求解复杂约束优化问题的多策略混合麻雀搜索算法

刘耿耿[1], 张丽媛[1], 刘 笛[2], 刘能现[1], 傅仰耿[1], 郭文忠[1], 陈国龙[1], 蒋伟进[3†]

(1. 福州大学 计算机与大数据学院，福州 350116；2. 华中科技大学 电子信息与通信学院，武汉 430000；
3. 湖南工商大学 计算机与信息工程学院, 长沙 410205)

摘 要: 针对麻雀搜索算法面对具有强约束、非凸性和不可微特征的复杂问题所存在的开发与探索能力不平衡、易陷入局部最优、过早收敛和种群多样性较低等不足, 提出一种求解复杂约束优化问题的多策略混合麻雀搜索算法. 首先利用反向学习策略构建双向初始化机制, 以达到获得分布更优的初始种群的目的. 其次设计了一种基于交叉与变异算子的位置更新公式, 扩大搜索范围, 丰富搜索机制, 来平衡算法探索和开发能力, 同时提高算法的收敛精度和速度. 最后使用社区学习策略对种群进行精炼, 强化开发能力与跳出局部极值的能力, 并保持种群的多样性. 本文算法分别在 23 个非约束优化基准测试函数、CEC2017 的 28 个实数约束优化问题和 4 个工程优化问题上进行了性能评估, 实验结果表明, 所提出的算法对比其他优化算法具有寻优能力强, 收敛精度高, 收敛速度快等优势, 可有效解决复杂约束优化问题.

关键词: 麻雀搜索算法；约束优化问题；多策略混合；测试函数；CEC2017；工程优化

中图分类号: TP301.6 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2022.0321

引用格式 : 刘耿耿，张丽媛，刘笛，等. 求解复杂约束优化问题的多策略混合麻雀搜索算法 [J] . 控制与决策.

Multi-strategy Hybrid Sparrow Search Algorithm for Complex Constrained Optimization Problems

LIU Geng-geng[1], *ZHANG Li-yuan*[1], *LIU Di*[2], *LIU Neng-xian*[1], *FU Yang-geng*[1], *GUO Wen-zhong*[1], *CHENG Guo-long*[1] *JIANG Wei-jin*[3†]

(1.School of Computer and Big Data, Fuzhou University, Fuzhou 350116, China; 2.School of Electronic Information and Communication, Huazhong University of Science and Technology, Wuhan 430000, China; 3.College of Computer and Information Engineering, Hunan University of Technology and Business,Changsha 410205, China)

Abstract: In view of the shortcomings of the sparrow search algorithm in the face of complex problems with strong constraints、non-convexity and non-differentiability, such as unbalanced exploitation and exploration ability、easy to fall into local optimum、premature convergence and low population diversity, a multi-strategy hybrid sparrow search algorithm for complex constrained optimization problems is proposed. Firstly, uses the opposition-based learning strategy to construct a bi-directional initialization mechanism to achieve the purpose of obtaining the initial population with better distribution. Secondly, a position update formula based on crossover and mutation operator is designed to expand the search range and enrich the search mechanism for balancing the exploration and exploitation ability of the algorithm, while improving the convergence accuracy and speed of the algorithm. Finally, the community learning strategy is used to refine the population, strengthen the exploitation ability and the ability to jump out of the local optima, and maintain the diversity of the population. The performance of the proposed algorithm is evaluated on 23 benchmark functions of unconstrained optimization, 28 real constrained optimization problems of CEC2017 and 4 engineering optimization problems. The experimental results show that the proposed algorithm compared with other optimization algorithms has advantages such as stronger optimization ability, higher convergence accuracy, faster convergence speed and so on, which can be used to effectively solve complex constrained optimization problems.

Keywords: sparrow search algorithm；constrained optimization problem；multi-strategy hybrid；benchmark function；CEC2017；engineering optimization

收稿日期: 2022-03-02；**录用日期:** 2022-07-16.

基金项目: 国家自然科学基金 (61877010, 11501114)、国家重点基础研究发展计划 (2011CB808000)、计算机体系结构国家重点实验室开放课题 (CARCHB202014)、福建省自然科学基金 (2019J01243).

†通讯作者. E-mail: jwjnudt@163.com.

0 引 言

约束优化问题 (Constrained Optimization Problem, COPs) 是优化领域中的一个重要问题, 广泛存在于工程实际且求解困难, 其研究具有重要的理论意义和工程价值. 其本质上是一个寻求决策变量 X 的过程, 使 X 能够最大化或最小化目标函数的同时满足对搜索空间的线性和非线性约束[1]. 约束一般可以分为等式约束和不等式约束, 分别以等式和不等式方程的形式表示, X 需要同时满足这两类约束. 故 COPs 可以定义为公式 (1):

$$\begin{aligned}&\min/\max: f(X)\\&\text{s.t}:\begin{cases}g_i(X)\le 0, i=1,2,...,l\\h_j(X)=0, j=l+1,l+2,...,m\end{cases}\end{aligned}\tag{1}$$

其中 $f(X)$ 是目标函数;$X=(x_1,x_2,...,x_D)\in S, L_i\le x_i\le U_i$; D 是 X 的维度; L_i 和 U_i 分别是 x_i 的下界和上界; $S=\Pi_i^D[L_i,U_i]$ 是搜索空间; m 是总约束数, 包括不等式和相等式约束; $g_i(X)$ 和 $h_j(X)$ 分别表示第 i 个不等式约束和第 (j-l) 个等式约束. 寻求优化问题的最优解是一项极具挑战性的任务, 这要求根据问题类型、计算成本和算法特性正确选择和使用算法. 相较于大多基于梯度下降的传统优化算法, 全局优化的启发式算法具有实现简单、鲁棒性强、搜索能力强和扩展性强的特点.

近年来, 元启发式算法得到了极大的发展. 代表性的算法有: 基于自然选择的遗传算法 (Genetic Algorithm, GA)[2], 全局搜索能力强, 但需要设置的参数较多且局部搜索能力较弱; 基于群体差异的差分进化 (Differential Evolution, DE) 算法[3] 具有非常优秀的寻优能力, 但存在收缩停滞和过早收敛问题; 受鸟群捕食行为启发的粒子群优化 (Particle Swarm Optimization, PSO)[4] 算法, 搜索速度快, 但迭代后期搜索能力较弱, 易陷入局部最优, 过早收敛; 灰狼优化 (Grey Wolf Optimization, GWO) 算法[5], 参数少, 实现简单, 搜索能力较强, 但种群多样性差; 鲸鱼优化算法 (Whale Optimization Algorithm, WOA)[6], 搜索能力较强, 扩展性高, 但收敛速度慢, 易陷入局部解; 正余弦算法 (Sine and Cosine Algorithm, SCA)[7] 是受数学模型启发的基于种群优化算法, 实现简单, 但优化精度不高; 受物理法则启发的阿基米德算法 (Archimedes Optimization Algorithm, AOA)[8], 具有较强的搜索能力, 但跳出局部最优能力不足; 还有基于库仑斯静电力定律和牛顿运动定律的人工电场算法 (ArtificialElectric Field Algorithm, AEFA)[9] 等等. 可以看到, 每个算法都有各自的优缺点, 适用于解决不同类型的问题, 正如 “无免费午餐定理” [10] 所指出的, 单一算法很难完美解决所有问题.

麻雀搜索算法 (Sparrow Search Algorithm, SSA)[11] 是 2020 年提出的一种受麻雀觅食和反捕食行为启发的群智能优化算法. SSA 具有较强的搜索能力, 且实现简单、参数少、可扩展性高, 已成功应用于医学科研、机器学习和工程实践等领域. 然而, 当解决复杂约束问题时, SSA 算法存在易陷入局部最优, 过早收敛, 求解精度低等不足. 本文针对 SSA 的不足, 提出了多策略混合麻雀搜索算法 (Multi-Strategy Hybrid SSA, MSHSSA). 首先, 利用反向学习策略[12] 构建双向初始化机制, 以获得最优的初始种群 X_{init}; 其次, 群智能优化算法搜索性能的优劣很大程度上依赖于探索和开发, 故通过结合交叉和变异算子, 提出了一个新的位置更新方式来生成新的种群 X_1, 有效扩展了探索和开发能力的同时提高收敛精度和速度; 最后, 采用社区学习策略实现对更新后种群 X_1 的精炼, 有利于在有希望区域挖掘更优解, 提高对局部极值的判断力, 增加种群的多样性.

1 相关工作

求解约束优化进化算法的性能取决于两个因素: 一是作为搜索引擎的进化算法, 二是用来处理约束的约束处理方法. 文献 [13] 将约束处理方法划分为: 1)、可行性法则. 2)、随机排序法. 3)、ε 约束处理法. 4)、惩罚函数法. 5)、多目标优化法. 6)、混合法. 其中可行性法则实现简单, 无需定义额外参数, 具有高普遍性 (适合与不同的算法结合, 而不去过多地改变算法本身), 得到了广泛的应用[13]. 故本文采用 MSHSSA 与可行性法则进行研究.

1.1 麻雀搜索算法研究现状

近年来, 许多研究者对麻雀搜索算法进行改进并应用于实际问题. 初始化, 全局探索和局部开发是麻雀搜索算法寻优三大步骤. 麻雀搜索算法使用简单的随机初始化方法来生成初始的种群, 对于种群迭代的智能算法, 初始种群的质量对最终的收敛精度有一定的影响. 文献 [14] 采用重心反向学习机制初始化种群, 获得分布更优的初始种群. 文献 [15,16] 利用具有遍历性和初始敏感性的混沌映射生成初始种群, 来提高 SSA 的种群多样性与质量. 麻雀搜索算法具有强开发能力和较快的收敛速度, 在算法迭代后期, 种群逐渐聚集在当前最优位置周围, 若该位置是局部最优, 则难以跳出从而使解的精度降低. 一种有效的算法需要平衡全局探索能力和局部开发能力[14]. 为了扩大搜索范围, 提高发现者全局搜索能

力, 文献 [14]、[16] 分别将学习系数和自适应惯性权重引入到发现者位置更新公式中, 丰富发现者搜索机制. 为了保持种群多样性和避免算法过早收敛, 文献 [16] 利用改进边界约束处理策略, 将超出边界的个体随机分配到当前最优个体的附近, 而不是简单的分配为边界值, 这就避免了个体聚集在边界而导致的种群多样性下降的缺陷. 文献 [15] 在发现者更新公式中加入上一代的全局最优和动态权重因子 w, 避免算法陷入局部最优的同时提高收敛速度; 在精炼阶段中根据选择概率 ps 变换执行柯西突变策略和对立学习策略动态更新种群, 进一步提高算法的搜索能力和收敛速度. 虽然上述文献所提出的各种改进麻雀搜索算法的性能在一定程度上都有所提升, 但寻优能力、种群多样性、收敛速度与解集的精度仍有较大的提升空间.

1.2 可行性法则

可行性法则是将个体区域划分为可行解和不可行解区域, 可行解优于不可行解, 可行解根据适应度值的好坏选择, 不可行解根据约束违规度选择. 文献 [17] 在 DE 中的贪婪选择方法中引入可行性法则, 可以有效解决含有大量不等式约束的约束优化问题. 文献 [18] 通过 DE、替换和变异三个操作将目标函数信息整合到可行性法则中形成了 FROFI 方法, 该方法合理地平衡了目标函数与约束函数, 成功应用于约束机械设计优化问题和约束多目标优化问题. 文献 [19] 通过融入基于可行性法则的速度和位置边界策略提出了 AEFA-C, 提高了 AEFA 算法的开发和探索能力, 增强其对求解约束优化问题的有效性.

2 多策略混合麻雀搜索算法

2.1 算法的总体流程

与其他算法相比, 如 PSO 算法粒子更新方式是向最优解移动, 而 SSA 的麻雀是直接跳跃到当前最优解附近, 因此收敛速度较快, 局部搜索能力强, 但这也导致算法探索问题领域不充分, 种群多样性降低, 易陷入局部最优. 特别是当 SSA 解决的复杂约束问题具有强约束、非凸性、含有多个局部最优解等特征, 这会导致 SSA 极易陷入某一局部最优, 难以找到全局最优解. 为了改善上述问题, 本文提出了一种多策略混合麻雀搜索算法 (Multi-Strategy Hybrid Sparrow Search Algorithm, MSHSSA). 图 1 为 MSHSSA 求解 COPs 流程图. 首先, 利用反向学习策略构建双向初始化机制优化了初始种群的精度和分布性, 再根据可行性法则将种群分为可行解和不可行解后排序、划分. 其次, 采用基于交叉与变异算子位置更新公式来更新种群, 扩大探索区域, 平衡了开发和探索能力, 提升了收敛速度与精度. 最后, 引入社区学习策略进行精炼得到最终种群, 为下一代保留更优父代来提供搜索方向, 加强寻优能力, 避免陷入局部最优. 描述 MSHSSA 实现过程的伪代码见附录中的算法 A2. 关于原麻雀搜索算法详细介绍与分析见附录.

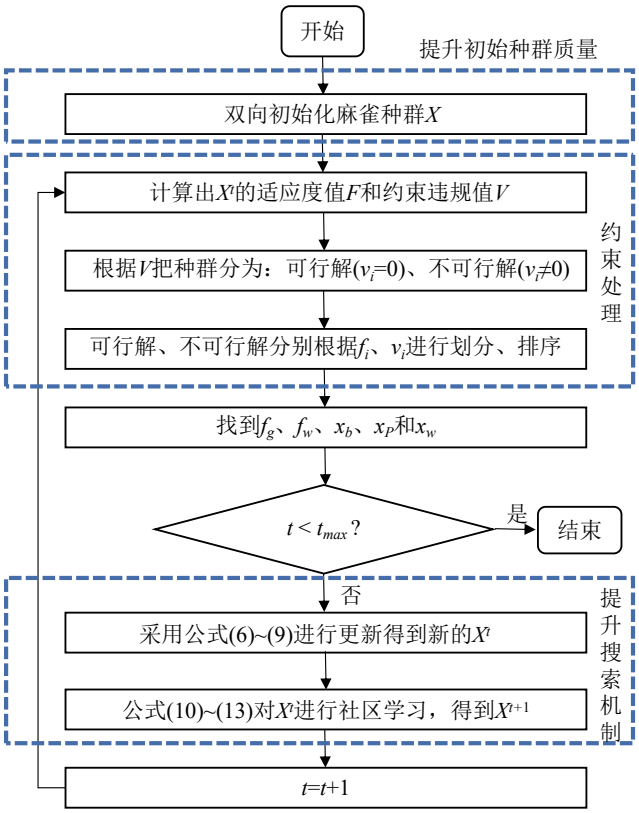

图 1 MSHSSA 流程图

2.2 双向初始化

群智能优化算法中初始种群的质量和多样性对算法的优化性能有很大的影响. 一个高质量的初始种群可以提高算法的收敛速度, 使算法有更多的机会找到全局最优解. SSA 通过简单随机初始化方式生成的初始种群存在种群多样性低、分布不均匀等不足. 反向学习 (Opposition-based Learning,OBL)[12] 作为一种优化技术, 通过提高初始解的质量和多样性, 已经成功地提升了许多元启发式算法性能. 本文结合反向学习策略构建双向初始化机制, 提高初始种群质量来加快并指导算法搜索全局最优解的同时保证种群多样性与均匀性. 表达式如下:

$$
\begin{aligned}
&x=[x_1,x_2,...,x_d], x'=[x'_1,x'_2,...,x'_d]\\
&x_j=lb_j+rand\times(ub_j-lb_j), j=1,2,3,...,d\\
&x'_j=ub_j+lb_j-x_j, j=1,2,3,...,d\\
&x=\begin{cases} x', f(x')\le f(x)\\ x, else\end{cases}
\end{aligned}
\tag{2}
$$

其中 $x=[lb, ub]$, lb 和 ub 分别表示 x 的下界和上界. 双向初始化首先同时考虑候选解的两个方向, 一个是随机产生解 x, 一个是与其相反方向解 x', 再从 $\{x\cup x'\}$ 中选择适应度值较好的解组成最终初始种群.

2.3 基于交叉与变异算子的位置更新

SSA 的寻优能力很大程度上取决于发现者的搜索能力. 因为发现者不仅是搜索食物的寻找者还是引导其他个体搜索食物的指导者. 所以发现者应当具备搜索范围广、寻优方式灵活和寻找能力强的搜索机制才能够去指引算法寻找到全局最优. 一旦发现者易陷入局部最优, 算法的性能也会随之下降. 故提升发现者的搜索能力是相当重要的. 由原发现者更新公式可知麻雀在 $R_2<ST$ 安全区域内, 通过当前位置乘以小于 1 的指数, 跳跃到靠近最优解的新位置, 以进行开发, 加快了收敛速度. 而当 $R_2\geq ST$ 时, 即当前位置处于不安全区域时, 麻雀从当前位置向外执行大于 0 的随机步长远离危险, 这会使个体没有充分利用其他个体的信息, 导致收敛速度慢和全局探索能力弱. 本文将差分进化算法中的交叉与变异算子引入到发现者位置更新公式中, 充分利用被忽略的个体信息, 扩大发现者搜索范围, 使其更加充分探索以优化全局探索能力和提高搜索速度与精度. 算子表达式如下:

变异算子: “DE/rand/1”

$$v_i^t = x_{r0}^t + F_i \times (x_{r1}^t - x_{r2}^t) \tag{3}$$

其中 $i\neq r0\neq r1\neq r2$ 均是 1,2,3,.....,n 中互不相同的整数; F_i 是第 i 个体的缩放因子;

交叉算子:

$$u_{i,j}^t = \begin{cases} v_{i,j}^t, rand(0,1) \leq CR_i || j = j_r \\ x_{i,j}^t, else \end{cases} \tag{4}$$

其中交叉概率 $CR_i \in [0,1]$, $j_r = \text{randint}(1, d)$ 是从 1 到 d 中的一个随机整数.

选择算子:

$$x_i^{t+1} = \begin{cases} u_i^t, f(u_i^t) \leq f(x_i^t) \\ x_i^t, else \end{cases} \tag{5}$$

从原参加者更新公式中可以看出, 若 $i>n/2$ 时, 其更新值为一个标准正态分布随机数乘于指数, 一个以自身与最差个体位置差值除于个体数平方的值为底数的指数, 即麻雀收敛于当前最优解的方式是直接跳跃到当前最优解附近, 而非通过向最优解移动. 从原侦察者更新公式中可以看出, 如果当前最优麻雀意识到危险 ($f_i=f_g$), 它会根据自身与最差个体位置差值和适应度差值的比值来决定移动到安全区域的步长. 参加者与侦察者的个体们均通过共享全局最差个体位置信息来搜索, 限制了种群的搜索范围和种群多样性, 导致算法易陷入局部最优. 因此本文提出了交叉与变异算子位置更新公式来提升参加者与侦察者的寻优能力. 交叉与变异算子能够让个体们共享并学习随机种群的位置信息, 这不仅实现了参加者位置移动更新方式, 而且扩大两者的搜索范围, 从而提升了算法探索未知领域的能力, 有效地降低算法陷入局部最优的概率, 提高了算法全局搜索能力与收敛性能. 改进后的位置更新公式如下:

原发现者位置更新公式调整更新为公式 (6):

$$X_{i,j}^t = \begin{cases} PX_{i,j}^t \times \exp(-\dfrac{i}{\alpha \cdot t_{\max}}), R_2 < ST \\ \begin{cases} C(M(PX_{i,j}^t)), r \leq CR_i || j = j_r \\ PX_{i,j}^t + Q \times L, else \end{cases}, R_2 \geq ST \end{cases} \tag{6}$$

$$PX^t = X^t \tag{7}$$

其中 C 和 M 分别表示交叉算子和变异算子. r 是从 0 到 1 的随机数.

原参加者位置更新公式调整更新为公式 (8):

$$X_{i,j}^t = \begin{cases} \begin{cases} C(M(PX_{i,j}^t)), r \leq CR_i || j = j_r \\ Q \times \exp(\dfrac{X_w^t - PX_{i,j}^t}{i^2}), else \end{cases}, i > \dfrac{n}{2} \\ X_P^{t+1} + \left|PX_{i,j}^t - X_P^{t+1}\right| \times A^+ \times L, i \leq \dfrac{n}{2} \end{cases} \tag{8}$$

原侦察者位置更新公式调整为公式 (9):

$$X_{i,j}^t = \begin{cases} X_b^t + \beta \times \left|PX_{i,j}^t - X_b^t\right|, f_i > f_g \\ \begin{cases} C(M(PX_{i,j}^t)), r \leq CR_i || j = j_r \\ PX_{i,j}^t + K \times (\dfrac{\left|PX_{i,j}^t - X_w^t\right|}{(f_i - f_w) + \varepsilon}), else \end{cases}, f_i = f_g \end{cases} \tag{9}$$

2.4 社区学习

SSA 很难能在每次搜索中都能够得到最可靠的全局最优解. 当遇到局部最优解时, 会导致算法过早收敛, 算法性能降低. 于是我们基于由公式 (6)-(9) 更新后得到的种群 X, 提出了社区学习策略, 通过以 $X=[X_1, X_2, ..., X_n]$ 中各个个体 X_i 为圆心, 指定半径 R_i, 划分出各个个体 X_i 的社区区域 O_i, X_i 在 O_i 中随机选择个体来学习, 生成新的个体 H_i, 然后根据适应度好坏进行选择操作, 得到最终的解集 $X\in [X, H]$. 该策略是利用了 SSA 随着迭代次数的增加, 种群会不断聚集在一个区域并趋向最优解的特性, 从而把父代更新为子代之间的步长作为半径, 划分出各自以

己为中心的社区来形成一个局部搜索区域. 社区学习利用父代与领域内个体之间的位置差异引导子代个体在社区内寻求更优解, 让子代个体在已知解集之间进行深度搜索, 促进个体向最优解方向移动或帮助个体跳出局部最优解, 充分开发了局部搜索区域, 为下一次迭代保留高质量解来指引算法搜索方向, 加快寻优速度, 维持种群多样性, 提升收敛精度. 具体表达式如下:

$$\begin{aligned} O_i^t &= \{k|dist(X_i^t, X_k^t) \le R_i, i \ne k\} \\ R_i &= \left\|PX_i^t - X_i^t\right\| \end{aligned} \tag{10}$$

$$H_{i,j}^t = X_{i,j}^t + rand \times (PX_{l,j}^t - X_{o,j}^t), o \in O_i^t \tag{11}$$

$$X_i^t = \begin{cases} H_i^t, f(H_i^t) \le f(X_i^t) \\ X_i^t, else \end{cases} \tag{12}$$

$$X_i^{t+1} = \begin{cases} X_i^t, f(X_i^t) \le f(PX_i^t) \\ PX_i^t, else \end{cases} \tag{13}$$

其中 l 是 1,2,3,...,n 中的一个随机整数, n 是种群规模.

3 约束处理

约束优化问题求解存在约束区域离散、等式约束、非线性约束等挑战. 如何处理可行解与不可行解的关系是算法能否高效求解约束优化问题的关键. 本文参考文献 [20], 引入约束违规值 $V=(v_1,v_2,...,v_n)$ 来处理约束. 可行性法则通过根据适应度值 F 和 V 对解集 X 进行可行解 (v_i=0) 与不可行解 (v_i≠0) 的划分与排序, 来综合考虑种群的可行性和优劣性, 从而充分处理和利用解集的关系, 使得本文算法在求解约束优化问题上具有更高效的性能. 表达式如下:

$$v_k = \frac{(\sum_{i=1}^{l} G_i(X_k) + \sum_{j=l+1}^{m} H_j(X_k))}{m} \tag{14}$$

$$\begin{aligned} G_i(X) &= \begin{cases} g_i(X), g_i(X) > 0 \\ 0, g_i(X) \le 0 \end{cases} \\ H_j(X) &= \begin{cases} |h_j(X)|, |h_j(X)| - \delta > 0 \\ 0, |h_j(X)| - \delta \le 0 \end{cases} \end{aligned} \tag{15}$$

$$Sort(X_i, X_j) = \begin{cases} X_i \gg X_j, f_i < f_j \&\& v_i = v_j = 0 \\ X_i \gg X_j, v_i = 0 \&\& v_j \ne 0 \\ X_i \gg X_j, v_j > v_i > 0 \end{cases} \tag{16}$$

其中 v_i 是第 i 个个体在 m 个约束函数值总和的算术平均值, v_i≠0 则说明第 i 个个体在 m 个约束函数中至少有一个不满足, 否则 m 个约束全部满足. $G_i(X_k)$ 和 $H_j(X_k)$ 分别表示第 k 个个体在第 i 个不等式约束违规值和第 j-l 个等式约束违规值. "≫" 表示优于, f_i<f_j 表示在最小 (大) 化问题中 X_i 的适应度小 (大) 于 X_j, $X_i \gg X_j$ 表示 X_i 占优, 排在 X_j 前面.

4 实验结果与分析

在本节中, 通过 MSHSSA 在 23 个经典函数和在 IEEE CEC2017 的 28 个实数约束优化问题[20] 上的实验结果评估其性能. 有关测试问题的定义、性质、表达式等详细介绍见附录. 此外, 为了直观地比较各类算法, 采用 Friedman 检验对各竞争算法进行统计分析排名. 为了公平起见, 以下所有实验均在 MATLAB R2020b、8GB RAM 和 Window-10 平台上进行. 注意除 L-SHADE44 的实验结果来自文献 [21], 其余算法代码从 GitHub 上获取.

4.1 实验设置

为了公平起见, 本文将所有算法的种群大小设置为 100, 最大优化迭代次数设置为 1000, 运行次数设置为 25, 终止条件设置为是否到达最大优化迭代次数, 并记录适应度的平均值 (Mean)、最优值 (Best)、标准差 (Std) 和可行率 (Feasibility Rate,FR)[20] 来比较衡量算法的性能.

4.2 CEC 2017 问题上的比较结果与分析

在本节中, MSHSSA 与其他对比算法在 CEC2017 问题上比较, 对比算法的参数设置与所参考的文献保持一致. 由于篇幅有限, MSHSSA 在非约束优化测试问题的比较研究实验详见附录.

表 1 到表 2 列出了 MSHSSA 与其他对比算法求解 CEC2017 测试问题上的结果与 Friedman 排名, 其中最佳解用黑体表示. 根据表 1 到表 2 可以得出, 在 C17-C19 和 C26-C28 问题上, 没有一个算法能够求解出满足约束条件的最优解. 而在 C01、C02、C04 和 C20 问题上, 几乎所有算法的可行率均为 100%. 在 C01-C04、C06、C07、C09、C10、C12-16、C20、C24 和 C25 这 16 个问题上, MSHSSA 均获得了可行解, 体现了优越的寻优性能.

在 28 个问题中, MSHSSA 在 27 个问题上优于 SSA, 其中 SSA 只确定了 10 个问题的可行解, 而 MSHSSA 求解出了 16 个问题的可行解, 这充分证明了本文提出的改进策略成功提高了 SSA 求解复杂约束问题的寻优性能, 这是因为双向初始化构建了一个更加多元化的初始搜索范围, 接着交叉与变异算子帮助 MSHSSA 建立了一个更加稳定的机制来平衡探索和开发能力, 提高求解出可行解的概率. 在过早收敛、陷入局部极值与产生不可行解的情况下, 发挥社区学习的作用, 跳出局部最优, 选择更优子代, 让算法探索更具方向性, 增强算法的寻优能力. 与 PSO、

SCA、WOA、GWO 和 AOA 这 5 种算法相比, 首先, MSHSSA 在 26 个问题上求解出的解决方案在 Best 和 Mean 值上明显最佳, 在 C05 问题上表现相当, 在 C09 问题上获得最佳 Mean 值. 其次, 就可行率方面而言, AOA、PSO、WOA、SCA 和 GWO 分别只确定了 10 、8、6、4 和 4 个可行解. 与 L-SHADE44 算法在 Best 和 Mean 值方面相比, MSHSSA 在 C03 、C06、C07、C09、C11、C14-C19 和 C26-C28 这 14 个问题上表示更优, 而在 C12 和 C24 问题上求解出的 Best 值与 L-SHADE44 算法相当. 在 C23 和 C25 问题, MSHSSA 和 L-SHADE44 算法分别求解出更优的 Best 值和 Mean 值. 根据上述实验结果与分析可以验证 MSHSSA 在求解 COPs 时的具有很大的潜力, 可以高效应对不同的优化挑战. 从最后一行的 Friedman 排名可以发现, MSHSSA 获得了第一, 再一次证明了 MSHSSA 对 SSA 的改进是成功和有效的.

表 1 算法在 CEC2017 测试问题中 C1-C13 结果的比较

F	指标	PSO	SCA	WOA	GWO	AOA	L-SHADE44	SSA	MSHSSA
C01	Best	1.99E+00	6.27E+03	2.07E+04	2.78E+04	9.06E+01	**0.00E+00**	1.29E+02	7.45E-01
	Mean	5.93E+00	8.96E+03	4.06E+04	7.70E+04	4.38E+02	**3.87E-30**	4.57E+02	2.34E+00
	Std	1.97E+00	2.50E+03	1.93E+04	1.87E+04	2.73E+02	**6.10E-30**	2.20E+02	1.13E+00
	FR	100	100	100	100	100	100	100	100
C02	Best	9.16E-01	5.87E+03	4.00E+04	2.30E+04	1.18E+02	**0.00E+00**	1.80E+02	7.38E-01
	Mean	7.66E+00	8.87E+03	1.16E+05	1.25E+05	5.67E+02	**5.26E-30**	3.02E+02	2.28E+00
	Std	4.49E+00	2.28E+03	3.36E+04	9.53E+04	2.83E+02	**8.39E-30**	1.05E+02	1.34E+00
	FR	100	100	8	100	100	100	100	100
C03	Best	7.23E+04	3.18E+04	7.72E+04	4.12E+04	2.99E+04	3.91E+04	8.17E+04	**2.51E+04**
	Mean	4.73E+05	9.55E+04	2.23E+05	3.14E+05	9.69E+04	3.55E+05	2.18E+05	**7.98E+04**
	Std	9.14E+05	1.01E+05	1.24E+05	1.22E+05	5.31E+04	4.47E+05	1.20E+05	**5.22E+04**
	FR	20	0	96	0	40	100	56	100
C04	Best	1.48E+02	3.28E+02	2.14E+02	4.64E+02	1.53E+02	**1.36E+01**	1.61E+02	1.09E+02
	Mean	1.91E+02	3.67E+02	2.83E+02	5.42E+02	1.95E+02	**1.36E+01**	2.09E+02	1.70E+02
	Std	2.63E+01	2.34E+01	5.67E+01	3.07E+01	2.23E+01	**5.44E-15**	1.79E+01	1.76E+01
	FR	100	100	100	100	100	100	100	100
C05	Best	3.97E+07	3.97E+07	3.97E+07	3.97E+07	3.97E+07	**0.00E+00**	3.97E+07	3.97E+07
	Mean	3.97E+07	3.97E+07	3.97E+07	3.97E+07	3.97E+07	**0.00E+00**	3.97E+07	3.97E+07
	Std	2.28E-08	2.28E-08	2.28E-08	1.51E-08	2.28E-08	**0.00E+00**	2.28E-08	2.28E-08
	FR	0	0	0	0	0	100	0	0
C06	Best	6.56E+02	7.13E+02	1.60E+03	1.26E+03	5.99E+02	3.12E+03	9.59E+02	**4.81E+02**
	Mean	1.95E+03	1.60E+03	2.97E+03	1.58E+03	1.77E+03	4.07E+03	3.02E+03	**1.15E+03**
	Std	7.78E+02	4.23E+02	8.53E+02	**6.81E+01**	3.41E+02	9.82E+02	1.43E+03	1.52E+02
	FR	0	0	0	0	0	0	0	**56**
C07	Best	-7.11E+02	-5.70E+02	-7.91E+02	-3.60E+02	-3.97E+03	-2.46E+02	-9.01E+02	**-1.18E+03**
	Mean	-4.08E+02	-3.71E+02	-6.35E+02	-3.48E+01	-2.87E+03	-1.09E+02	-6.78E+02	**-8.90E+02**
	Std	1.54E+02	9.74E+01	1.05E+02	9.26E+01	9.74E+02	**8.87E+01**	1.05E+02	1.15E+02
	FR	0	0	0	0	32	96	0	84
C08	Best	2.04E+00	8.15E+00	3.53E+01	8.38E+00	1.75E+00	**-2.80E-04**	7.84E-01	2.22E-01
	Mean	4.30E+00	1.33E+01	7.45E+01	8.84E+00	3.23E+00	**-2.80E-04**	1.42E+00	4.58E-01
	Std	1.37E+00	4.62E+00	1.38E+01	2.37E-02	9.86E-01	**0.00E+00**	4.01E-01	1.76E-01
	FR	0	0	0	0	0	**100**	0	0
C09	Best	-1.74E-01	2.09E+00	**-6.09E-01**	7.26E+00	-4.18E-01	-2.67E-03	-3.19E-01	-2.60E-01
	Mean	6.69E-01	3.05E+00	2.09E+00	9.13E+00	4.03E-01	-2.67E-03	2.91E+00	**-8.62E-03**
	Std	7.15E-01	7.84E-01	2.16E+00	3.07E-02	6.41E-01	**1.33E-18**	4.92E+00	1.08E-01
	FR	0	0	0	0	8	100	16	92
C10	Best	2.12E+00	3.26E+01	5.23E+01	2.78E+01	1.07E+00	**-1.00E-04**	3.67E-01	3.22E-02
	Mean	4.42E+00	3.92E+01	8.44E+01	4.21E+01	2.13E+00	**-1.00E-04**	3.28E+00	1.69E-01
	Std	2.18E+00	4.03E+00	1.89E+01	4.23E-01	9.64E-01	**0.00E+00**	1.11E+01	7.49E-02
	FR	0	0	0	0	0	100	0	4
C11	Best	-2.36E+03	-2.12E+03	-2.39E+03	-1.07E+03	-1.33E+03	-9.25E-01	-2.81E+03	**-2.86E+03**
	Mean	-1.20E+03	-1.74E+03	-2.11E+03	-6.85E+01	-1.06E+03	-8.75E-01	-1.68E+03	**-2.30E+03**
	Std	6.50E+02	2.47E+02	1.64E+01	2.67E+02	1.36E+02	**1.10E-01**	1.04E+03	9.06E+02
C12	Best	3.17E+02	5.12E+03	4.33E+02	1.53E+04	5.93E+01	**3.98E+00**	2.94E+01	**3.98E+00**
	Mean	7.35E+02	6.62E+03	6.75E+02	1.72E+04	1.80E+02	**3.99E+00**	6.59E+01	2.44E+01
	Std	2.23E+02	9.44E+02	1.55E+02	1.86E+02	2.46E+02	**1.44E-03**	2.67E+01	8.72E+00
	FR	0	0	0	0	0	**100**	0	16
C13	Best	6.80E+05	8.74E+07	6.34E+05	1.37E+09	3.02E+03	**2.34E-01**	9.03E+04	7.35E+01
	Mean	3.79E+06	3.40E+08	4.58E+06	1.78E+09	4.90E+05	**6.15E+00**	6.76E+05	2.05E+03
	Std	2.81E+06	1.32E+08	2.90E+06	2.23E+07	9.61E+05	**4.72E+00**	5.81E+05	3.70E+03
	FR	0	0	0	0	16	100	0	100

表 2　算法在 CEC2017 测试问题中 C14-C28 结果的比较

F	指标	PSO	SCA	WOA	GWO	AOA	L-SHADE44	SSA	MSHSSA
C14	Best	1.89E+01	2.09E+01	2.03E+01	2.11E+01	4.63E-01	1.56E+00	8.72E-01	**9.12E-01**
	Mean	2.00E+01	2.10E+01	2.08E+01	2.13E+01	3.66E+00	1.83E+00	1.06E+01	**1.33E+00**
	Std	4.90E-01	8.21E-02	1.51E-01	**3.34E-02**	5.29E+00	8.34E-02	9.67E+00	2.05E-01
	FR	0	0	0	0	0	100	4	72
C15	Best	1.45E+01	2.44E+01	2.11E+01	3.67E+01	1.45E+01	1.18E+01	1.41E+01	**1.14E+01**
	Mean	1.59E+01	3.23E+01	3.01E+01	4.37E+01	1.69E+01	1.92E+01	1.61E+01	**1.51E+01**
	Std	2.17E+00	4.53E+00	3.59E+00	3.24E+01	2.34E+00	3.61E+00	1.82E+00	**8.99E-01**
	FR	0	0	0	0	0	84	8	96
C16	Best	1.70E+02	3.06E+02	1.32E+02	5.65E+02	1.13E+02	1.15E+02	1.40E+02	**1.13E+02**
	Mean	1.98E+02	3.47E+02	1.95E+02	6.42E+02	1.50E+02	**1.46E+02**	1.74E+02	**1.46E+02**
	Std	1.21E+01	1.89E+01	2.41E+01	**1.28E-01**	2.47E+01	1.40E+01	1.80E+01	1.56E+01
	FR	36	0	100	0	56	100	24	100
C17	Best	8.04E-01	2.15E+00	9.14E-01	4.34E+00	1.69E-01	9.48E-01	6.49E-01	**1.09E-01**
	Mean	1.00E+00	2.59E+00	1.02E+00	5.70E+00	9.73E-01	9.97E-01	9.98E-01	**9.68E-01**
	Std	5.01E-02	2.49E-01	2.90E-02	4.74E-02	2.72E-01	**1.67E-02**	8.21E-02	3.99E-02
	FR	0	0	0	0	0	0	0	0
C18	Best	4.45E+02	5.13E+03	5.57E+02	1.51E+04	4.40E+01	2.91E+02	1.13E+01	**3.91E+01**
	Mean	9.92E+02	7.06E+03	1.12E+03	1.98E+04	2.23E+02	2.79E+03	2.04E+02	**1.08E+02**
	Std	3.78E+02	9.96E+02	4.20E+02	5.53E+02	2.60E+02	4.35E+03	5.50E+01	**4.37E+01**
	FR	0	0	0	0	0	0	0	0
C19	Best	5.31E+01	6.88E+01	6.79E+01	1.06E+02	1.28E+01	5.60E-06	2.91E+01	**-1.46E+01**
	Mean	6.61E+01	8.30E+01	9.30E+01	1.27E+02	3.53E+01	6.20E-06	6.14E+01	**-1.10E-01**
	Std	7.22E+00	7.91E+00	1.06E+01	7.96E+00	1.29E+01	**3.40E-07**	1.92E+01	7.60E+00
	FR	0	0	0	0	0	0	0	0
C20	Best	2.80E+00	7.93E+00	5.41E+00	8.65E+00	1.85E+00	**1.28E+00**	2.44E+00	1.63E+00
	Mean	5.13E+00	8.57E+00	6.78E+00	1.13E+01	4.68E+00	**1.42E+00**	3.85E+00	2.88E+00
	Std	1.22E+00	3.57E-01	9.46E-01	9.13E-01	1.44E+00	**9.91E-02**	1.28E+00	7.26E-01
	FR	100	100	100	100	100	100	100	100
C21	Best	6.47E+02	1.31E+04	9.81E+02	7.41E+04	1.24E+02	**9.78E+00**	1.51E+02	7.54E+01
	Mean	9.88E+02	2.19E+04	1.73E+03	1.00E+05	6.30E+02	**2.25E+01**	2.86E+02	1.30E+02
	Std	2.20E+02	4.55E+03	5.44E+02	1.17E+03	4.80E+02	**8.91E+00**	7.40E+01	3.27E+01
	FR	0	0	0	0	0	**100**	0	0
C22	Best	8.93E+05	2.78E+09	9.70E+06	3.65E+10	4.88E+04	**3.16E+00**	2.80E+05	1.71E+04
	Mean	6.67E+06	5.66E+09	3.95E+07	6.41E+10	5.32E+06	**1.27E+03**	1.37E+06	2.96E+05
	Std	8.28E+06	2.17E+09	2.89E+07	2.07E+09	9.93E+06	**1.41E+03**	8.08E+05	2.32E+05
	FR	0	0	0	0	0	**100**	0	0
C23	Best	2.09E+01	2.10E+01	2.07E+01	2.11E+01	2.07E+01	1.50E+00	2.05E+01	**1.48E+00**
	Mean	2.10E+01	2.11E+01	2.09E+01	2.16E+01	2.09E+01	**1.72E+00**	2.09E+01	1.64E+01
	Std	**5.61E-02**	7.82E-02	9.12E-02	9.94E-02	8.92E-02	1.27E-01	2.23E-01	6.16E+00
	FR	0	0	0	0	0	**100**	0	0
C24	Best	1.23E+01	4.59E+01	1.91E+01	1.00E+02	1.22E+01	**1.18E+01**	**1.18E+01**	**1.18E+01**
	Mean	1.57E+01	5.65E+01	3.87E+01	1.21E+02	1.52E+01	**1.22E+01**	1.50E+01	1.46E+01
	Std	1.92E+00	5.49E+00	1.03E+01	2.26E+00	1.40E+00	1.04E+00	1.17E+00	**8.64E-01**
	FR	4	0	0	0	0	100	0	96
C25	Best	1.59E+02	5.49E+02	1.77E+02	1.11E+03	1.38E+02	1.13E+02	1.63E+02	**1.03E+02**
	Mean	1.96E+02	6.77E+02	2.48E+02	1.30E+03	1.84E+02	**1.42E+02**	1.90E+02	1.75E+02
	Std	1.61E+01	7.00E+01	6.78E+01	3.43E+00	1.87E+01	1.39E+01	1.71E+01	**1.31E+01**
	FR	24	0	0	0	36	100	36	100
C26	Best	8.65E-01	5.09E+00	1.01E+00	1.91E+01	9.55E-01	9.55E-01	9.33E-01	**8.13E-01**
	Mean	9.94E-01	6.81E+00	1.07E+00	2.49E+01	1.07E+00	9.97E-01	1.01E+00	**9.93E-01**
	Std	4.61E-02	9.65E-01	4.17E-02	1.33E+00	1.53E-01	**1.64E-02**	2.82E-02	2.82E-02
	FR	0	0	0	0	0	0	0	0
C27	Best	6.86E+02	1.66E+04	1.69E+03	5.39E+04	2.25E+02	1.19E+03	2.35E+02	**1.18E+02**
	Mean	1.54E+03	2.75E+04	3.31E+03	9.76E+04	8.82E+02	9.32E+03	4.12E+02	**3.78E+02**
	Std	8.38E+02	6.10E+03	1.23E+03	5.03E+03	8.74E+02	8.79E+03	1.81E+02	**8.56E+01**
	FR	0	0	0	0	0	0	0	0
C28	Best	1.10E+02	1.05E+02	1.17E+02	1.56E+02	8.51E+01	4.52E+01	1.24E+02	**6.63E+01**
	Mean	1.35E+02	1.30E+02	1.47E+02	1.83E+02	1.49E+02	1.43E+02	1.48E+02	**1.03E+02**
	Std	1.67E+01	1.28E+01	1.73E+01	7.49E+00	1.60E+01	2.45E+01	1.87E+01	**1.41E+01**
	FR	0	0	0	0	0	0	0	0
Rank		4.57	6.16	5.68	7.39	3.71	2.86	3.91	**1.71**

4.3 实际应用

在本节中, 用 MSHSSA 求解 4 个多维非线性结构设计优化问题: Himmelblau 的非线性优化问题, 弹簧设计问题, 三杆桁架设计问题和工字梁设计问题.

4.3.1 Himmelblau 非线性优化问题

Himmelblau 非线性优化问题的目标是找到一个决策向量 Z, 它将最小化以下模型的目标函数 f.

$$\begin{aligned}
&\text{Consider}: Z=[z_1,z_2,z_3,z_4,z_5]\\
&\text{Minimize}: f(Z)=5.3578547{z_3}^3+0.8356891z_1z_5\\
&\qquad\qquad +37.293239z_1+40729.141\\
&\text{s.t}:\\
&g_1(Z)=85.334407+0.0056858z_2z_5+0.0006262z_1z_4\\
&\qquad -0.0022053z_3z_5\\
&g_2(Z)=80.51249+0.0071317z_2z_5+0.0029955z_1z_2\\
&\qquad +0.0021813{z_3}^2\\
&g_3(Z)=9.300961+0.0047026z_3z_5+0.0012547z_1z_3\\
&\qquad +0.0019085z_3z_4\\
&0\le g_1\le 92, 90\le g_2\le 110, 20\le g_3\le 25\\
&\text{Variable range}:\\
&78\le z_1\le 102, 33\le z_2\le 45, 27\le z_3,z_4,z_5\le 45
\end{aligned}$$

表 3 列出了 MSHSSA 和其他对比算法求解 Himmelblau 非线性问题的最佳解. 根据表 3 可以得出, MSHSSA 求解出的最佳可行解为 Z=(102, 33, 29.1, 27, 40), f=98500. 与其他算法对比, MSHSSA 获得了最优可行解, AOA 因违背约束条件 g_3, 故解是不可行的. 表 4 列出了 MSHSSA 与其他算法在 Himmelblau 非线性问题上的统计结果. 根据表 4 可以得出, 在获得可行解的所有算法中, MSHSSA 在可行解的最优值、平均值和标准差方面都达到了最佳, 这表明了 MSHSSA 求解 Himmelblau 非线性问题时具有有效性和精度高的特点. 由于篇幅有限, 求解其他 3 个工程优化问题的实验和关于算法的时间复杂度分析详见附录.

5 总结

本文提出了一种多策略麻雀搜索算法, 用于求解复杂约束优化问题. 本算法引入双向初始化机制调节初始种群分布均匀性, 提高初始解质量的同时使个体搜索更具方向性; 加入交叉与变异算子改进位置更新公式, 提升算法的全局探索遍历性, 根据收敛情况自适应调整算法的开发与探索能力, 加快收敛速度, 提高寻优精度; 融入社区学习策略精炼种群, 维持种群多样性的同时保证算法充分发掘有希望区域, 有利于算法摆脱局部极值点; 引入可行性法则, 高效复杂处理约束优化问题. 利用 23 个非约束优化测试问题、28 个实数约束优化问题和 4 个工程设计问题验证本文算法的性能. 综合实验结果表明, 本文算法具有较好的全局收敛性, 鲁棒性和种群多样性, 在求解复杂优化问题上比原算法表现出更优秀的寻优能力和更高效的综合性能. 然而, MSHSSA 也存在一些限制与缺陷, 即在面对极具欺骗性和极难搜索问题时, 还需要进一步提升全局搜索能力与局部开发能力, 且对比原算法也增加了一定的时间复杂度. 在未来的研究当中, 将进一步研究结合不同策略和约束处理方式来求解复杂约束优化问题.

表 3 不同算法对 Himmelblau 非线性问题的最佳解决方案

算法	决策变量的最优值					约束			f
	z_1	z_2	z_3	z_4	z_5	g_1	g_2	g_3	Best
SSA	1.02E+02	3.30E+01	2.91E+01	2.70E+01	4.00E+01	9.20E+01	1.02E+02	2.00E+01	9.85E+04
WOA	1.02E+02	3.31E+01	2.99E+01	4.46E+01	3.08E+01	9.20E+01	9.60E+01	2.00E+01	1.09E+05
SCA	1.02E+02	3.30E+01	2.92E+01	2.70E+01	3.98E+01	9.20E+01	1.02E+02	2.00E+01	9.99E+04
GWO	1.02E+02	3.30E+01	2.91E+01	2.70E+01	4.00E+01	9.20E+01	9.82E+01	2.00E+01	9.85E+04
PSO	1.02E+02	3.30E+01	2.91E+01	2.70E+01	4.00E+01	9.20E+01	1.02E+02	2.00E+01	9.85E+04
AEFA-C	1.02E+02	3.30E+01	2.97E+01	3.81E+01	3.46E+01	9.20E+01	9.68E+01	2.01E+01	1.07E+05
AOA	7.80E+01	3.30E+01	2.70E+01	2.70E+01	2.70E+01	9.01E+01	9.62E+01	1.68E+01	6.94E+04
MSHSSA	1.02E+02	3.30E+01	2.91E+01	2.70E+01	4.00E+01	9.20E+01	1.02E+02	2.00E+01	9.85E+04

表 4 不同算法对 Himmelblau 非线性问题的统计结果

算法	Best	Mean	Std	FR(%)
SSA	9.85E+04	1.02E+05	7.24E+03	100
WOA	1.09E+05	1.21E+05	8.99E+03	56
SCA	9.99E+04	1.03E+05	2.55E+03	36
GWO	9.85E+04	9.86E+04	7.67E+01	52
PSO	9.85E+04	9.99E+04	2.11E+03	100
AEFA-C	1.07E+05	1.26E+05	1.24E+04	100
AOA	6.94E+04	6.94E+04	0.00E+00	0
MSHSSA	9.85E+04	**9.85E+04**	1.54E+01	100

参考文献 (References)

[1] Karaboga D, Akay B. A modified artificial bee colony (ABC) algorithm for constrained optimization problems[J]. Applied soft computing, 2011,11(3): 3021-3031.

[2] Goldberg D E . Genetic Algorithm in Search, Optimization, and Machine Learning[M]. Addison-Wesley Pub. Co,1989.

[3] Storn R, Price K. Differential Evolution–A Simple and Efficient Heuristic for global Optimization over Continuous Spaces[J]. Journal of Global Optimization, 1997,11(4):341-359.

[4] Kennedy J, Eberhart R. Particle swarm optimi- zation[C]. Proceedings of ICNN'95-international conference on neural networks. IEEE, 1995,4: 1942-1948.

[5] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in engineering software, 2014,69: 46-61.

[6] Mirjalili S, Lewis A. The whale optimization algorithm[J]. Advances in engineering software, 2016,95: 51-67.

[7] Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems[J]. Knowledge-based systems, 2016,96: 120-133.

[8] Hashim F A, Hussain K, Houssein E H, et al. Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems[J]. Applied Intelligence, 2021,51(3): 1531-1551.

[9] Yadav A. AEFA: Artificial electric field algorithm for global optimization[J]. Swarm and Evolutionary Computation, 2019,48: 93-108.

[10] Wolpert D H, Macready W G. No free lunch theorems for optimization[J]. IEEE transactions on evolutionary computation, 1997,1(1): 67-82.

[11] Xue J, Shen B. A novel swarm intelligence optimization approach: sparrow search algorithm[J]. Systems Science & Control Engineering, 2020,8(1): 22-34.

[12] Tizhoosh H R. Opposition-based learning: a new scheme for machine intelligence[C]. International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06). IEEE, 2005,1: 695-701.

[13] 李智勇, 黄滔, 陈少淼, 李仁发. 约束优化进化算法综述 [J]. 软件学报, 2017,28(6):1529−1546. http://www.jos.org.cn/1000- 9825/5259.htm.
(Li Z Y, Huang T, Chen S M, Li R F. Overview of Constrained Optimization Evolutionary Algorithms[J]. Journal of Software, 2017,28(6):1529-1546 (in Chin-ese). http://www.jos.org.cn/1000-9825/5259.htm.)

[14] Yuan J, Zhao Z, Liu Y, et al. DMPPT control of photovoltaic microgrid based on improved sparrow search algorithm[J]. IEEE Access, 2021,9: 16623-16629.

[15] Wang P, Zhang Y, Yang H. Research on Economic Optimization of Microgrid Cluster Based on Chaos Sparrow Search Algorithm[J]. Computational Intell-igence and Neuroscience, 2021,2021.

[16] Liang Q, Chen B, Wu H, et al. A Novel Modified Sparrow Search Algorithm with Application in Side Lobe Level Reduction of Linear Antenna Array[J]. Wireless Communications and Mobile Computing, 2021,2021.

[17] Zielinski, Karin, and Rainer Laur. Constrained single-objective optimization using differential evolution[C]. 2006 IEEE International Conference on Evolutionary Computation. IEEE, 2006: 223-230.

[18] Wang Y ,Wang BC ,Li HX ,Yen GG. Incorporating objective function information into the feasibility rule for constrained evolutionary optimization[J]. IEEE Transactions on Cybernetics, 2015,46(12): 2938−2952.

[19] Yadav A, Kumar N. Artificial electric field algorithm for engineering optimization problems[J]. Expert Systems with Applications, 2020,149: 113308.

[20] Wu G, Mallipeddi R, Suganthan P N. Problem Definitions and Evaluation Criteria for the CEC 2017 Competition and Special Session on Constrained Single Objective Real-Parameter Optimization. 2016.

[21] Polakova R. L-SHADE with competing strategies applied to constrained optimization[C]. 2017 IEEE congress on evolutionary computation (CEC). IEEE, 2017: 1683-1689.

作者简介

刘耿耿 (1988−), 男, 副教授, 博士生导师, 从事集成电路设计自动化算法、计算智能及其应用、机器学习等研究, E-mail: liugenggeng@fzu.edu.cn;

张丽媛 (1998−), 女, 硕士研究生, 从事智能优化算法的研究, E-mail: liyuan123zhang@163.com;

刘　笛 (1997−), 女, 博士研究生, 从事智能优化算法的研究, E-mail: 201971140@yangtzeu.edu.cn.

刘能现 (1983−), 男, 助理研究员, 博士研究生, 从事进化算法及其在现实问题中的应用的研究, E-mail: lylnx@fzu.edu.cn.

傅仰耿 (1981−), 男, 教授, 博士生导师, 从事数据挖掘、机器学习、智能系统、认知图谱等研究, E-mail: ygfu@qq.com.

郭文忠 (1979−), 男, 教授, 博士生导师, 从事多媒体智能信息处理、网络计算与分析、数据挖掘等研究, E-mail: fzugwz@163.com.

陈国龙 (1965−), 男, 教授, 博士生导师, 从事智能信息处理、计算机网络信息安全等方向的基础理论及其应用研究, E-mail: fzucgl@163.com.

蒋伟进 (1964−), 男, 教授, 博士, 从事社会计算、群智协同、分布式网络研究, E-mail: jwjnudt@163.com.