doi：10. 12068/j. issn. 1005－3026. 2026. 20250011

多策略融合的冠豪猪优化算法及其应用

朱 方[1,2]，唐 浩[3]

（1. 东北大学秦皇岛分校 计算机与通信工程学院，河北 秦皇岛 066004；2. 东北大学秦皇岛分校 河北省海洋感知网络与数据处理重点实验室，河北 秦皇岛 066004；3. 国网信阳供电公司，河南 信阳 464000）

摘 要：针对原始冠豪猪算法全局开发能力不足、易陷入局部最优等难题，提出一种基于多策略融合的冠豪猪优化算法(dimensional transformation and collective multi-strategy crested porcupine optimization，DCCPO). 首先，采用Sobol序列化策略初始化种群，并在种群收敛阶段引入周期性种群减少策略；在防御阶段，引入维度变换策略，以增强对最优种群周围空间的探索；此外，在视觉策略中引入Cauchy逆累计分布算子以避免陷入局部最优，而在身体攻击策略中引入自适应T分布和切向飞行算子，以平衡全局探索和局部开发. 为验证DCCPO算法的性能，本文使用CEC2017测试函数进行对比实验，并通过工程问题应用证明DCCPO相较其他算法具有更优的性能和适用性.

关 键 词：冠豪猪优化算法；Sobol序列化；切向飞行算子；自适应T分布；工程应用

中图分类号：TP 393 **文献标志码：**A **文章编号：**1005-3026(2026)04-0001-07

Improved Crested Porcupine Optimization Algorithm Based on Multi-strategy Fusion and Its Applications

ZHU Fang[1,2]，*TANG Hao*[3]

（1. School of Computer and Communication Engineering，Northeastern University at Qinhuangdao，Qinhuangdao 066004，China；2. Hebei Key Laboratory of Marine Perception Network and Data Processing，Northeastern University at Qinhuangdao，Qinhuangdao 066004，China；3. State Grid Xinyang Power Supply Company，Xinyang 464000，China. Corresponding author：ZHU Fang，E-mail：zhufang@neuq.edu.cn）

Abstract：To address the limitations of the original crested porcupine optimization（CPO）algorithm，such as insufficient global exploration capability and a tendency to fall into local optima，dimensional transformation and collective multi-strategy crested porcupine optimization，named DCCPO，is proposed. First，a Sobol sequence-based strategy was used to initialize the population，and a periodic population reduction strategy was introduced during the population convergence phase. In the defense phase，a dimensional transformation strategy was adopted to enhance the exploration of the space surrounding the optimal population. Additionally，a Cauchy inverse cumulative distribution operator was incorporated into the visual strategy to avoid local optima，while an adaptive T-distribution and tangential flight operator were introduced into the body attack strategy to balance global exploration and local exploitation. To validate the performance of the DCCPO algorithm，comparative experiments using the CEC2017 benchmark functions were conducted. Furthermore，the application of DCCPO to engineering problems demonstrated that it outperformed other algorithms in terms of both performance and applicability.

Key words：crested porcupine optimization algorithm；Sobol sequence；tangential flight operator；adaptive T-distribution；engineering application

优化问题渗透到现代科学和工程的广泛领域，证明其在不同领域的重要性和普遍性. 相较

收稿日期：2025-02-24
基金项目：2024年度河北省“三三三人才工程”资助项目（C2024001）.
作者简介：朱 方（1981—），男，河北秦皇岛人，东北大学副教授.
通信作者：朱 方，E-mail：zhufang@neuq.edu.cn.

传统基于梯度的算法和单纯形方法,元启发算法由于其灵活性、高稳定性和简单直观被广泛用于求解各类复杂优化问题.元启发算法通常分为基于个体和基于群体两类,群体智能优化算法通过从植物、动物、昆虫、人类等种群生存的社会行为中获得灵感,模仿自然界的集体行为.相较基于单个个体的元启发算法,基于群体的元启发算法具有较为广泛的搜索空间和更强的全局寻优能力,诸如鲸鱼优化算法(WOA)[1]、粒子群优化算法(PSO)[2]、灰狼优化算法(GWO)[3]、麻雀优化算法(SSA)[4]、蜣螂优化算法[5]等在连续和离散问题上均得到广泛应用.

以冠豪猪各种防御行为为灵感,Abdel-Basset等[6]提出冠豪猪优化算法(crested porcupine optimizer,CPO).CPO具有较快的求解速度,同时平衡全局搜索和局部开发,在大规模优化问题上表现出色.与GWO、WOA、SSA、人造大猩猩部队优化器(GTO)[7]和黏菌算法(SMA)[8]相比,在基准测试函数的评估、Wilcoxon秩和检验分析以及工程设计实际问题的解决过程中,CPO算法展现出了卓越的性能与突出的成效,不仅在数值计算中精准高效,在面对复杂工程场景时也能提供高质量的解决方案,充分证明了其在多领域应用的可靠性与优势.Fan等[9]使用CPO-LSTM模型进行光伏发电短期负荷预测,表明使用CPO算法优化LSTM网络参数,提高了模型的鲁棒性和泛化能力.Tian等[10]使用CPO计算了三次谐波并注入到逆变器的输入电压中,降低了B,C两相逆变器两相输出电压的总谐波含量.Zang等[11]构建CPO-BiTCN-BiLSTM-SA模型对交流流量进行预测,使用CPO对BiTCN-BiLSTM-SA模型进行优化,得到最佳的网络超参数,表明使用CPO优化后的模型交通预测性能明显提高.可以看出,CPO算法在一些优化问题上取得不错的效果,但是其种群多样性不足.同时,在第1种防御策略中,捕食者向冠豪猪靠近时,容易陷入局部最优.此外,在第4种防御策略中,当冠豪猪用羽毛对捕食者进行攻击时,个体之间缺乏信息交互且局部开发不足,表明CPO算法仍然存在较大的改进空间.

本文提出一种改进的冠豪猪优化算法(DCCPO):采用Sobol序列化策略初始化种群,并在种群收敛阶段,通过周期性减少种群规模来加速收敛,再通过周期性恢复种群规模,从而提高种群的多样性,扩大算法的搜索范围.第2防御阶段引入维度变换策略,以增强对最优种群周围空间的探索;在视觉策略中引入Cauchy逆累积分布算子避免陷入局部最优;在身体攻击策略中引入自适应T分布和切向飞行算子增强信息共享,平衡全局探索和局部开发,提高算法的求解精度.

1 改进冠豪猪优化算法

1.1 Sobol序列初始化种群

在求解复杂问题时,原始CPO算法通过随机生成位置的方式来初始化种群个体位置.这种方式存在明显不足,难以保障种群的多样性.而种群多样性对于算法性能至关重要,为了切实提升种群多样性,增强算法的全局寻优能力,使种群位置能够均匀分布于整个问题的解空间成为关键.在众多序列生成方法中,Sobol序列优势显著.其计算周期相对较短,采样速度更快,尤其在处理高维度序列时,展现出更高的效率,能够更高效地实现种群位置在解空间的均匀分布.基于此,本文采用Sobol序列对种群进行初始化,以期借助其优势克服原始CPO算法在初始化环节的弊端,提升算法整体性能,使其在复杂问题求解中发挥更出色.假定最优解的取值区间为$[x_{\min}, x_{\max}]$,Sobol序列所生成的随机数$K_n \in [0,1]$,那么种群的初始位置如式(1)所示:

$$x_n = x_{\min} + K_n \cdot (x_{\max} - x_{\min}). \tag{1}$$

1.2 Cauchy逆累积分布算子

若随机变量X的取值特性契合特定的概率密度函数形式,那么X就被认定为服从柯西分布.该柯西分布所对应的概率密度函数如式(2)所示:

$$f(x;a,b) = \frac{1}{\pi}\left[\frac{b}{(x-a)^2 + b^2}\right]. \tag{2}$$

在该分布中,参数a用于确定分布峰值所处的位置,故称为位置参数;参数b用于界定分布最大值一半处的半宽范围,被称作尺度参数.这种概率分布一般被叫做柯西分布,在物理学领域也常被称作洛伦兹分布或布赖特-维格纳分布.当a取值为0,b取值为1时,该分布被定义为标准柯西分布,其对应的概率密度函数如式(3)所示:

$$f(x;0,1) = \frac{1}{\pi}\left[\frac{1}{x^2+1}\right]. \tag{3}$$

其累积分布函数计算为

$$f(x;a,b) = \frac{1}{\pi}\arctan\left(\frac{x-a}{b}\right) + \frac{1}{2}. \tag{4}$$

当柯西分布累积分布函数能够以反函数形

式进行计算时，借助逆变换法便能够生成服从均匀分布的随机数．而柯西分布累积分布函数的反函数如式(5)所示：

$$f^{-1}(p;a,b)=a+b\tan\left(\pi\left(p-\frac{1}{2}\right)\right),\tag{5}$$

$$p=\mathrm{rand}\,n(1,d).\tag{6}$$

其中：p是在[0,1]范围内均匀分布的随机数；d代表函数的维数；且a取值为0，b取值为0.01．采用Cauchy逆累积分布算子，冠豪猪的视觉防御策略如式(7)所示：

$$\boldsymbol{x}_i^{t+1}=\boldsymbol{x}_i^t+\tau_1\times\left|2\times\tau_2\times\boldsymbol{x}_{\mathrm{CP}}^t-\boldsymbol{y}_i^t\right|\times0.01\times\tan\left(\pi\left(p-\frac{1}{2}\right)\right).\tag{7}$$

其中：$\boldsymbol{x}_{\mathrm{CP}}^t$是$t$时刻的最佳解；$\boldsymbol{y}_i^t$是在当前$\boldsymbol{x}_i$和从种群中随机选择的解之间生成的向量，表示迭代$t$时捕食者的位置；$\tau_1$是基于正态分布的随机数；$\tau_2$和$p$是在[0,1]范围内均匀分布的随机数．

1.3　自适应T分布

在CPO算法的迭代进程步入后期时，算法中的个体迅速向当前最优位置的周边聚集，种群的搜索能力大幅度减弱，容易陷入局部最优解，众多学者选择引入变异算子进行扰动，在变异算子中，T分布由于具有高斯变异和柯西变异两者的优点而被广泛使用．

$$p_t(x)=\frac{\Gamma\left(\frac{m+1}{2}\right)}{\sqrt{m\pi}\times\Gamma\left(\frac{m}{2}\right)}\times\left(1+\frac{x^2}{m}\right)^{-\frac{m+1}{2}},\quad -\infty<x<\infty;\tag{8}$$

$$\Gamma\left(\frac{m+1}{2}\right)=\int_0^{+\infty}x^{\frac{m+1}{2}-1}\mathrm{e}^{-x}\mathrm{d}x.\tag{9}$$

其中：m表示自由度的数量，数值越小，曲线的走势越趋于平坦．在T分布中，若m取值为1，此时该分布与柯西分布相同；当m趋向于无穷大时，T分布就近似于正态分布．

鉴于此，为提高算法从局部最优解中跳出的概率，本研究依据迭代次数提出一种自适应T分布方法，用于对种群进行扰动操作，其表达式如式(10)所示：

$$\mathrm{T}=\mathrm{T}\left(\frac{a^t}{\beta}\right)=\mathrm{T}\left(\frac{1}{\beta}\times\exp\left(\frac{\ln t_{\mathrm{c}}}{t_{\mathrm{c}}}\right)^t\right),\tag{10}$$

$$t_{\mathrm{c}}=\frac{\mathrm{iter}_{\max}}{5}.\tag{11}$$

其中：$\frac{a^t}{\beta}$表示自由度；t表示当前迭代次数；$\mathrm{iter}_{\max}$是最大迭代次数；β是比例因子，在本研究中其值设置为10.

1.4　切向飞行算子

切向飞行具有均衡开发和探索搜索的优点．为增强CPO算法的收敛能力，避免出现精度不足的问题，本研究引入切向飞行算子作为比例因子，以此对步长进行控制．切向飞行算子在形式上与柯西算子具有一定相似性，其数学表达式如式(12)所示：

$$f=\tan\left(v\frac{\pi}{2}\right),\tag{12}$$

$$v=\mathrm{rand}\,n(1,d).\tag{13}$$

在区间[0,1]上均匀分布的随机数用v表示；函数的维数用d表示．自适应T分布和切向飞行算子用于改进冠豪猪物理攻击策略，表达式如式(14)所示：

$$\left.\begin{aligned}\boldsymbol{x}_i^{t+1}=&\boldsymbol{x}_{\mathrm{CP}}^t+\left(\alpha\left(1-\tau_4\right)+\tau_4\right)\times\left(\delta\times\boldsymbol{x}_{\mathrm{CP}}^t-\boldsymbol{x}_i^t\right)\times\\&\mathrm{T}\left(\frac{1}{m}\times\exp\left(\frac{\ln t_{\mathrm{c}}}{t_{\mathrm{c}}}\right)^t\right)-\\&\tau_5\times\delta\times\gamma_t\times\boldsymbol{F}_i^t,\mathrm{rand}<0.5;\\\boldsymbol{x}_i^{t+1}=&\boldsymbol{x}_{\mathrm{CP}}^t+\left(\alpha\left(1-\tau_4\right)+\tau_4\right)\times\\&\left(\delta\times\boldsymbol{x}_{\mathrm{CP}}^t-\boldsymbol{x}_i^t\right)\times\tan\left(v\frac{\pi}{2}\right)-\\&\tau_5\times\delta\times\gamma_t\times\boldsymbol{F}_i^t,\mathrm{rand}\geqslant0.5.\end{aligned}\right\}\tag{14}$$

其中：α为收敛速度因子；δ为随机生成数，取值1或−1；τ_4，τ_5为区间[0,1]内的均匀分布随机值；$\boldsymbol{F}_i^t$为影响第i个捕食者$\boldsymbol{x}_i$的平均力．

1.5　维度变换策略

传统的改进策略对于最优种群周围空间的探索并不全面，从当前种群中随机生成一个解，往往会导致重新搜索空间中没有希望的区域，造成计算浪费．因此，Tizhoosh提出了反向学习策略，即引入一个随机解及其对应的反向解：$Y^{\mathrm{new}}=a_i+b_i-Y^{\mathrm{old}}$．实验结果表明，相较于随机解，反向候选解抵达全局最优解的概率往往更高．然而，当最优解的变化幅度极为微小时，这样会造成反向解出现同质化的情况，即对一个较差的解进行了多次尝试，造成重复计算，使得算法的寻优效能降低，进而引发过早收敛的现象．因此，本文提出一种新型的最优种群变异的策略(维度上升和维度降级)，假设最优种群中单独的个体为$x_i^1,x_i^2,\cdots,x_i^{\dim}$，$i=N_{\mathrm{best}},1,\cdots,N/3$，$N$为种群个数．

在高维度空间中难以定义旋转，因此本文提出维度降级的概念，对于单独的个体随机对维度的k个分支进行降级，假设降级为x_i^3，那么生成的子集X_{new}如式(15)所示：

$$x_i^1, x_i^2, x_i^4, \cdots, x_i^{\dim} = x_{\text{new},i}^1, x_{\text{new},i}^2, x_{\text{new},i}^3, \cdots, x_{\text{new},i}^{\dim-1}. \quad (15)$$

假设这个点为高维度空间中,圆心为0半径为R的超球面,如式(16)所示:

$$R = \sqrt{x_{\text{new},i}^1 + x_{\text{new},i}^2 + x_{\text{new},i}^3 + \cdots + x_{\text{new},i}^{\dim-1}}. \quad (16)$$

选择把超球面换为极坐标,转换公式如式(17)所示:

$$\left.\begin{aligned} x_{\text{new},i}^1 &= R \times \sin\sigma_1, \\ x_{\text{new},i}^2 &= R \times \sin\sigma_1 \times \cos\sigma_2, \\ &\vdots \\ x_{\text{new},i}^{\dim-2} &= R \times \sin\sigma_1 \times \sin\sigma_2 \times \cdots \times \cos\sigma_{\dim-2}, \\ x_{\text{new},i}^{\dim-1} &= R \times \sin\sigma_1 \times \sin\sigma_2 \times \cdots \times \sin\sigma_{\dim-2}. \end{aligned}\right\} \quad (17)$$

其中:角度范围$\sigma_1, \sigma_2, \cdots, \sigma_{\dim-3}$为$[0, \pi]$,$\sigma_{\dim-2}$为$[0, 2\pi]$.对于得到的$x_{\text{new},i}$,这里选择把降级的维度按照原值填充回来,即$x_{\text{new},i}^1, x_{\text{new},i}^2, x_i^3, x_{\text{new},i}^4, \cdots, x_{\text{new},i}^{\dim}$,这就是新的迭代点.

1.6 循环种群减少策略

原算法中提出了一种循环种群减少策略,通过周期性减少种群规模来加速收敛,再通过周期性恢复种群规模来提高多样性,避免陷入局部最优.为了优化种群规模的动态变化,对原有公式进行了改进,引入了正弦函数来实现非线性变化.新的公式如式(18)所示:

$$\text{Pop}_{\text{size}} = \text{fix}\left(N_{\min} + \left(N - N_{\min}\right) \times \left(1 - \left(\sin\left(\frac{\text{rem}\left(t, \frac{T_{\max}}{t}\right)}{\frac{T_{\max}}{T}} \times \pi - \frac{\pi}{2}\right) + 1\right) \times 0.5\right)\right). \quad (18)$$

其中:N表示初始种群的规模大小;$N_{\min}$代表新生成种群里个体数量的最小值;t表示当前的函数评估次数;$T_{\max}$是函数评估的最大次数;T是周期数.在保持原始方程的周期性和范围特征的同时,这个改进版本通过引入正弦函数实现了非线性变化.因此,种群规模在每个周期开始时逐渐减少,在中期阶段加速减少,并在周期结束时达到最大减少速度.

为了评估改进的周期性种群规模控制(ICPRC)策略在不同周期设置下的表现,本文对T=2和T=5两种情况进行了对比分析.图1和图2分别展示了这两种设置下ICPRC和常规周期性种群规模控制(CPRC)的种群规模变化曲线.

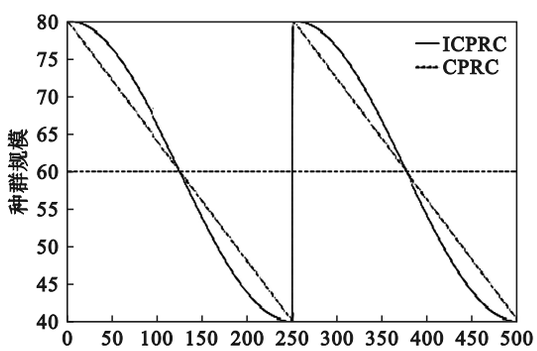


图1 T=2时种群下降曲线

Fig. 1 Population decline curves when T=2

1.7 DCCPO算法复杂度分析

采用大O符号法来分析DCCPO算法的时间复杂度.假定种群大小用N表示,优化问题的维度用符号D表示,最大迭代次数则用符号T表示.原始CPO算法的时间复杂度是$O(T\times D)$,改进后的DCCPO算法初始阶段的时间复杂度和原始CPO算法一致.Sobol序列初始化、引入柯西逆累积分布算子、自适应T分布以及切向飞行算子并未增加算法的时间复杂度.关于维度变换策略和改进的循环种群减少策略,前者重新定义了种群减少的方法,而没有产生额外的时间或空间成本,基于原始框架进行构建.但后者的时间和空间复杂度均为$O(D)$,涵盖了维度减少、极坐标表示和维度升级等步骤.在考虑整个优化过程时,假设进行T次迭代,种群大小为N,则总体复杂度为$O(T\times N\times D)$.这是因为每次迭代处理N个个体,每个个体具有D个维度.因此,改进的DCCPO算法保持与原始CPO算法相同的时间复杂度,即$O(T\times D)$.这表明DCCPO中实施的改进并未增加算法的计算复杂度,保持了其效率,同时提升了性能.算法流程图见图3.

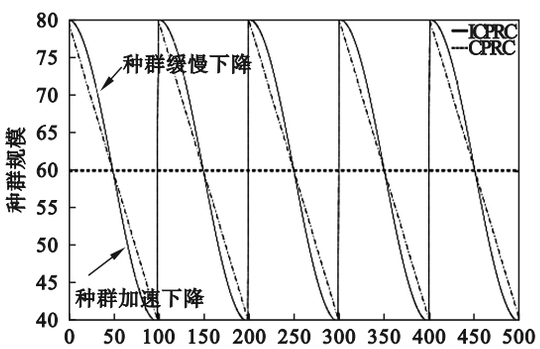


图2 T=5时种群下降曲线

Fig. 2 Population decline curves when T=5

2 实验结果与分析

2.1 实验设置

为了测试DCCPO算法的性能和优越性,本

文使用CEC2017[12]测试函数中30维的基准问题进行实验研究.CEC2017的分类情况详见表1.CEC2017测试函数主要涵盖单峰函数、简单多峰函数、混合函数以及合成函数这几类.单峰问题只存在一个全局最优解,一般可用来检验算法的开发能力.与单峰函数相比,简单多峰函数具有多个局部最优解,故而能够用于测试算法的寻优能力.而混合函数和合成函数由于自身结构复杂,可以更高效地检验算法的全局寻优能力,以及脱离局部最优的能力.

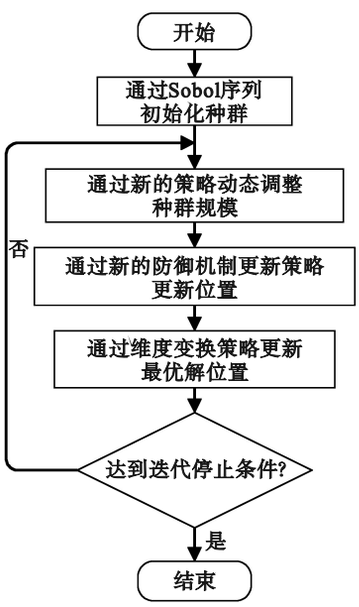


图3　DCCPO算法流程图

Fig. 3　DCCPO algorithm flowchart

表1　CEC2017分类

Table 1　CEC2017 classification

CEC2017 F_1 – CEC2017 F_3	单峰函数
CEC2017 F_4 – CEC2017 F_{10}	简单多峰函数
CEC2017 F_{11} – CEC2017 F_{20}	混合函数
CEC2017 F_{21} – CEC2017 F_{30}	合成函数

为了验证所提出DCCPO算法的有效性,本文选用冠豪猪算法(CPO)、灰狼优化算法(GWO)、鲸鱼优化算法(WOA)、正弦余弦算法(SCA)[13]、多元宇宙优化算法(MVO)[14]进行对比实验,各算法的参数设置与相关参考文献保持一致,见表2.本文将所有算法的初始种群规模设定为30,迭代次数上限为500,以保证实验的公正性.为降低实验结果的偶然性影响,本文选取的评价标准为:5种对比算法与DCCPO算法在各个测试函数上分别独立运行100次,以这些求解结果的均值和标准差作为评判依据.

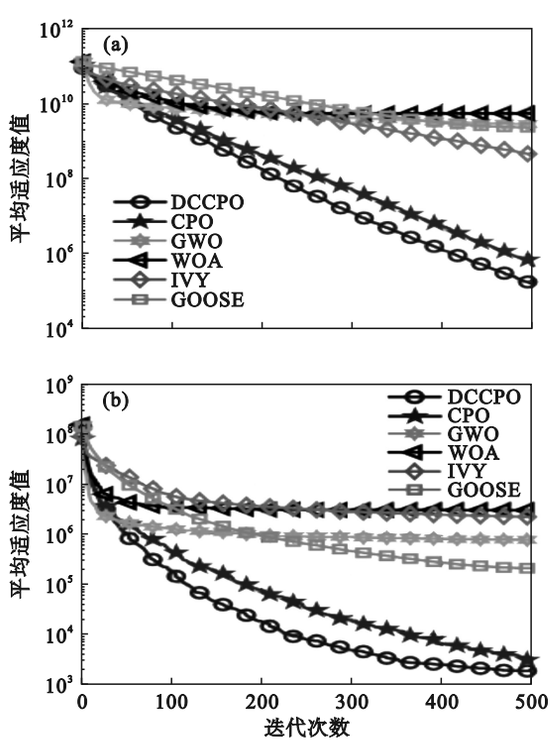
表2　参数设置

Table 2　Parameter settings

算法	参数	取值
SCA	a	2
WOA	a_1	(2,0)
	a_2	(2,1)
	b	1
GWO	a	(2,0)
	A	$[-a,a]$
	C	[0,2]
MVO	W_{max}	1
	W_{min}	0.2
CPO	α	0.2
	T_f	0.8

2.2　实验过程与结果分析

图4是6种算法在4个基准函数上的迭代曲线对比图,测试函数如表3所示.实验结果可以看出,DCCPO相较其他5种算法,收敛曲线比较平滑且收敛速度更快,其他算法的收敛曲线表现出不同程度的动荡,这表明DCCPO算法不仅在求解精度方面表现更为出色,而且在算法稳定性上也更具优势.综上所述,DCCPO算法在收敛速度和寻优精度上优势明显,显示了其在全局搜索及避免陷入局部最优方面的优异能力.



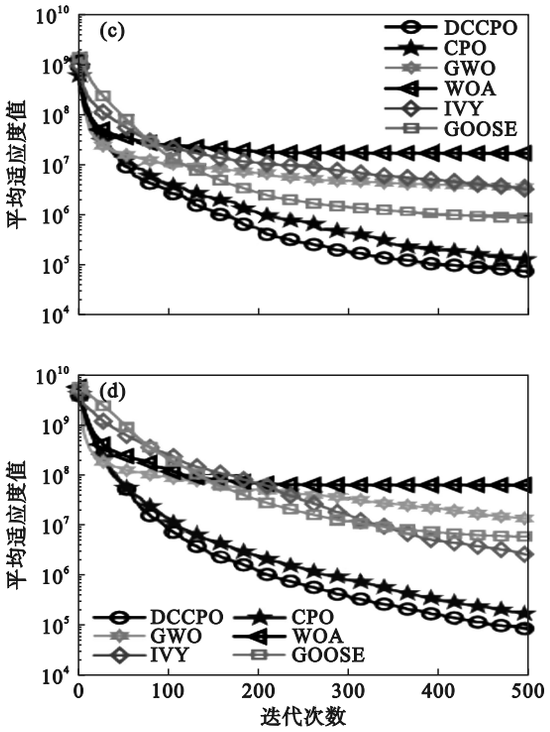


图4 6种算法迭代对比曲线

Fig.4 Iteration comparison curves of six algorithms

(a)—偏移旋转拉斯特里金函数;

(b)—偏移旋转鲁纳斯克双拉斯特里金函数;

(c)—混合函数4;(d)—组合函数4.

表3 测试函数

Table 3 Test functions

偏移旋转拉斯特里金函数(CEC2017 F10)
偏移旋转鲁纳斯克双拉斯特里金函数(CEC2017 F4)
混合函数4(N=4)(CEC2017 F14)
组合函数4(N=4)(CEC2017 F24)

2.3 应用分析

2.3.1 压力容器设计问题

压力容器设计问题旨在满足生产需求的前提下,尽可能削减其总费用$f(c)$,该问题包含4个设计变量,分别为壳体厚度$T_s(x_1)$、封头厚度$T_h(x_2)$、壳体半径$R(x_3)$和容器长度$L(x_4)$.

目标函数为

$$\min f(c)=0.6224x_1x_3x_4+1.7781x_2x_3^2+3.1661x_1^2x_4+19.84x_1^2x_3. \tag{19}$$

约束条件为

$$\left.\begin{aligned} g_1(c)&=-x_1+0.019\,3x_3\leqslant 0,\\ g_2(c)&=-x_3+0.009\,54x_3\leqslant 0,\\ g_3(c)&=-\pi x_3^2x_4-\frac{4}{3}\pi x_3^3+1\,296\,000\leqslant 0,\\ g_4(c)&=x_4-240\leqslant 0. \end{aligned}\right\} \tag{20}$$

其中:x_1,x_2,x_3,x_4为变量标量,取值范围为

$$0\leqslant x_1,x_2\leqslant 99;10\leqslant x_3,x_4\leqslant 200. \tag{21}$$

实验中种群规模设定为30,迭代次数设为500,让各算法独立运行100次,所得结果如表4所示.相较其他算法,DCCPO获得了最小成本5 885.33,表明DCCPO在求解压力容器工程问题时具有良好的实用性.

表4 压力容器问题实验结果

Table 4 Experimental results of pressure vessel problem

算法	x_1	x_2	x_3	x_4	$f(c)$
DCCPO	**0.778**	**0.385**	**40.32**	**199.80**	**5 885.33**
CPO	0.779	0.386	40.35	199.57	5 890.34
GWO	0.780	0.386	40.35	199.68	5 899.32
WOA	0.890	0.542	45.82	135.43	6 517.88
SCA	0.841	0.398	40.89	200	6 484.20
MVO	0.793	0.425	41.02	191.64	6 044.85

2.3.2 拉伸/压缩弹簧设计

拉伸/压缩弹簧设计的核心目标是实现弹簧制造成本的最小化,该弹簧具有3个结构参数,分别为钢丝直径$d(x_1)$、平均弹簧圈直径$D(x_2)$和有效弹簧圈数$N(x_3)$.图5为拉伸/压缩弹簧及其相关参数.



图5 拉伸/压缩弹簧设计结构图

Fig. 5 Spring tension/compression design structure diagram

目标函数为

$$\min f(c)=(N+2)Dd^2. \tag{22}$$

约束条件为

$$\left.\begin{aligned} g_1(c)&=1-\frac{D^3N}{71\,785d^4}\leqslant 0,\\ g_2(c)&=\frac{4D^2-dD}{12\,566(Dd^3-d^4)}+\frac{1}{5\,108d^2}-1\leqslant 0,\\ g_3(c)&=1-\frac{140.45d}{D^2N}\leqslant 0,\\ g_4(c)&=\frac{D+d}{1.5}-1\leqslant 0. \end{aligned}\right\} \tag{23}$$

边界约束为

$$0.05\leqslant x_1\leqslant 2,0.25\leqslant x_2\leqslant 1.3,2\leqslant x_3\leqslant 15. \tag{24}$$

种群规模设定为30,迭代次数设为500,各算法独立运行100次.所得结果如表5所示.可以看出,相较其他算法,DCCPO获得了最小成本0.012 663,表明DCCPO在求解拉伸/压缩弹簧工程问题时具有良好的实用性.

2.4　算法局限性讨论

尽管DCCPO在测试函数和工程设计问题上展现出优越性能，仍需指出其存在的局限性.首先，引入的多种策略（如Sobol序列、Cauchy逆累积分布、自适应T分布和维度变换）在提升性能的同时，也增加了算法的计算复杂度，在对实时性要求极高的场景中可能受限.其次，算法包含若干关键参数（如种群减少周期、自适应参数、维度变换条件等），虽然本文采用了经验设置，但其敏感性及在不同问题上的最优配置仍需进一步研究以获得更鲁棒的性能.最后，本文实验主要聚焦于连续单目标优化问题（以中低维度为主），对于超高维问题、离散优化、多目标优化等复杂类型，算法的有效性和适应性有待未来深入探索.

表5　拉伸/压缩弹簧问题实验结果

Table 5　Experimental results of tension/compression spring problem

算法	d	D	N	$f(c)$
DCCPO	**0. 051 668**	**0. 356 27**	**11. 312**	**0. 012 663**
CPO	0. 051 669	0. 356 22	11. 32	0. 012 667
GWO	0. 051 607	0. 354 73	11. 429	0. 012 687
WOA	0. 051 835	0. 360 25	11. 085	0. 012 666
SCA	0. 050	0. 317 27	14. 092	0. 012 763
MVO	0. 050	0. 314 32	14. 455	0. 012 931

3　结　　语

1）提出了一种新的循环种群减少策略，通过周期性减少种群规模来加速收敛，并应用Sobol序列化策略对种群进行初始化操作，增强种群的多样性.

2）在视觉策略中引入Cauchy逆累积分布算子避免陷入局部最优.

3）在身体攻击策略中引入自适应T分布和切向飞行算子增强信息共享，使全局探索与局部开发达到平衡状态，进而提升算法的求解精度.

4）为强化对最优种群周边空间的探索，引入了维度变换策略.

5）通过仿真实验，与其他5种算法进行对比，结果表明DCCPO算法在寻优能力、收敛速度、防止早熟等方面更具优势.针对压力容器设计和拉伸/压缩弹簧设计两个工程问题进行了仿真实验，实验结果验证了DCCPO算法的有效性与可靠性.

参考文献：

[1] Mirjalili S, Lewis A. The whale optimization algorithm[J]. *Advances in Engineering Software*, 2016, 95: 51-67.

[2] Kennedy J, Eberhart R. Particle swarm optimization[C]// Proceedings of ICNN'95-International Conference on Neural Networks. New York, 1995: 1942-1948.

[3] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer [J]. *Advances in Engineering Software*, 2014, 69: 46-61.

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

[5] Xue J K, Shen B. Dung beetle optimizer: a new meta-heuristic algorithm for global optimization[J]. *The Journal of Supercomputing*, 2023, 79(7): 7305-7336.

[6] Abdel-Basset M, Mohamed R, Abouhawwash M. Crested porcupine optimizer: a new nature-inspired metaheuristic [J]. *Knowledge-Based Systems*, 2024, 284: 111257.

[7] Abdollahzadeh B, Soleimanian Gharehchopogh F, Mirjalili S. Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems [J]. *International Journal of Intelligent Systems*, 2021, 36 (10): 5887-5958.

[8] Li S M, Chen H L, Wang M J, et al. Slime mould algorithm: a new method for stochastic optimization[J]. *Future Generation Computer Systems*, 2020, 111: 300-323.

[9] Fan Y L, Ma Z, Tang W W, et al. Using crested porcupine optimizer algorithm and CNN-LSTM-attention model combined with deep learning methods to enhance short-term power forecasting in PV generation[J]. *Energies*, 2024, 17 (14): 3435.

[10] Tian A, Zhang F H, Xiao P. Harmonic self-compensation control for bidirectional grid tied inverter based on crown porcupine optimization algorithm[J]. *Electronics*, 2024, 13 (13): 2607.

[11] Zang J, Cao B, Hong Y. Research on the fiber-to-the-room network traffic prediction method based on crested porcupine optimizer optimization [J]. *Applied Sciences*, 2024, 14(11): 4840.

[12] Onay F K, Aydemir S B. Chaotic hunger games search optimization algorithm for global optimization and engineering problems[J]. *Mathematics and Computers in Simulation*, 2022, 192: 514-536.

[13] Mirjalili S. SCA: a sine cosine algorithm for solving optimization problems [J]. *Knowledge-Based Systems*, 2016, 96: 120-133.

[14] Mirjalili S, Mirjalili S M, Hatamlou A. Multi-verse optimizer: a nature-inspired algorithm for global optimization [J]. *Neural Computing and Applications*, 2016, 27: 495-513.