHADE 已在参考文献中对 CEC2014 函数进行了测试.CEC2014 中部分函数独立运行 30 次后的结果如表 8 所示.

由表 8 可知, 在单峰函数上,L-SHADE 具有明显优势, 而 PSO 和 HCUGOA 性能相似, 例如对于 CEC01,L-SHADE 寻优精度是-14, 而 HCUGOA 和其他 GOA 变体的最优精度是 +05; 在多峰问题上,6 个 GOA 变体算法表现非常相似, 在混合和复合函数中, 除 L-SHADE 之外,HCUGOA 的寻优结果比对比算法更接近理论值, 且比 PSO 更稳定, 验证了 HCUGOA 具有良好的有效性和稳定性. 此外, 从实验数据可以看出 L-SHADE 与 GOA 变体相比具有明显的优势,HCUGOA 以及其他 GOA 算法均无法达到 L-SHADE 的水平, 这可能是因为差分进化算法本身在寻优过程中具有比 GOA 更具优势的原因.

4 结 论

为改善 GOA 缺陷, 本文受柯西变异和 PSO 的启发, 提出新型的位置更新方式以增加种群多样性, 增强全局探索能力; 在柯西变异和反向学习的基础上, 对目标值进行变异更新, 使算法避免陷入局部最优; 为了更好的平衡全局探索和局部开发, 将均匀分布函数引入到控制参数 c, 构建新函数. 五组实验显示 HCUGOA 对基准测试函数的求解结果要优于 MS-WOA、m-SCA 等群智能算法, 与新改进的蝗虫算法 IGOA 和 MGOA 相比也具有竞争性;Wilcoxon 秩和检验证明了本文算法具有更优的显著性差异; 通过对 CEC2014 部分函数求解结果也表明了本文算法的有效性. 在后续研究中, 考虑将改进的蝗虫优化算法应用到 WSN 路由能耗寻优问题中, 找到能耗最小的广播路径, 以进一步验证算法的性能.

参考文献 (References)

[1] Saremi S, Mirjalili S, Lewis A. Grasshopper optimization algorithm: Theory and Application [J]. Advances in Engineering Software, 2017, 105: 30-47.

[2] KapilevichViacheslav, Seno Shigeto, Matsuda Hideo,

Takenaka Yoichi. Chromatin 3D reconstruction from chromosomal contacts using a genetic algorithm[J]. IEEE/ACM transactions on computational biology and bioinformatics,2018.

[3] A. Jayaprakash,C. KeziSelvaVijila. Feature selection using ant colony optimization (ACO) and Road Sign Detection and Recognition (RSDR) system[J]. Cognitive Systems Research, 2019, 58.

[4] 徐桂萍. 基于维度学习策略的粒子群算法的研究与应用 [D]. 吉林大学:2019.
(Xu G P. Study on particle swarm optimization based with dimensional learning strategy[D]. Jilin university: 2019.)

[5] 何庆, 魏康园, 徐钦帅. 基于混合策略改进的鲸鱼优化算法 [J]. 计算机应用研究, 2019, 36(12): 3647-3651+3665.
(He Q, Wei K Y, Xu Q S. Mixed strategy based improved whale optimization algorithm [J]. Application Research of Computers, 2019,36(12):3647-3651+3665)

[6] 徐钦帅, 何庆, 魏康园. 改进蚁狮算法的无线传感器网络覆盖优化 [J]. 传感技术学报, 2019, 32(02): 266-275.
(Xu Q S, He Q, Wei K Y. Wireless sensor network coverage optimization based on improved ant lion algorithm[J]. Journal of Transduction Technology, 2019, 32(02): 266-275.)

[7] 张达敏, 陈忠云, 辛梓芸, et al. 基于疯狂自适应的樽海鞘群算法 [J/OL]. 控制与决策: 1-9[2019-10-23]. https://doi.org/10.13195/j.kzyjc.2019.0012.
(Zhang D M, Cheng Z Y, Xin Z Y, et al. Salpswarm algorithm based on craziness and adaptive [J/OL]. Control and Decision: 1-9[2019-10-23]. https://doi.org/10.13195/j.kzyjc.2019.0012.)

[8] Gupta S, Deep K. A hybrid self-adaptive sine cosine algorithm with opposition based learning, Expert Systems with Applications, 2019, 119: 210-230.

[9] 闫旭, 叶春明. 混合蝗虫优化算法求解作业车间调度问题 [J]. 计算机工程与应用, 2019, 55(06): 257-264.
(Yan X, Ye C M. Hybrid grasshopper optimization algorithm for job shop scheduling problem[J]. Computer Engineering and Applications, 2019, 55(06): 257-264.)

[10] 黄超, 梁圣涛, 张毅, et al. 基于多目标蝗虫优化算法的移动机器人路径规划 [J/OL]. 计算机应用: 1-7[2019-10-23].http://kns.cnki.net/kcms/detail/51.1307.TP.20190819.0948.068.html.
(Huang C, Liang S T, Zhang Y, et al. Mobile robot path planning based on multi-target grasshopper optimi- zation algorithm [J/OL]. Computer Application: 1-7[2019-10-23].http://kns.cnki.net/kcms/detail/51.1307.TP.20190819.0948.068.html.)

[11] 崔鹤, 薛媛. 基于 GOA-SVR 的在线英语学习人数预测 [J]. 微型电脑应用, 2019, 35(05): 128-131.
(Cui H, Xue Y. Prediction of online English learning population based on GOA-SVR[J]. Microcomputer Applications, 2019, 35(05): 128-131.)

[12] 潘峰, 孙红霞. 基于蝗虫算法的图像多阈值分割方法 [J]. 电子测量与仪器学报, 2019, 33(01): 149-155.
(Pan F, Sun H X. Image multi-threshold segmentation method based on grasshopper optimization algorithm[J]. Journal of Electronic Measurement and Instrument, 2019, 33(01): 149-155.)

[13] Zhao R, Ni H, Feng H, et al. A dynamic weight grasshopper optimization algorithm with random jumping[J]. Advances in Computer Communication and Computational Sciences. Advances in Intelligent Systems and Computing, 2019: 401-413.

[14] Luo J, Chen H, Zhang Q, et al. An improved grasshopper optimization algorithm with application to financial stress prediction[J], Applied Mathematical Modelling, 2018(64): 654-668

[15] KamelM A, Jurado S, Ebeed F, et al. Modified grasshopper optimization framework for optimal power flow solution[J], Electrical Engineering, 2019: 121-148.

[16] 李洋州, 顾磊. 一种基于曲线自适应和模拟退火的蝗虫优化算法 [J/OL]. 计算机应用研究: 1-9[2019-10-25]. https://doi.org/10.19734/j.issn.1001-3695.2018.07.0580.
(Li Y Z, Gu L. A grasshopper optimization algorithm based on curve adaptive and simulated annealing[J/OL]. Computer Application Research: 1-9[2019-10-25]. https://doi.org/10.19734/j.issn.1001-36 95.2018.07.058.)

[17] Tizhoosh H R. Opposition-based learning: a new scheme for machine intelligence [C]// Proc of Computational Intelligence for Modelling. Vienna, Austria: Computer Society, 2005, 1: 695-701.

[18] 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.

[19] Tanabe R, Fukunaga A S. Improving the search performance of SHADE using linear population size reduction[C]//2014 IEEE Congress on Evolutionary Computation (CEC). Beijing, China: IEEE, 2014: 1658-1665.

作者简介

何庆 (1982−), 男, 副教授, 博士, 主要研究方向为大数据应用, 进化计算, E-mail: qhe@gzu.edu.cn;

林杰 (1995−), 女, 硕士研究生, 主要研究方向为进化计算, 数据挖掘, E-mail: 1376135318@qq.com.

徐航 (1995−), 男, 硕士研究生, 主要研究方向为认知无线电, E-mail: 1518818139@qq.com.