

基于窦性变异的改进人工蜂群白骨顶鸡算法及应用

张羽,何庆

(贵州大学大数据与信息工程学院,贵州 贵阳 550025)

摘要:针对白骨顶鸡算法(COOT)存在求解精度低、收敛速度较慢和易陷入局部最优的问题,该文提出一种基于窦性变异的改进人工蜂群白骨顶鸡算法(ICOOT)。首先,采用精英反向学习策略初始化个体位置,增加初始个体寻优多样性;其次,考虑到人工蜂群算法强大的搜索能力,提出一种以全局最优值引导的改进人工蜂群搜索策略,更新白骨顶鸡个体的位置,以提高COOT的搜索能力和收敛精度;最后,引入窦性变异策略对最优个体进行扰动,一方面使算法能够有效跳出局部最优,另一方面提高最优个体质量。利用12个基准测试函数对ICOOT进行寻优性能测试,将ICOOT应用于拉力/压力弹簧优化工程设计问题,并与其他元启发式算法进行了比较和分析,结果验证了改进的算法的可行性和优越性。

关键词:白骨顶鸡算法;精英反向学习;人工蜂群算法;窦性变异策略;工程设计问题

中图分类号:TP301

文献标识码:A

文章编号:1673-629X(2024)04-0162-06

doi:10.20165/j.cnki.ISSN1673-629X.2024.0024

Improved Artificial Bee Colony Coot Algorithm Based on Cosine Mutation and Its Application

ZHANG Yu, HE Qing

(School of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China)

Abstract: Aiming at the problems of low solution accuracy, slow convergence and local optimality in COOT algorithm, we propose an improved artificial bee colony white-bone top chicken algorithm (ICOOT) based on cosine mutation. Firstly, the elite opposition-based learning strategy is used to initialize the individual position and increase the diversity of the initial individual search. Secondly, considering the powerful search ability of the artificial bee colony algorithm, an improved artificial bee colony search strategy guided by the global optimal value is proposed to update the positions of the white-boned top hen individuals to improve the search capability and convergence accuracy of the COOT. Finally, the sinus variation strategy is introduced to perturb the optimal individual, which on the one hand makes the algorithm jump out of the local optimal effectively, and on the other hand improves the quality of the optimal individual. Twelve benchmark test functions are used to test the optimization performance of the ICOOT. The ICOOT is applied to the problem of tension/pressure spring optimization engineering design, and is compared and analyzed with other meta-heuristic algorithms, which verifies the feasibility and superiority of the improved algorithm.

Key words: COOT; elite opposition-based learning; artificial bee colony algorithm; cosine mutation strategy; engineering design problem

0 引言

对于现实世界的问题,随机优化方法已被用于解决各种组合问题。这些优化问题是非线性的,计算昂贵的,多模态的,且拥有较大的解决空间。随着社会的发展和技术的不断更新,元启发式算法能够在合理的时间内解决这些复杂的问题。常见的元启发式算法有

粒子群优化算法^[1](Particle Swarm Optimization, PSO)、布谷鸟搜索算法^[2](Cuckoo Search, CS)、萤火虫算法^[3](Firefly Algorithm, FA)、鲸鱼优化算法^[4](Whale Optimization Algorithm, WOA)、灰狼算法^[5](Grey Wolf Optimizer, GWO)等。

No Free Lunch^[6]定理表明一个特定的优化算法并

收稿日期:2023-08-16

修回日期:2023-12-20

基金项目:国家自然科学基金(62166006);贵州省省级科技计划项目(黔科合支撑[2023]一般093);贵州省省级科技计划项目(黔科合 ZK 字[2021]335)

作者简介:张羽(1996-),女,硕士研究生,研究方向为进化计算、深度学习;通讯作者:何庆(1982-),男,博士,教授,研究方向为大数据应用、进化计算。

不能解决每个问题,因为每个问题都有自己的复杂性和性质。这证明了每个算法在求解特定问题时想要有更好的效果就需要针对性的优化。白骨顶鸡算法^[7](The Coot Algorithm, COOT)是 Naruei 等人于 2021 年提出的一种新的元启发式算法,其灵感来源于白骨顶鸡(水鸟)群的行为。

为了提高 COOT 的性能,众多学者对其进行了改进。例如:Hu 等人^[8]采用反向学习和狩猎策略以加快 COOT 的全局收敛速度和求解精度;Hong 等人^[9]引入一种自适应权值用于求解初始收敛性问题,然后采用混沌方法在整个迭代过程中调整最理想的问题解决方案,将改进后的 COOT 用于深度信念网络(DBN)模型并对存在气候变量的风力发电模型进行优化;Houssein 等人^[10]采用控制随机化、过渡因子和调整位置等 7 种策略去缓解 COOT 易陷入局部最优的问题;张吉祥等人^[11]提出一种增强型白骨顶鸡优化算法,利用拉丁超立方抽样方法均匀初始化种群,加入掉队机制和引入二次插值策略,以提高算法的收敛速度与寻优精度;李雪利等人^[12]引入扰动因子和贪心策略,平衡算法的局部和全局搜索能力;贾鹤铭等人^[13]引入复合突变策略和随机反向策略以提高白骨顶鸡优化算法的全局搜索能力;周雪荃等人^[14]提出一种基于柯西变异和差分进化的混沌白骨顶鸡算法,通过柯西变异使白骨顶鸡位置发生扰动,以提高全局搜索能力,引入差分进化策略和 logistic 混沌因子,从而增加种群多样性且提高算法跳出局部最优的能力。

综上所述,上述文献对 COOT 做出了一定的改进,各具优势,但针对于 COOT 存在的搜索能力弱、寻优速度和精度不高、跳出局部最优能力差的缺点,探索更适合解决实际优化问题的方法仍然是一个值得研究的问题。因此,该文提出一种基于变异性改进的人工蜂群白骨顶鸡算法(ICOOT)。通过引入精英反向学习增加初始个体的寻优遍历性,提高算法收敛速度和精度;考虑到人工蜂群算法强大的搜索能力,在链式移动过程中引入改进的人工蜂群搜索策略,以此来平衡算法的搜索能力,加快算法收敛速度;最后,引入变异性策略对最优个体进行扰动,一方面使算法能够有效跳出局部最优,另一方面提高最优个体质量。最后,通过 12 个基准测试函数的寻优对比实验以及工程设计优化问题验证了 ICOOT 的优越性。

1 COOT

白骨顶鸡是一种小型水鸟,COOT 主要是模拟白骨顶鸡在水面上觅食过程中的两种不同运动模式。在水面上,白骨顶鸡主要有四种不同移动的方式:(1)在水面上随机移动;(2)链式移动;(3)根据领头鸟群的情况调整自身位置;(4)领头鸟群走向最佳区域。

COOT 具体实现原理介绍如下。

1.1 随机移动

为了实现随机移动,在搜索空间考虑一个随机位置,如公式 1 所示:

$$Q = \text{rand}(1, d) * (\text{ub} - \text{lb}) + \text{lb} \quad (1)$$

式中, d 为优化问题的维度, ub 为搜索空间的上界, lb 为搜索空间的下界。随机移动的行为探索了搜索空间的不同位置。如果算法陷入局部最优,这个行为将会使算法跳出局部最优。白骨顶鸡的新位置如公式 2 所示:

$$\text{CPos}(i) = \text{CPos}(i) + A \times r_1 \times (Q - \text{CPos}(i)) \quad (2)$$

式中, $\text{CPos}(i)$ 为白骨顶鸡的当前位置, r_1 为 $[0, 1]$ 区间中的随机数,参数 A 如公式 3 所示:

$$A = 1 - L \times \left(\frac{1}{\text{Iter}_{\max}} \right) \quad (3)$$

式中, L 为迭代次数, Iter_{\max} 为最大迭代次数。

1.2 链式移动

链式移动有两种方式可实现。第一种:两个个体的平均位置链式运动;第二种:首先计算两个个体之间的距离向量,然后将其中一个个体移动到另一个个体,大约是移动距离向量的一半。COOT 使用的是第一种方法,并根据公式 4 计算出个体的新位置。

$$\text{CPos}(i) = 0.5 \times (\text{CPos}(i-1) + \text{CPos}(i)) \quad (4)$$

式中, $\text{CPos}(i-1)$ 为跟随在 $\text{CPos}(i)$ 后的个体。

1.3 根据领头鸟群的情况调整位置

一般情况,水鸟群的活动由水鸟群的领导者们领导,其他个体必须根据水鸟群的领导者调整自己的位置,并向它们移动。COOT 考虑领导者的平均位置,种群可以根据该平均位置更新其位置。考虑到平均位置会导致过早收敛,为了实现这一运动,使用公式 5 中的机制来选择领导者。

$$J = 1 + (i \text{MOD} \text{NL}) \quad (5)$$

式中, i 是当前个体的索引号, NL 是领导者的数量, J 是领导者的索引号。根据选定好的领导者计算个体的下一个位置,如公式 6 所示:

$$\text{CPos}(i) = \text{LPos}(j) + 2 \times r_2 \times \cos(2r\pi) \times (\text{LPos}(j) - \text{CPos}(i)) \quad (6)$$

式中, $\text{CPos}(i)$ 为个体的当前位置, $\text{LPos}(j)$ 为选定的领导者的位置, r_2 为区间 $[0, 1]$ 中的随机数, r 为区间 $[-1, 1]$ 中的随机数。

1.4 领导小组走向最佳区域(领导者运动)

种群必须朝着一个目标(最佳区域)前进,因此领导者需要更新他们对目标的位置。在跟随者向领导者移动的同时,领导者也要向目标移动。有时,领导者必

须离开当前位置,才能找到更好的位置,根据公式 7 更新领导者的位置:

$$\text{LPos}(i) = \begin{cases} B \times r_3 \times \cos(2r\pi) \times (\text{CBest} - \text{LPos}(i)) + \text{gBest}, r_4 < 0.5 \\ B \times r_3 \times \cos(2r\pi) \times (\text{CBest} - \text{LPos}(i)) - \text{gBest}, r_4 \geq 0.5 \end{cases} \quad (7)$$

式中, CBest 是已找到的最佳位置, r_3 和 r_4 是区间 $[0, 1]$ 中的随机数, r 是区间 $[-1, 1]$ 中的随机数, B 根据公式 8 计算:

$$B = 2 - L \times \left(\frac{1}{\text{Iter}_{\max}} \right) \quad (8)$$

2 ICOOT 原理介绍

2.1 精英反向学习

COOT 采用随机初始化种群,所得到的种群随机性高,容易导致种群聚集、重叠的现象发生,使种群搜索速度慢,算法多样性不足。针对这一问题,该文引入精英反向学习^[15] (Elite Opposition-Based Learning, EOBL)策略初始化种群,精英反向学习是通过对当前问题的可行解进行估计,构造其反向解,以此来增加种群的多样性。生成反向白骨顶鸡种群的移动方式可表示为公式 9 和公式 10:

$$\text{CP\ddot{o}s}(i) = \text{lb} + \text{ub} - \text{CPos}(i) \quad (9)$$

$$\text{CPos}(i) = \begin{cases} \text{CP\ddot{o}s}(i), f(\text{CP\ddot{o}s}(i)) < f(\text{CPos}(i)) \\ \text{CPos}(i), f(\text{CP\ddot{o}s}(i)) \geq f(\text{CPos}(i)) \end{cases} \quad (10)$$

式中, lb 为搜索空间下界, ub 为搜索空间上界。如公式 10 所示,如果生成反向种群的适应度较好,则选择反向种群作为初始种群;反之,选择保持原始位置。

2.2 改进的人工蜂群策略

人工蜂群算法 (Artificial Bee Colony, ABC) 是 Karaboga 等人^[16] 在 2005 年提出的一种元启发式算法,该算法主要是模仿蜜蜂采蜜行为来对复杂问题提供优化方案。其主要由三种蜜蜂角色组成,分别为:引领蜂、跟随蜂和侦查蜂。人工蜂群搜索策略的数学模型如公式 11 所示。

$$\text{CBest}(i) = \text{CPos}(i) + \varphi(\text{CPos}(i) - \text{CPos}(k)) \quad i \neq k \quad (11)$$

式中, CBest(i) 为临近的蜜源位置, CPos(i) 为当前引领蜂的位置, CPos(k) 为蜜蜂的随机位置, φ 为区间 $[-1, 1]$ 中的随机数。

人工蜂群算法中的引领蜂发现蜜源信息,选择较好的蜜源作为初始蜜源并招募跟随蜂,这个过程不仅有助于在探索阶段找到最优解,而且保持了其他候选最优解的多样性,但其开发能力较弱。因此,该文在人工蜂群算法的搜索中引入全局最优解引导策略,通过

将当前解远离从种群中随机选择的一个候选解,产生了引导的全局最优解,增加了引领蜂在邻域最优解的搜寻机会,改进的人工蜂群搜索策略的数学模型描述如公式 12 所示。

$$\text{CBest}(i) = \text{CPos}(i) + \varphi(\text{CPos}(i) - \text{CPos}(k)) + \text{rand}(\text{Pbest} - \text{CPos}(i)), i \neq k \quad (12)$$

式中, Pbest 为当前引领蜂的最优位置,即全局最优位置。COOT 以鸟群随机移动而进行随机搜索,这种单一的搜索模式使 COOT 缺少能动性,易陷入局部最优,求解精度低。因此,该文引进改进的人工蜂群搜索策略,帮助 COOT 增加搜索范围,提高 COOT 的开采能力。

2.3 窦性变异策略

为了增强 COOT 个体在后期的相互吸引能力,受文献^[17]启发,该文提出了一种余弦函数来指导当前最优个体进行扰动,使 COOT 跳出局部最优。对于余弦函数来说,其在 $[0, 2\pi]$ 范围区间上的值为 $[-1, 1]$,因此可以使 COOT 对当前的全局最优位置进行小范围的搜索,提高算法的局部极值点利用能力,引入窦性变异的 COOT 位置更新公式如公式 13 和公式 14 所示:

$$\text{CBest}(i+1) = \cos(2\pi \times \text{rand}(1, \text{dim})) \times \text{CBest}(i) \quad (13)$$

$$\text{CBest}(i) = \begin{cases} \text{CBest}(i+1), f(\text{CBest}(i)) < T \\ \text{CBest}(i), \text{otherwise} \end{cases} \quad (14)$$

式中, CBest(i) 为全局最优位置, dim 为优化问题维度。如公式 14 所示,如果余弦窦变后个体的适应度较好,则选择突变结果;反之,选择保持原始位置。窦性变异策略可以避免算法陷入局部最优,同时加快算法收敛。

3 仿真实验分析

3.1 参数设置说明

在本节中,测试功能用于评估 ICOOT 的性能。选取粒子群算法 (PSO)、鲸鱼优化算法 (WOA)、灰狼优化算法 (GWO)、白骨顶鸡优化算法 (COOT) 与 ICOOT 进行基准函数寻优对比。为了确保算法之间比较的公平性,算法的基本参数设置为相同的值,包括种群规模 $N=30$,最大迭代次数 $T_{\max}=500$,维度 $d=30$ 。基准测试函数中,实验以 12 个基准测试函数对 ICOOT 进行寻优性能对比实验,其中包括单峰测试函数 $f_1 - f_6$ 和复杂多峰测试函数 $f_7 - f_{12}$ 。实验运行环境为 Windows10, CPU 为 Intel Core i7-6500U, 主频为 3.40 GHz, 内存为 8 GB, 64 bit 操作系统, 模型在 Matlab R2021(a) 上搭建。

3.2 ICOOT 各改进策略基准测试函数寻优消融实验分析

为了验证 ICOOT 各改进策略的有效性,本节对各改进策略进行消融实验对比分析。实验的对比算法分别为原始 COOT(COOT)、引入精英反向学习的 COOT(JCOOT)、改进人工蜂群引导的 COOT(RCOOT)、基于变异性改进的 COOT(DCOOT)以及该文三个策略融合改进的 COOT(ICOOT)。各算法独立运行 30 次,图 1 显示各对比算法在单峰基准函数 f_1 以及多峰测试函数 f_8 的寻优平均收敛曲线。横坐标为迭代次数,纵坐标为

适应度值。从图 1 可以看出,不管是单峰测试函数还是多峰测试函数,针对 COOT 提出的三个改进策略都体现出了一定的优势,其中对于单峰测试函数 f_1 ,各对比算法相对原始 COOT 都有寻优精度和速度上的提升,对于多峰测试函数 f_8 ,该文引入的三个策略都能直接寻找到最优值,且三个策略组合起来更是体现出了很大的寻优速度优势。说明提出的每个改进策略都提高了 COOT 的寻优精度,且各策略的组合能在很大程度上加快 COOT 的寻优速度。

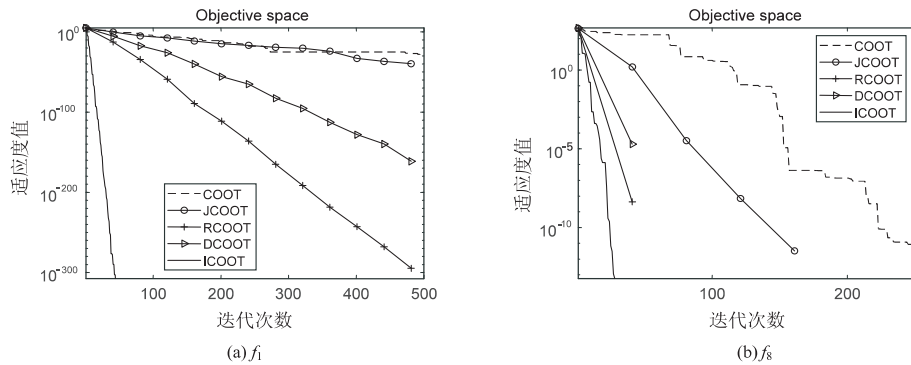


图 1 ICOOT 算法和各改进策略算法收敛对比图(30 维)

3.3 ICOOT 基准测试函数寻优对比分析

为了验证 ICOOT 对于基准测试函数寻优的优势,更加直观地分析仿真实验结果,选取 PSO、WOA、GWO、COOT 与 ICOOT 进行基准测试函数寻优性能对比实验,各算法独立运行 30 次,图 2 显示部分基准函数的平均收敛曲线。横坐标为迭代次数,纵坐标为适应度值。由图 2 可看出,无论是单峰函数还是复杂多峰函数,ICOOT 都能达到理论最优值,特别的,对于单峰函数 f_6 ,前期扩大搜索范围,后期加快算法收敛,说

明引入改进的人工蜂群搜索策略的 ICOOT 对于基本函数搜索上有一定的优势;从多峰函数 f_7 中可看出,ICOOT 在迭代次数接近 300 时能有效跳出局部最优解,说明引入的变异性能够有效帮助算法跳出局部极值点,从多峰函数 f_9 中看出,ICOOT 相比于其他算法在相同数量的迭代下能够快速收敛到理论最优值,说明引入融合多策略的改进策略可以有效地帮助 COOT 在基础函数寻优上展现明显优势,加快算法收敛速度。

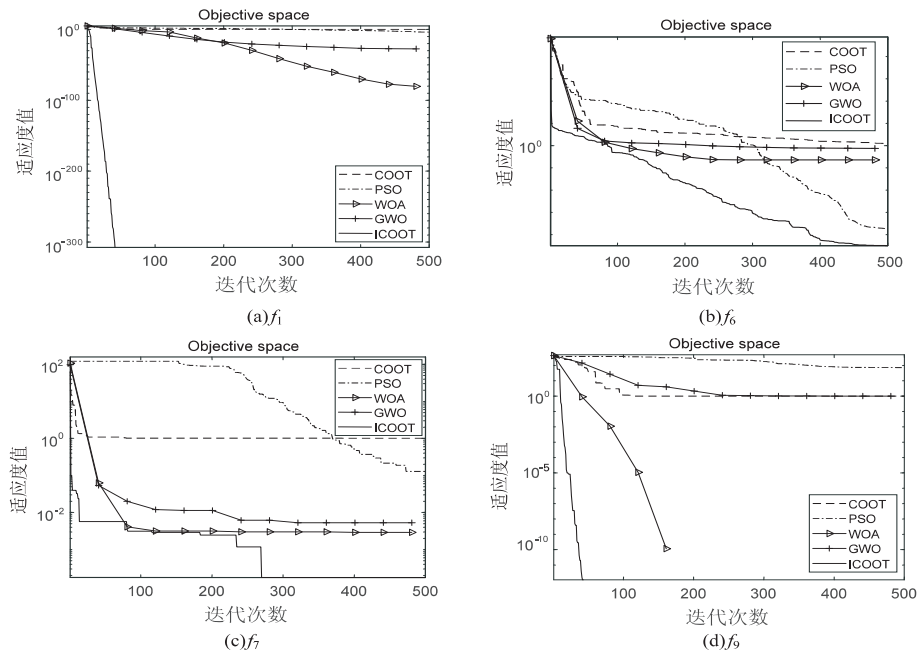


图 2 各算法寻优对比曲线(30 维)

选取 PSO、WOA、GWO、COOT 与 ICOOT 进行基准测试函数寻优性能对比,各算法统一取维度 $d = 30$,最大迭代次数 $T_{\max} = 500$,种群规模 $N = 30$,各函数运行 30 次取平均值和标准差。对比结果如表 1 所示。

表 1 各算法寻优结果对比(30 维)

函数	COOT		PSO ^[4]		WOA ^[4]		GWO ^[5]		ICOOT	
	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
f_1	1.45×10^{-24}	7.97×10^{-24}	1.40×10^{-4}	2.11×10^{-4}	1.41×10^{-30}	4.91×10^{-30}	6.59×10^{-28}	6.34×10^{-5}	0	0
f_2	5.67×10^{-14}	1.88×10^{-13}	4.21×10^{-2}	4.54×10^{-2}	1.06×10^{-21}	2.39×10^{-21}	7.18×10^{-17}	2.90×10^{-2}	0	0
f_3	2.86×10^{23}	1.56×10^{-22}	7.01×10^1	2.21×10^1	5.39×10^{-7}	2.93×10^{-6}	3.29×10^{-6}	7.91×10^1	0	0
f_4	1.48×10^{-13}	6.00×10^{-13}	1.08×10^0	3.17×10^{-1}	7.2×10^{-2}	3.97×10^{-1}	5.61×10^{-7}	1.31×10^0	0	0
f_5	3.18×10^1	7.87×10^0	9.67×10^1	6.01×10^1	2.78×10^1	7.63×10^{-1}	2.68×10^1	6.99×10^1	2.67×10^1	0.79×10^0
f_6	1.93×10^{-1}	1.27×10^{-1}	1.02×10^{-4}	8.28×10^{-5}	3.11×10^0	5.32×10^{-1}	0.82×10^0	1.26×10^{-4}	7.64×10^{-5}	1.77×10^{-5}
f_7	4.40×10^{-3}	4.80×10^{-3}	1.22×10^{-1}	4.49×10^{-2}	1.42×10^{-3}	1.14×10^{-3}	2.21×10^{-3}	1.00×10^{-1}	4.64×10^{-6}	3.73×10^{-6}
f_8	1.89×10^{-13}	9.22×10^{-13}	4.67×10^1	1.16×10^1	0	0	3.11×10^{-1}	4.73×10^1	0	0
f_9	6.45×10^{-10}	3.52×10^{-9}	2.76×10^{-1}	5.09×10^{-1}	7.40×10^0	9.89×10^0	1.06×10^{-13}	7.78×10^{-2}	8.88×10^{-16}	0
f_{10}	2.11×10^{-15}	1.13×10^{-14}	9.21×10^{-3}	7.74×10^{-3}	0.28×10^{-3}	0.15×10^{-3}	4.49×10^{-3}	6.66×10^{-3}	0	0
f_{11}	9.77×10^{-2}	2.85×10^{-1}	6.92×10^{-3}	1.19×10^{-2}	3.39×10^{-1}	2.14×10^{-1}	5.34×10^{-2}	2.07×10^{-2}	2.52×10^{-6}	2.03×10^{-6}
f_{12}	5.48×10^{-1}	5.74×10^{-1}	6.68×10^{-3}	8.91×10^{-3}	1.88×10^0	2.66×10^{-1}	6.54×10^{-1}	4.47×10^{-3}	6.44×10^{-2}	5.32×10^{-2}

由表 1 对比结果可知,在 f_1 、 f_4 、 f_8 和 f_{10} 上,ICOOT 能够收敛为 0,达到理论最优值;在单峰 f_5 、 f_6 和多峰 f_7 、 f_9 和 f_{11} 上,相较于其他四个算法,ICOOT 的平均收敛值较小,更接近理论最有值。由此,ICOOT 在单峰和多峰上表现出了寻优性能的优势,体现了 ICOOT 的鲁棒性。

4 ICOOT 在拉力/压力弹簧优化设计问题的算例分析

优化算法用于解决工程算例应用的优化问题,即为工程应用优化分配节约等问题,又是检验优化算法本身的可行性和稳定性。为了进一步探讨 ICOOT 在处理实际工程应用案例的优越性,对拉力/压力弹簧优化设计问题进行优化处理,并将 ICOOT 的优化结果与基本遗传算法(GA)^[18]、正余弦优化算法(SCA)^[19]、引力搜索算法(GSA)^[20]、灰狼优化算法(GWO)、粒子群优算法(PSO)以及白骨顶鸡优化算法(COOT)的优化结果进行对比。

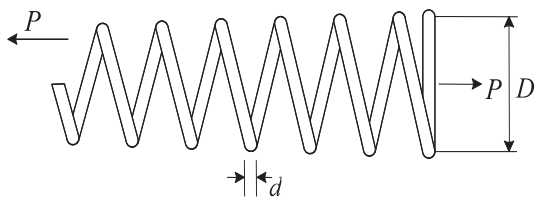


图 3 拉力/压力弹簧问题的示意图

拉力/压力弹簧优化设计问题是指在约束条件下,优化降低弹簧压力/拉力的重量,约束条件为剪切应力、振动频率和最小挠度。有三个设计变量:线圈直径

(d)、平均弹簧圈直径(D)和绕圈数量(P),弹簧优化设计问题模型如图 3 所示。

拉力/压力弹簧优化设计问题数学模型描述如公式 15 所示:

$$\vec{x} = [x_1 \quad x_2 \quad x_3] = [d \quad D \quad P]$$

$$\min f(\vec{x}) = (x_3 + 2)x_1^2 x_2$$

$$\text{s. t. } \begin{cases} g_1(\vec{x}) = 1 - \frac{x_2^3 x_3}{71 \, 785 x_1^4} \leq 0 \\ g_2(\vec{x}) = \frac{4x_2^2 - x_1 x_2}{12 \, 566 (x_2 x_1^3 - x_1^4)} + \frac{1}{5 \, 108 x_1^2} \leq 0 \\ g_3(\vec{x}) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0 \\ g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \end{cases}$$

$$0.05 \leq x_1 \leq 2.0; 0.25 \leq x_2 \leq 1.3$$

$$2.0 \leq x_3 \leq 15.0 \quad (15)$$

式中, x_1 、 x_2 、 x_3 分别为线圈直径(d)、平均弹簧圈直径(D)和绕圈数量(P)的变量, $f(x)$ 为目标函数, $g_1(x)$ 、 $g_2(x)$ 、 $g_3(x)$ 、 $g_4(x)$ 为目标约束条件。

仿真实验各算法独立运行 50 次取平均值,ICOOT 与其他 7 种算法求解拉力/压力弹簧优化设计问题寻优对比结果如表 2 所示。

由表 2 可知,ICOOT 对于拉力/压力弹簧设计问题的优化结果比其他对比算法的好,说明 ICOOT 算法对于工程设计问题的优化效果是优秀的,也进一步说明 ICOOT 在优化目标问题上的可行性和适用性。

表2 拉力/压力弹簧优化设计问题结果对比

算法	d	D	P	平均值
GA ^[21]	0.050 1	0.310 1	14.000 0	0.013 04
SCA ^[21]	0.050 7	0.334 8	12.722 7	0.012 71
GSA ^[5]	0.050 0	0.317 3	14.228 7	0.012 87
GWO ^[5]	0.051 69	0.356 7	11.288 9	0.012 66
PSO ^[21]	0.050	0.310 4	15.000 0	0.013 19
COOT	0.051	0.340 9	12.281 8	0.012 67
ICOOT	0.518	0.358 6	11.181 2	0.012 66

5 结束语

为了改善 COOT 存在的搜索能力较弱、收敛速度慢、精度不高等问题,提出一种基于窦性变异的改进人工蜂群白骨顶鸡算法(ICOOT),采用精英反向学习策略对白骨顶鸡种群进行初始化,增加了算法寻优多样性;提出一种全局最优解引导的人工蜂群搜索策略,以提高算法的搜索能力和收敛精度,最后引入窦性变异策略,避免 ICOOT 陷入局部最优值和提高最优个体质量。通过对 12 个基准测试函数进行仿真实验分析,结果证明提出的 ICOOT 具有很好的寻优性能。最后通过拉力/压力弹簧优化设计问题,进一步验证了 ICOOT 在实际工程应用时的有效性。下一步的研究工作是将 ICOOT 应用到多目标优化问题中。

参考文献:

- [1] KENNEDY J, EBERHART R. Particle swarm optimization [C]//Proceedings of ICNN'95 - international conference on neural networks. Perth:IEEE, 1995:1942-1948.
- [2] YANG X S, DEB S. Cuckoo search: recent advances and applications[J]. Neural Computing and Applications, 2014, 24: 169-174.
- [3] YANG X S, HE X. Firefly algorithm: recent advances and applications[J]. International Journal of Swarm Intelligence, 2013, 1(1): 36-50.
- [4] MIRJALILI S, LEWIS A. The whale optimization algorithm [J]. Advances in Engineering Software, 2016, 95: 51-67.
- [5] MIRJALILI S, MIRJALILI S M, Lewis A. Grey wolf optimizer [J]. Advances in Engineering Software, 2014, 69: 46-61.
- [6] WOLPERT D H, MACREARY W G. No free lunch theorems for optimization[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 67-82.
- [7] NARUEI I, KEYNIA F. A new optimization method based on COOT bird natural life model[J]. Expert Systems with Applications, 2021, 183: 115352.
- [8] HU G, ZHONG J, WANG X, et al. Multi-strategy assisted chaotic coot-inspired optimization algorithm for medical feature selection: a cervical cancer behavior risk study[J]. Computers in Biology and Medicine, 2022, 151: 106239.
- [9] WANG H Y, CHEN B, PAN D, et al. Optimal wind energy generation considering climatic variables by deep belief network (DBN) model based on modified coot optimization algorithm (MCOA) [J]. Sustainable Energy Technologies and Assessments, 2022, 53: 102744.
- [10] HOUSSEIN E H, HASHIM F A, FERAHTIA S, et al. Battery parameter identification strategy based on modified coot optimization algorithm[J]. Journal of Energy Storage, 2022, 46: 103848.
- [11] 张吉祥, 张孟健, 王德光. 增强型白骨顶鸡优化算法及其应用[J/OL]. 小型微型计算机系统: 1-10.
- [12] 李雪利, 杜逆索, 欧阳智, 等. 基于扰动因子和贪心策略的白骨顶优化算法[J]. 智能计算机与应用, 2023, 13(6): 38-49.
- [13] 贾鹤鸣, 李玉海, 文昌盛, 等. 改进白骨顶鸡优化算法的WSN网络覆盖[J]. 福建工程学院学报, 2022, 20(6): 561-566.
- [14] 周雪莹, 杜逆索, 欧阳智. 基于柯西变异和差分进化的混沌白骨顶鸟算法[J]. 计算机科学, 2023, 50(8): 209-220.
- [15] 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). Vienna: IEEE, 2005: 695-701.
- [16] GAO W, LIU S. A modified artificial bee colony algorithm[J]. Computers & Operations Research, 2012, 39(3): 687-697.
- [17] CHEN L, TIAN Y, MA Y. An improved grasshopper optimization algorithm based on dynamic dual elite learning and sinusoidal mutation[J]. Computing, 2022, 104: 981-1015.
- [18] HOLLAND J H. Genetic algorithms[J]. Scientific American, 1992, 267(1): 66-73.
- [19] RIZK-ALLAH R M, HASSANIEN A E. A comprehensive survey on the sine - cosine optimization algorithm[J]. Artificial Intelligence Review, 2023, 56(6): 4801-4858.
- [20] RASHEDI E, NEZAMABADI-POUR H, SARYAZDI S. GSA: a gravitational search algorithm [J]. Information Sciences, 2009, 179(13): 2232-2248.
- [21] KAUR S, AWASTHI L K, SANGAL A L, et al. Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization[J]. Engineering Applications of Artificial Intelligence, 2020, 90: 103541.