

动态调整成长方式的郊狼优化算法及其应用

严逍亚¹, 王振雷¹, 王 昕²

(1. 华东理工大学 能源化工过程智能制造教育部重点实验室, 上海 200237;

2. 上海交通大学 电工与电子技术中心, 上海 200240)

摘要: 郊狼优化算法在迭代运行时种群多样性降低, 收敛速度变慢, 易陷入局部最优, 并且在求解约束优化问题时难以获得可行解。提出一种动态调整成长方式的郊狼优化算法(DGCOA)。在狼群进化中引入变异交叉策略, 增强种群多样性, 基于郊狼成长策略加入全局最优个体指导搜索, 使得每个子种群中的个体从不同的方向快速逼近最优解位置, 并根据种群中个体相似度对郊狼位置更新方式进行调整, 平衡算法的全局探索与局部开发能力。在求解约束优化问题时, 利用自适应约束处理方法构建新的适应度函数, 协调优化目标和约束违反度。基于CEC2006对22个测试函数和3个工程设计问题进行仿真, 结果表明, 与COA、ICTLBO、ODPSO等算法相比, DGCOA算法具有较高的收敛精度和稳定性, 适用于求解复杂优化问题。

关键词: 郊狼优化算法; 变异交叉; 约束处理; 测试函数; 工程优化

开放科学(资源服务)标志码(OSID):



中文引用格式: 严逍亚, 王振雷, 王昕. 动态调整成长方式的郊狼优化算法及其应用[J]. 计算机工程, 2022, 48(7): 73-81.

英文引用格式: YAN X Y, WANG Z L, WANG X. Coyote optimization algorithm with dynamically adjusting growth mode and its application[J]. Computer Engineering, 2022, 48(7): 73-81.

Coyote Optimization Algorithm with Dynamically Adjusting Growth Mode and Its Application

YAN Xiaoya¹, WANG Zhenlei¹, WANG Xin²

(1. Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China; 2. Center of Electrical & Electronic Technology, Shanghai Jiao Tong University, Shanghai 200240, China)

[Abstract] As the search continues, Coyote Optimization Algorithm(COA) has several disadvantages of the population reduced diversity, slow convergence speed, ease to fall into local optimum and unable to obtain a feasible solution when solving a constrained optimization problem. To overcome these disadvantages, Coyote Optimization Algorithm with Dynamically adjusting Growth mode(DGCOA) is proposed. First, the mutation crossover strategy is introduced to enhance the population diversity. Subsequently, the global optimal individual is added to guide the search so that individuals in each sub-population can quickly approach the optimal solution position from different directions, and based on the similarity degree of individuals in the population, the updating method of coyote growth was adjusted to effectively enhance the ability of balance global exploration and local development of the algorithm. When solving constrained optimization problems, a new fitness function is constructed by using the adaptive constraint processing method to coordinate the optimization objective and constraint violation degree. The simulation results of 22 test functions and 3 engineering design problems on CEC2006 reveal that the DGCOA algorithm has higher convergence accuracy and stability than COA, ICTLBO, and ODPSO algorithms. Consequently, DGCOA can solve complex optimization problems more effectively.

[Key words] Coyote Optimization Algorithm(COA); mutation crossover; constraint handling; test function; engineering optimization
DOI: 10. 19678/j. issn. 1000-3428. 0062074

0 概述

在科学研究和工程应用中亟待解决的优化问题越

来越多, 其多数为具有不确定性、多峰值、多模态特性的复杂优化问题。随着优化问题的复杂性增加, 传统的数学方法很难有效地处理此类复杂优化问题。此外,

基金项目: 国家重点研发计划(2018YFB1701103); 国家自然科学基金重大项目(61890930-3); 国家杰出青年科学基金(61725301); 中央高校基本科研业务费专项资金(222202117006)。

作者简介: 严逍亚(1997—), 女, 硕士研究生, 主研方向为智能优化算法; 王振雷, 教授、博士; 王 昕, 副教授、博士。

收稿日期: 2021-07-13 **修回日期:** 2021-09-01 **E-mail:** 18738979156@163.com

很多实际应用中的优化问题都需要在满足不同类型的约束条件下进一步追求解的质量,因为各种各样约束条件的存在,缩减了问题的可行区域,搜索过程变得复杂,这更增加了计算的复杂性和求解的难度。

对比经典的数学优化方法,智能算法具有结构简单、适用性强、参数较少、鲁棒性强等优点,这些进化算法往往同合适的约束处理方法相结合,从而更有效地处理约束方法,成为优化领域研究发展的趋势之一,如粒子群算法^[1-2]、差分进化算法^[3]、教学优化算法^[4-5]、人工蜂群算法^[6]、鲸鱼优化算法^[7]和水波优化算法^[8]等。进化算法与约束处理方法结合方式多样,如文献[4]针对算法寻优时出现的不同情况分别调用合适的约束处理方法,从而提高了对约束条件处理的有效性。文献[9]提出一种由学习和进化两个阶段构成的约束进化算法,学习阶段用于挖掘约束与目标函数之间的相关性,进化阶段利用相关性通过加权更新方法和存档替换机制进行约束优化。文献[10]设计一种两阶段约束进化方法,通过当前种群的状态调整适应度评价策略,使种群在一个阶段中跨越不可行区域到达可行区域,在另一个阶段沿着可行边界扩散,以自适应地平衡目标和约束条件。虽然许多群智能算法已在求解约束优化领域取得了较好的表现,但寻优精度、稳定性都有了进一步提升的空间。因此,设计寻优精度与稳定性更高的算法依然是研究的热点。

郊狼优化算法(Coyote Optimization Algorithm, COA)为PIEREZAN教授^[11]在2018年提出的一种模拟郊狼成长、生死等现象的新型群智能算法,其独特的算法结构为优化过程中探索与开发的平衡提供了新的机制,故不少学者对其进行研究^[12-13],并应用到电力系统和物联网等领域^[14-16]。文献[17]对基础COA算法进行了改进,利用全局最优指导最差郊狼更新,同时加入一种随机扰动操作来增加种群的多样性,使算法能更大范围地进行全局寻优。文献[18]提出一种相互作用的文化趋势,使其值受到子种群个体彼此间作用的影响,从而增强了算法的全局搜索能力,将改进后的郊狼优化算法应用到了医学图像增强问题上获得了满意的效果。文献[19]用组外贪心策略取代原COA算法的组内贪心策略,提高了算法的运行速度,同时通过郊狼间信息共享的方式增加了种群多样性。文献[20]将COA和文化算法联系起来,提出了文化郊狼算法,改善了COA算法在探测与开发之间的平衡。

虽然COA算法贡献了不同的算法结构和机制来实现优化过程中探索 and 开发两者之间的平衡^[15],但是在以下方面还有待提升:COA通过子种群内头狼和组文化趋势两者来指导组内郊狼社会位置更新,虽然增强了算法的开发能力,但在求解具有多个局部极值的优化问题时,子种群中头狼处于局部最优位置的概率大幅增加,由于头狼在搜索中的指导作用,子种群中其余郊狼很容易被头狼引导至其处于的局部最优位置,从而造成算法的早熟收敛;在COA算法中每次迭代时,子种群内郊狼在相同的头

狼和组文化趋势指导下一直保持着恒定的成长方式,随着迭代的进行,种群相似度变高,弱化了算法的搜索能力;子种群内郊狼的信息共享程度不高,在求解复杂优化问题时,全局搜索能力较弱。

本文提出一种动态调整成长方式的郊狼优化算法(Coyote Optimization Algorithm with Dynamically adjusting Growth mode, DGCOA)。通过引入变异交叉策略增强种群多样性,并提出一种新的郊狼成长更新方法,不仅加入全局最优个体来指导搜索,而且新的成长更新策略中根据种群中个体相似度的变化及时对郊狼成长方式进行调整,弥补基础COA不同个体间信息交互不足的问题,平衡算法全局探索和局部开发能力。将改进的COA算法与自适应约束处理方法相结合来协调优化目标和约束违反度,提高算法处理约束优化问题时的效率。

1 郊狼优化算法

1.1 种群初始化及随机分组

在郊狼算法中,解个体被随机分在 N_p 个子种群,每个子种群中包含 N_c 只个体。第 p 个子种群里第 c 只郊狼个体在第 j 维的初始位置可表示为:

$$\text{soc}_{c,j}^p = lb_j + r_j \cdot (ub_j - lb_j) \quad (1)$$

其中: lb_j 和 ub_j 分别代表第 j 维搜索空间的下限和上限; r_j 为 $[0,1]$ 内的一个随机数。

每只郊狼对环境的适应度可用目标函数值表示为:

$$\text{fit}_c^p = f(\text{soc}_c^p) \quad (2)$$

1.2 郊狼成长

郊狼成长受到所在子种群的头狼alpha和文化趋势 $\text{cult}_j^{p,t}$ 影响:

$$\alpha^{p,t} = \left\{ \text{soc}_c^{p,t} \mid \underset{c=\{1,2,\dots,N_c\}}{\text{argmin}} \text{fit}(\text{soc}_c^{p,t}) \right\} \quad (3)$$

$$\text{cult}_j^{p,t} = \begin{cases} \frac{O_{\frac{(N_c+1)}{2}-j}^{p,t}}{2}, & N_c \text{ 是奇数} \\ \frac{O_{\frac{N_c}{2}-j}^{p,t} + O_{\frac{(N_c+1)}{2}-j}^{p,t}}{2}, & \text{其他} \end{cases} \quad (4)$$

其中: $O^{p,t}$ 代表郊狼的社会位置排名。

两者的影响因子 δ_1 和 δ_2 可按下式计算:

$$\delta_1 = \alpha^{p,t} - \text{soc}_{cr_1}^{p,t} \quad (5)$$

$$\delta_2 = \text{cult}_j^{p,t} - \text{soc}_{cr_2}^{p,t} \quad (6)$$

其中: cr_1 和 cr_2 为在该子种群内任意选择的两只郊狼。

郊狼新的社会位置可按式(7)计算:

$$\text{new_soc}_c^{p,t} = \text{soc}_c^{p,t} + r_1 \cdot \delta_1 + r_2 \cdot \delta_2 \quad (7)$$

其中: r_1 和 r_2 为 $[0,1]$ 内的随机数。

COA算法择优选择对环境适应度更高的郊狼保存下来,该过程可表示为:

$$\text{soc}_c^{p,t+1} = \begin{cases} \text{new_soc}_c^{p,t+1}, & \text{new_fit}_c^{p,t} < \text{fit}_c^{p,t} \\ \text{soc}_c^{p,t}, & \text{new_fit}_c^{p,t} \geq \text{fit}_c^{p,t} \end{cases} \quad (8)$$

1.3 郊狼生死

COA算法用 $\text{age}_c^{p,t} \in N$ 来表示郊狼的年龄,结合郊狼位置信息和环境因素影响来产生新狼:

$$\text{pup}_c^{p,t} = \begin{cases} \text{soc}_{r_1,j}^{p,t}, & \text{rand}_j < P_s, j=j_1 \\ \text{soc}_{r_2,j}^{p,t}, & \text{rand}_j \geq P_s + P_a, j=j_2 \\ R_j, & \text{其他} \end{cases} \quad (9)$$

其中: r_1 和 r_2 为在第 p 个群落里任意选择的两匹郊狼; j_1 和 j_2 为任意选择的两个自变量维度; P_s 为离散概率; P_a 为关联概率; R_j 为第 j 维自变量取值范围内的任意数; $\text{rand}_j \in [0, 1]$ 是对应第 j 维的随机数。

P_s 和 P_a 可用公式表示为:

$$P_s = 1/D \quad (10)$$

$$P_a = (1 - P_s)/2 \quad (11)$$

其具体出生死亡选择过程可用算法1描述。其中, ω 代表第 p 个子种群内比新出生的小狼适应度低的郊狼, φ 代表这些郊狼的数目。

算法1 出生死亡选择行为伪代码

1. 计算 ω, φ
2. if $\varphi = 1$ then
3. 新出生的小狼存活, ω 中唯一的一只郊狼死亡
4. else if $\varphi > 1$ then
5. 新出生的小狼存活, ω 中年龄最大的郊狼死亡
6. else
7. 新出生的小狼存活
8. end if

某些郊狼有时会离开原始种群加入到另一个子种群里,此概率可用 P_e 来表示为:

$$P_e = 0.005 \cdot N_c^2 \quad (12)$$

2 改进的郊狼优化算法

2.1 变异交叉策略

在COA算法中,各个子种群的头狼和文化趋势引导着群体向潜力区域前进。但随着搜索的迭代进行,种群中的解个体会相互靠近,使得群体的多样性逐步下降。当所有解个体聚集到一起时,整个种群多样性不能得到更新,以致算法迭代有效性下降,搜索出现停滞。在这种情况下,可对解个体进行变异操作,在其附近位置产生新解,以此来增加种群多样性。因此,引入差分进化算法中的变异交叉策略到COA算法中,用于提高种群多样性,减小算法早熟收敛的可能,提高算法收敛精度。

首先选取第 c 只郊狼对应的社会位置 $\text{soc}_c^{p,t}$ 为父体向量,对应的变异向量可用公式表示为:

$$V_c^{p,t} = \text{soc}_c^{p,t} + F \cdot (\text{soc}_{r_1}^{p,t} - \text{soc}_{r_2}^{p,t}) \quad (13)$$

其中: r_1 和 r_2 为在第 p 个子群落里任意选择的两只郊狼; $\text{soc}_{r_1}^{p,t} - \text{soc}_{r_2}^{p,t}$ 为差分向量; F 是变异因子,用其对差分向量进行缩放,控制搜索步长,一般在 $[0, 2]$ 之内选择,通常取0.5,其值的大小决定了种群的分布情况。通过设置 F 在进化过程中的值,可以有效增加候选个体的数量,提高种群多样性,使种群在全局范围内进行搜索,增强算法的稳定性。

利用父体向量和变异向量通过离散杂交产生实验向量:

$$U_{c,j}^{p,t} = \begin{cases} V_{c,j}^{p,t}, & \text{rand}_j < CR, j=j_{\text{rand}} \\ \text{soc}_{c,j}^{p,t}, & \text{其他} \end{cases} \quad (14)$$

其中: $j=1, 2, \dots, D$; j_{rand} 是 $[1, D]$ 之间的随机整数; $CR \in [0, 1]$ 是交叉概率; $\text{rand}_j \in [0, 1]$ 是对应于第 j 维的随机数。

通过变异交叉策略产生 N_c 个实验向量构成实验种群 $U^{p,t}$ 。合并父体种群 $\text{soc}^{p,t}$ 与实验种群 $U^{p,t}$ 获得组合子种群 $h^{p,t} = \text{soc}^{p,t} \cup U^{p,t}$,按照下文提出自适应约束处理方法,从中选择 N_c 个较好个体形成更新的子种群 $\text{soc}^{p,t}$ 。

2.2 新的成长更新策略

2.2.1 全局最优影响因子的方式

在求解约束优化问题时,郊狼对环境的适应度 $\text{fit}_c^{p,t}$ 不但考虑目标函数值 $f(\text{soc}_c^{p,t})$,还要考虑到约束违反度 $G(\text{soc}_c^{p,t})$ 。考虑到约束违反度,当条件1或条件2满足时,说明郊狼 X_i 对环境的适应度优于郊狼 X_j ,记为 $\text{fit}(X_i) < \text{fit}(X_j)$ 。

$$1) G(X_i) = G(X_j) \wedge f(X_i) < f(X_j);$$

$$2) G(X_i) < G(X_j).$$

在所有的郊狼中,对环境适应度最高的郊狼即为在第 t 次迭代时整个种群的最优个体 G_{best}^t 。故在第 t 次迭代时,整个种群的最优个体 G_{best}^t 可用公式表示为:

$$G_{\text{best}}^t = \left\{ \text{soc}_c^{p,t} \mid \underset{p=\{1, 2, \dots, N_p\}}{\text{argmin}} \text{fit}_c^{p,t} \right\} \quad (15)$$

在COA算法中,每个子种群中的郊狼仅通过所在子种群的头狼 $\alpha^{p,t}$ 以及该子种群的文化趋势 $\text{cult}^{p,t}$ 两者来指导其社会位置的更新,并没有考虑到整个种群的最优位置 G_{best}^t 对郊狼位置更新的影响。因此,加入整个种群的最优个体 G_{best}^t 的方向信息来改进各个子种群中郊狼的位置更新方式。 G_{best}^t 对第 p 个子种群内的第 c 只郊狼在第 t 次迭代时位置更新的影响因子 δ_3 为:

$$\delta_3 = G_{\text{best}}^t - \text{soc}_c^{p,t} \quad (16)$$

在引入全局最优引导搜索后,各个子种群中的郊狼各自以更快的收敛速度从不同的方向朝着可行区域进行靠拢逼近,从而大幅加快了COA的收敛速度。

2.2.2 动态调整郊狼成长的方式

由式(3)可知,同一子种群内郊狼在相同的头狼和文化趋势与两头随机挑选的郊狼的差值共同指导下保持着恒定的成长机制,而在每次迭代过程中,同一子种群内头狼和文化趋势相同,且随机挑选的郊狼个体相似度无法保证,因此容易使得产生的新解相似度过高,以致于弱化算法的搜索能力,不易实现全局搜索。此外,郊狼在迭代过程中一直朝着所在子种群内头狼和文化趋势方向移动,增加了算法陷入局部最优的可能。

为此,本文根据子种群内郊狼个体的相似度动态调整郊狼成长更新方式,如式(17)所示:

$$\text{new_soc}_c^{p,t} = \begin{cases} \text{soc}_c^{p,t} + r_3 \cdot \delta_3 + r_1 \cdot \delta_1 + r_2 \cdot \delta_2, & R > P_r \\ \text{soc}_c^{p,t} + r_3 \cdot \delta_3 + r_4 \cdot (\text{soc}_{cr_3}^{p,t} - \text{soc}_{cr_4}^{p,t}), & R \leq P_r \end{cases} \quad (17)$$

其中: r_1, r_2, r_3 和 r_4 为 $[0, 1]$ 内的随机数; cr_3 和 cr_4 分别为在该子种群内任意选择的两只不同于 cr_1 和 cr_2 的郊狼; R 为子种群中郊狼相似度; P_r 为一常数。

计算方式如式(18)和式(19)所示:

$$\text{Count}_{\max} = \frac{N_c \times (N_c + 1)}{2} \quad (18)$$

$$R = \frac{\text{Count}}{\text{Count}_{\max}} \quad (19)$$

其中:Count为相似解的对数;Count_{max}为子种群内解的最大对数。通过两者的比值来确定种群相似度。

新的成长更新方式引入了全局最优解来指导位置更新,加快了算法的收敛速度。当相似度高时,根据方式1对郊狼进行位置更新,郊狼的进化由cult^{p,t}、alpha^{p,t}以及G_{best}^t三者共同确定的方向信息来进行指导。此时,由于子种群内郊狼相似度高,δ₁和δ₂的值较小,从而使得算法在全局最优解附近展开精细搜索,增强了算法的局部开发能力;当相似度低时,根据方式2对郊狼进行位置更新,郊狼的进化由G_{best}^t以及soc_{cr₃}^{p,t} - soc_{cr₄}^{p,t}共同确定的方向信息来进行指导。此时子种群内郊狼相似度低,子种群内郊狼的差值模型不仅使得子种群内信息得到共享,实现了子种群内郊狼的经验交流,而且由于soc_{cr₃}^{p,t} - soc_{cr₄}^{p,t}值较大,从而加大了算法的搜索步长,增强了算法的全局搜索能力。

算法2 新的成长更新策略伪代码

```
1. 初始化 Count = 0, 根据式(18)计算 Countmax
2. for i = 1: Nc - 1
3. for j = i + 1: Nc
4. if |f(soccr3p,t) - f(soccr4p,t)| < |f(cultp,t) - f(alphap,t)|
5. Count = Count + 1
6. end if
7. end for
8. end for
9. 根据式(19)计算种群相似度 R
10. if R > Pr
11. 根据方式1更新郊狼的社会位置
12. else
13. 根据方式2更新郊狼的社会位置
14. end if
```

在算法迭代前期进行随机全局搜索时,种群相似度较低,方式2以更大的概率指导郊狼的进化,全局搜索能力较强,算法可以凭借更高的概率找到全局最优解;在迭代后期进行局部开发时,种群相似度较高,方式1以更大的概率指导郊狼的进化,局部开发能力较强,提高了算法的收敛精度。随着迭代的进行,在新的成长更新策略指导下,算法根据种群的相似度选择合适的位置更新方式,有效地平衡了算法的全局搜索能力与局部开发能力,使得算法的优化性能得到提升。

2.3 边界处理方法

在算法寻优过程中,当新产生的解个体超出自变量上下限时,通常将其设为上下限值,但是目标函数的最优解往往不在搜索边界上,简单地将越界变量置于上下限上,会造成资源的浪费。故将越界分量按下式进行处理^[21],增强种群多样性,同时降低了下次迭代产生新位置越过寻优边界的几率。

$$x'_i = \begin{cases} ub_j - \min(x'_i - ub_j, ub_j - lb_j) \times r, & x'_i > ub_j \\ lb_j + \min(lb_j - x'_i, ub_j - lb_j) \times r, & x'_i < lb_j \end{cases} \quad (20)$$

其中:r为[0,1]内的随机数;ub_j、lb_j分别为x'_j的上限值和下限值。

2.4 约束处理方法

合理有效地对约束进行处理,平衡可行解与不可行解的关系是获得约束优化问题最优解的关键。考虑到在求解约束优化问题时,郊狼种群在寻优迭代中必定会经过不可行情形、半可行情形和可行情形3种状态,因此,本文在执行完上述新的成长更新策略之后,依据各个子种群在迭代过程中所处的情形,分别调用3种不同的约束处理技术来与改进后的郊狼优化算法相结合,实现目标函数和约束违反度两者的平衡,使得改进后的郊狼优化算法可以充分发挥其新的成长更新方式的搜索性能。

1)在不可行情形下,利用文献[22]提出约束处理技术把无约束优化问题构造为多目标优化问题,然后通过Pareto支配方法找出子代种群中的所有非支配个体,并分别随机替换一个父代种群中被它支配的个体(如果存在的话)。然后找出上述非支配个体中约束违反度最小的最优个体,若其没有替换父代种群中的个体,则使其随机替换父代种群中的某个个体,已达到利用最优个体持续地引导种群向可行区域靠近的目的。

2)在半可行情形下,利用一种适应值转换机制^[23]把约束优化问题构造为无约束优化问题,依据新构造的解适应度函数对个体进行选择,将携带重要信息的不可行个体和优秀的可行个体保留到下一代群体中。具体过程如下:

(1)将种群分成可行集合Z₁和不可行集合Z₂,对应解X_i的目标函数值f(X_i)按式(21)转换为:

$$f'(X_i) = \begin{cases} f(X_i), & i \in Z_1 \\ \max\{\phi \cdot f(X_{\text{best}}) + (1-\phi) \cdot f(X_{\text{worst}}), f(X_i)\}, & i \in Z_2 \end{cases} \quad (21)$$

其中:φ为可行解比例;X_{best}和X_{worst}分别为可行集合中最优及最差解。

(2)对转换得到的目标函数值和约束违反度按式(22)和式(23)进行标准化:

$$f_{\text{nor}}(X_i) = \frac{f'(X_i) - \min_{j \in Z_1 \cup Z_2} f'(X_j)}{\max_{j \in Z_1 \cup Z_2} f'(X_j) - \min_{j \in Z_1 \cup Z_2} f'(X_j)} \quad (22)$$

$$G_{\text{nor}}(X_i) = \begin{cases} 0, & i \in Z_1 \\ \frac{G(X_i) - \min_{j \in Z_2} G(X_j)}{\max_{j \in Z_2} G(X_j) - \min_{j \in Z_2} G(X_j)}, & i \in Z_2 \end{cases} \quad (23)$$

(3)构造出的适应度函数可表示为:

$$F(X_i) = f_{\text{nor}}(X_i) + G_{\text{nor}}(X_i) \quad (24)$$

可以看出在这种方法下,可行解依据目标函数值进行评价选择,不可行解同时考虑了目标函数值和约束违反度后进行评价选择,实现了目标函数和约束违反度之间的平衡。

3)在可行情形下,种群中全部是可行解,约束优化问题能够被当作无约束优化问题求解,利用目标函数对个体进行评价即可。

改进的郊狼优化算法根据寻优过程中种群在解

空间的不同状态,自主选择合适有效的约束处理方法,完成对约束优化问题求解。

2.5 算法实施步骤

将上述差分变异策略引入COA算法,并修改郊狼成长方式形成改进后的COA算法。结合自适应约束处理方法,DGCOA算法的实施步骤如算法3所示。DGCOA算法流程如图1所示。

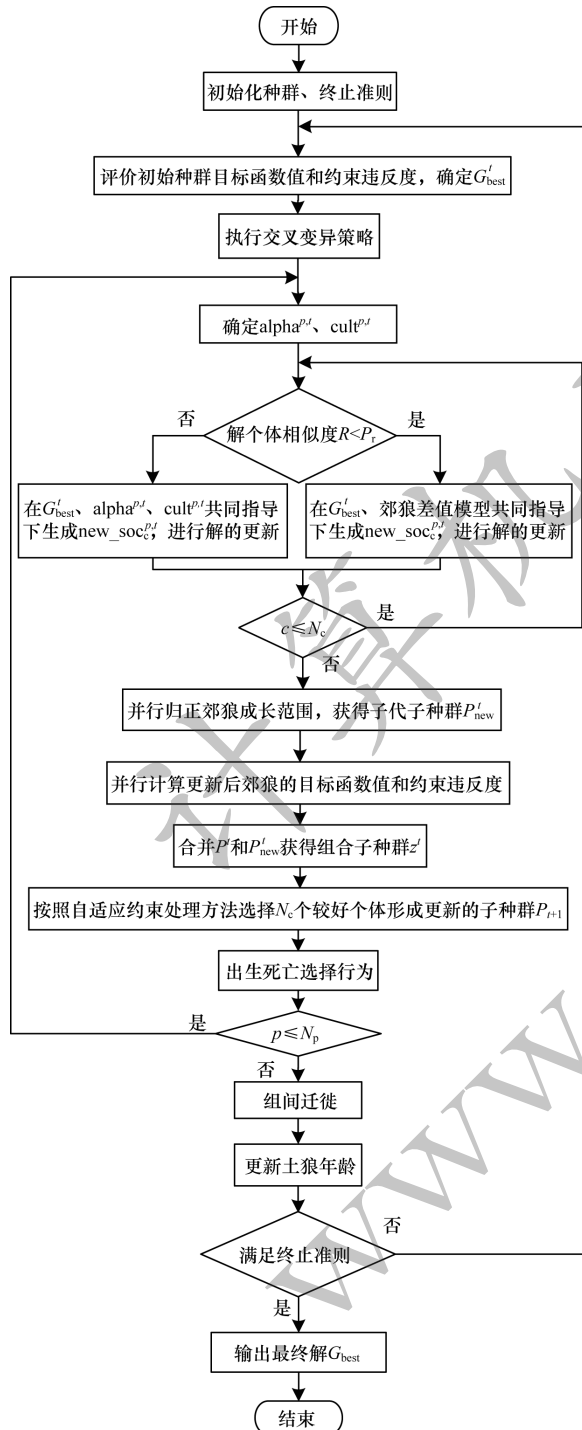


图1 DGCOA算法流程

Fig.1 DGCOA algorithm procedure

算法3 DGCOA算法伪代码

输入 郊狼种群初始参数 N_c, N_p 等

输出 全局最优解 G_{best}

1. 初始化种群参数,由式(1)获得 $N_c \times N_p$ 个初始解并随机分组

2. While(不满足终止条件)do

3. 计算种群中每个郊狼的 $f(\text{soc}_c^{p,t})$ 和 $G(\text{soc}_c^{p,t})$, 根据条件1和条件2及式(15)找出整个种群的全局最优解 G_{best}^t

4. for 每个子种群 p do

5. 执行变异交叉策略

6. 根据条件1和条件2及式(3)确定子种群 p 中的头狼 $\alpha^{p,t}$

7. 按照式(4)确定子种群 p 中的文化趋势 $\text{cult}^{p,t}$

8. 按照算法2更新郊狼的社会位置

9. 按照式(20)并行归正郊狼成长范围,获得对应的子代子种群 p_{new}^t

10. 并行计算更新后郊狼的目标函数值 $f(p_{new}^t)$ 和约束违反度 $G(p_{new}^t)$

11. 合并 p^t 和 p_{new}^t 获得组合子种群 z^t , 按照自适应约束处理方法从 z^t 中选择 N_c 个较好的个体形成新的子种群 p^{t+1}

12. end for

13. 按照式(9)和算法1模拟郊狼出生死亡选择行为

14. end for

15. 根据式(12)确定分散概率,进行组间迁徙

16. 更新郊狼的年龄 $\text{age}_c^{p,t} = \text{age}_c^{p,t} + 1$

17. 更新迭代次数 $t = t + 1$

18. end while

19. 选择整个种群中表现最优的郊狼,其社会位置即为全局最优解 G_{best}

改进的郊狼优化算法通过执行变异交叉策略和边界处理方法,增加了种群多样性,降低了算法早熟收敛现象发生的概率,提高了算法的稳定性;通过引入全局最优引导搜索,大幅加快了算法的收敛速度;通过种群中个体相似度来对郊狼成长更新方式进行动态选择,提高了算法的收敛精度。综上所述,改进后的郊狼优化算法无论是在寻优稳定性,还是在收敛精度和收敛速度方面都做出了改进,寻优能力具有很大竞争性。

3 数值实验与分析

3.1 参数设置

本文将 DGCOA 的寻优结果与 COA、ICTLBO^[4]、ODPSO^[2]、IWWO^[8]、E-BRM^[24]和 MGABC^[6]算法进行对比。其中,ICTLBO与本文DGCOA均是针对约束问题的分种群算法。DGCOA算法中的参数设置为: $N_p = 10$, $N_c = 14$; 变异交叉策略中的变异算子和交叉算子分别设置为: $F = 0.5$, $CR = 0.8$; 相似度阈值根据实验结果对比取最优值 $P_r = 0.3$; 等式约束容忍度 $\delta = 0.0001$; 最大函数评价次数(Max_NFEs)设定为240 000。COA中相关参数与DGCOA设置相同。

各算法独立运行25次,记录各算法运行结果的平均值Mean和标准差Std如表1所示。表中 $f(X^*)$ 为目前已知最优解 X^* 处的目标函数值,NF表示在全部运行次数内没有找到可行解,将在全部对比算法中取得最优的平均值和标准差加黑标注。部分算法在测试函数

上的收敛曲线对比如图2所示。其中选取的对比算法别取自于对应的参考文献。图2中ICTLBO的收敛曲线均为对应文献中表现最优的约束进化算法,其结果分

别取自于对应的参考文献。图2中ICTLBO的收敛曲线由其源代码运行所获得。

表1 不同算法在CEC2006测试函数上最终结果比较

Table 1 Comparison of final results of different algorithms on CEC2006 test function

最优解	指标	DGCOA	COA	ICTLBO	ODPSO	IWWO	E-BRM	MGABC
g01= -15.000 000 000 0	Mean	-15	-14.16	-15	-15	-15	-14.998 5	-13.553 54
	Std	0.00E+00	1.28E+00	0.00E+00	3.63E-16	0.00E+00	6.0E-04	1.64E+00
g02= -0.803 619 104 2	Mean	-0.767 338 805	-0.755 250 535	-0.799 622	-0.740 480 694	-0.678 2	-0.801 550	-0.789 062 9
	Std	2.88E-02	3.31E-02	5.17E-03	4.34E-02	1.11E-01	3.86E-03	1.20E-02
g03= -1.000 500 100 0	Mean	-1.000 500 1	-0.183 780 634	-1.000 5	-0.997 140 653	-1.000 5	-0.976 384	-1.000 383
	Std	4.67E-13	3.75E-01	1.97E-13	1.29E-02	0.00E+00	4.41E-02	4.11E-05
g04= -30 665.538 671 783 4	Mean	-306 65.538 67	-30 665.538 67	-30 665.539	-306 65.538 67	-30 665.538 7	-30 653.533	-30 665.54
	Std	3.71E-12	3.71E-12	7.43E-12	9.89E-06	0.00E+00	6.27E+00	1.05E-11
g05= 5 126.496 714 007 1	Mean	5 126.496 71	5 345.782 04	5 126.496 7	5 181.393 285	5 126.496 9	5 304.65	5 467.756
	Std	2.78E-12	3.11E+02	1.86E-12	7.47E+01	1.00E-04	1.89E+02	3.31E+02
g06= -6 961.813 875 580 2	Mean	-6 961.813 876	-6 961.813 876	-6 961.813 9	-6 961.813 876	-6 961.813 9	-6 961.735	-695 9.489
	Std	0.00E+00	0.00E+00	3.71E-12	0.00E+00	0.00E+00	5.10E-02	1.18E+00
g07= 24.306 209 068 1	Mean	24.306 209 068	24.306 567 27	24.306 2	24.334 879 41	24.601 2	26.463 11	24.780 64
	Std	3.86E-12	6.41E-04	5.40E-14	5.86E-02	4.14E-01	1.09E+00	3.12E-01
g08= -0.095 825 041 5	Mean	-0.095 825 041	-0.095 825 041	-0.095 825	-0.095 825 041	-0.095 8	-0.095 825	-0.095 825
	Std	1.42E-17	1.42E-17	0.00E+00	5.67E-18	0.00E+00	0.00E+00	0.00E+00
g09= 680.630 057 374 5	Mean	680.630 057 4	680.630 114	680.63	680.630 057 4	680.6301	681.054 1	680.630 9
	Std	8.71E-13	3.13E-05	4.64E-13	9.27E-13	0.00E+00	2.32E-01	5.13E-04
g10= 7 049.248 020 528 6	Mean	7 049.278 32	7 149.029 3	7 049.312 859	7 053.512 497	7 132.005 5	10 412.32	7 357.461
	Std	2.33E-02	2.61E+02	8.39E-02	1.19E+01	4.55E+01	3.11E+02	1.22E+02
g11= 0.749 900 000 0	Mean	0.749 9	0.989 904	0.7499	0.749 9	0.749 9	0.880 593	0.750 025
	Std	1.13E-16	5.00E-02	1.13E-16	1.13E-16	0.00E+00	7.95E-02	3.40E-05
g12= -1.000 000 000 0	Mean	-1	-1	-1	-1	-1	-1	-1
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
g13= 0.053 941 514 0	Mean	0.179 479 598	0.471 047 915	0.207 885 952	0.340 434 355	0.053 9	0.891 924 9	0.171 074
	Std	1.82E-01	1.86E-01	1.92E-01	2.36E-01	0.00E+00	3.50E-01	1.75E-01
g14= -47.764 888 459 5	Mean	-47.764 888 459 5	NF	-47.764 9	-47.757 801 8	-47.760 9	-42.950 8	-47.246 220
	Std	2.04E-12	NF	2.10E-08	1.25E-02	0.00E+00	1.37E+00	2.86E-01
g15= 961.715 022 289 9	Mean	961.715 02	961.762 714 9	961.715 02	961.746 474 5	961.715	965.022	962.173 70
	Std	5.80E-13	2.20E-01	4.64E-13	6.41E-02	0.00E+00	3.00E+00	7.77E-01
g16= -1.905 155 258 6	Mean	-1.905 155 259	-1.905 021 7	-1.905 155	-1.905 155 259	-1.905 1	-1.874 86	-1.905 155
	Std	4.53E-16	6.68E-04	2.79E-15	7.81E-16	0.00E+00	4.88E-02	0.00E+00
g17= 8 853.539 674 806 4	Mean	8 885.492	NF	8 880.595 253	8 923.934 753	8 853.537 1	NF	8 915.998
	Std	3.44E+01	NF	3.69E+01	7.26E+01	2.70E-03	NF	7.08E+01
g18= -0.866 025 403 8	Mean	-0.866 025	-0.866 023 675	-0.866 025 4	-0.827 792 699	-0.797 3	-0.618 16	-0.865 773 5
	Std	3.33E-16	3.05E-06	1.48E-13	7.80E-02	1.12E-01	1.26E-01	3.00E-04
g19= 32.655 592 950 2	Mean	32.655 592 95	33.933 193 73	32.657 019 94	33.216 633 82	57.391 8	219.391 2	NF
	Std	9.52E-13	3.75E+00	1.56E-03	1.19E+00	1.82E+00	6.99E+01	NF
g21= 193.724 510 070 0	Mean	230.398 4	NF	193.724 51	235.859 785 5	193.785 8	NF	NF
	Std	60.021 811 98	NF	6.00E-11	6.27E+01	1.21E-02	NF	NF
g23= -400.055 100 000 0	Mean	-388.822 878	135.999 732	-400.037 16	-311.737 659	-356.205 1	36.039 46	NF
	Std	1.15E+01	3.20E+02	7.44E-02	1.21E+02	9.59E+00	7.99E+01	NF
g24= -5.508 013 271 6	Mean	-5.508 013 272	-5.508 013 272	-5.508 013	-5.508 013 272	-5.508	-5.508 013 3	-5.508 013
	Std	9.06E-16	9.06E-16	5.66E-15	9.06E-16	0.00E+00	0.00E+00	1.78E-15

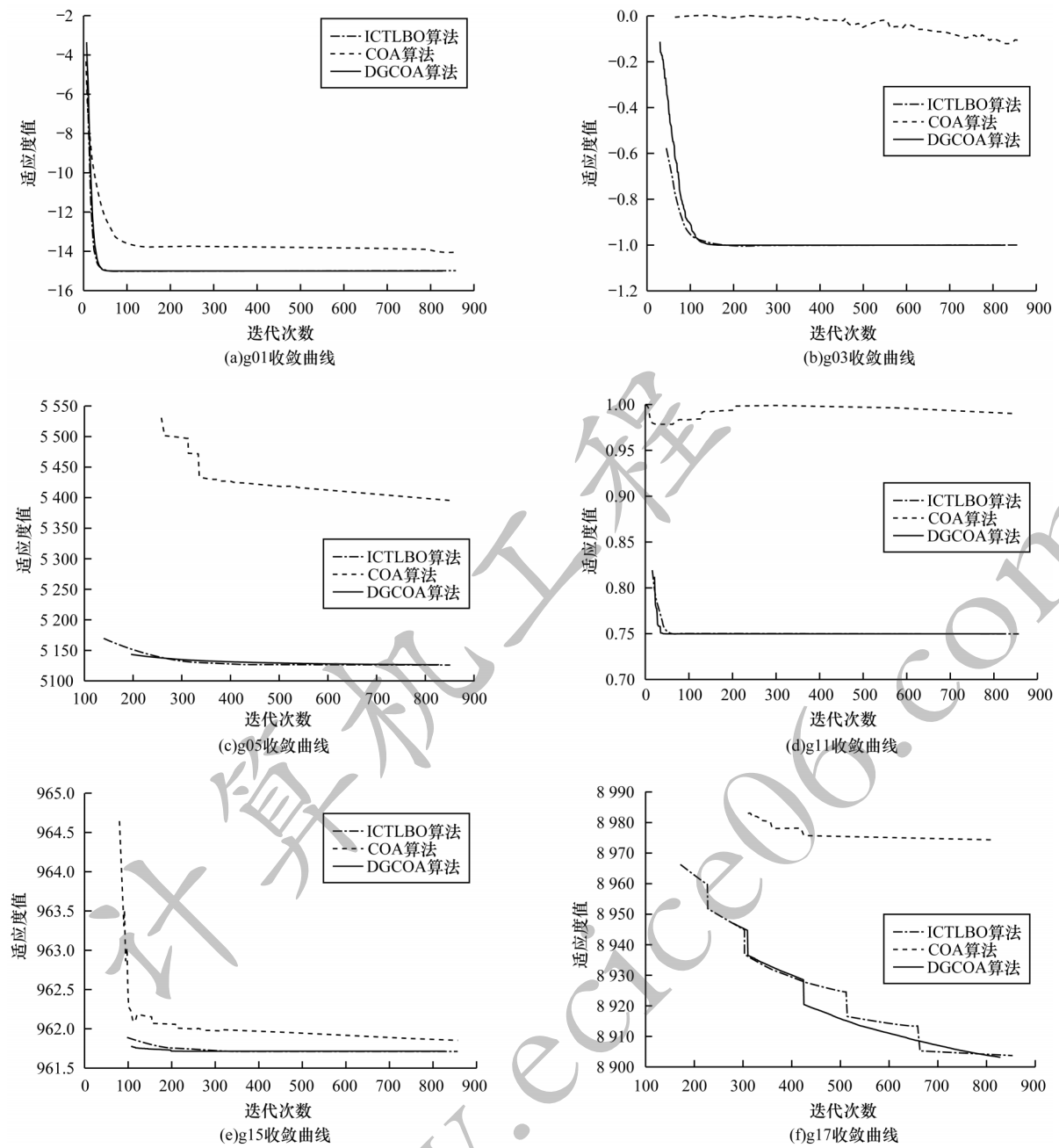


图2 DGCOA、COA与ICTLBO算法收敛曲线

Fig.2 Convergence curves of DGCOA,COA and ICTLBO algorithms

3.2 测试函数结果与分析

本文采用平均值(Mean)和标准差(Std)作为评价各算法优化性能的标准,平均值越小算法的收敛精度越高,标准差越小算法的稳定性越好。

由表1可以看出,在收敛精度方面,DGCOA在函数g02、g13、g17、g21和g23上获得了较高的收敛精度,分别取得了第4、第3、第3、第3和第2的排名。在其余17个函数上,DGCOA在7种对比算法中获得了最优的平均值,即收敛精度最高。特别地,在函数g10、g19上,仅有DGCOA算法能够以较高精度收敛至其最优值,获得全局最优解;在稳定性方面,DGCOA在函数g01、g04、g06、g10、g12、g18和g19上取得了最小的标准差,即稳定性最高。相较COA算

法仅在函数g06、g12和g24上可以稳定地获得最优解,对于函数g14、g17和g21无法获得可行解的情况,DGCOA无论是在收敛精度还是在稳定性方面都获得了很大的提高;相较其他6种算法,DGCOA在大部分测试函数上的收敛精度和稳定性也都能够获得具有高竞争力的结果。

由图2的算法收敛曲线可以看出,相较COA算法,在相同函数评估次数的前提下,对于图中所有测试函数,DGCOA算法均能以更快的收敛速度收敛到其所能找到的最优值附近,并具有更高的收敛精度。和同为分种群类智能优化算法的ICTLBO相比,在函数g01上,DGCOA收敛速度和收敛精度都具有较高竞争性;在函数g02和g05上,DGCOA在早期迭代

阶段收敛速度稍逊于 ICTLBO,但在后期,DGCOA 同样具有较高竞争性,与 ICTLBO 的收敛速度和收敛精度不相上下;在函数 g11 和 g15 上,DGCOA 早期迭代阶段收敛速度优于 ICTLBO,且在迭代后期能以较高的收敛精度收敛于最优解;在函数 g17 上,DGCOA 在早期迭代阶段收敛速度稍逊于 ICTLBO,没有很快找到可行解,但在后期,DGCOA 的收敛速度超过了 ICTLBO。

综上所述,图2进一步证明了改进 COA 算法无论是在收敛精度还是在收敛速度方面都得到了很大提升。

3.3 Wilcoxon 符号秩检验

本文采用 Wilcoxon 符号秩和检验统计方法^[19]来评价 DGCOA 与表1中其他算法两两之间差异的显著性。其中 $R+$ 表示正秩总和, $R-$ 表示负秩总和。对两种算法之间的统计比较结果采用符号“+”“-”“ \approx ”表示。“+”和“-”分别说明前一个算法明显优于或明显差于后2个算法;“ \approx ”说明2个算法没有明显差异。Wilcoxon 符号秩和检验结果如表2所示。

表2 Wilcoxon 符号秩和检验结果
Table 2 Wilcoxon signed-rank and test results

算法	$R+$	$R-$	结果
DGCOA -to- COA	184.5	5.5	+
DGCOA -to- ICTLBO	85.0	168.0	\approx
DGCOA -to- ODPSO	232.0	21.0	+
DGCOA -to- IWWO	172.0	81.0	\approx
DGCOA -to- E-BRM	200.5	9.5	+
DGCOA -to- MGABC	159.5	30.5	+

从表2可以看出,DGCOA 算法明显优于 COA、ODPSO、E-BRM 和 MGABC 算法,与 ICTLBO 和 IWWO 算法之间无明显差异。

4 工程优化问题结果及分析

为进一步检验 DGCOA 算法解决实际工程问题的有效性,选取3个较为经典的约束工程设计问题^[5]对 DGCOA 算法进行测试。

1) Welded Beam 问题

本文 DGCOA 算法运行结果和其他算法在 Welded Beam 问题上的比较如表3所示。可以看出 ICTLBO、IFOA 和 DGCOA 在每次运行中均能获得最优解,且 DGCOA 取得了最小的标准差,即 DGCOA 稳定性更好。

表3 不同算法在 Welded Beam 问题上的结果比较
Table 3 Comparison of results of different algorithms on the Welded Beam problem

算法	Best	Worst	Mean	Std
UABC ^[5]	1.724 852	—	1.724 853	1.70E-06
IFA ^[25]	1.724 894	1.725 321	1.724 978	1.33E-05
ICTLBO ^[4]	1.724 852	1.724 852	1.724 852	1.75E-11
IFOA ^[21]	1.724 852	1.724 852	1.724 852	1.11E-15
DGCOA	1.724 852	1.724 852	1.724 852	9.06E-16

2) Tension String Design 问题

本文 DGCOA 算法运行结果和其他算法在 Tension String Design 问题上的比较如表4所示。可以看出 IFOA 和 DGCOA 在每次运行中均能获得最优解,且 DGCOA 取得了最小的标准差,即 DGCOA 稳定性更好。

表4 不同算法在 Tension String Design 问题上的结果比较
Table 4 Comparison of results of different algorithms on Tension String Design problem

算法	Best	Worst	Mean	Std
CVI-PSO ^[4]	0.012 665 0	0.012 842 0	0.012 731 0	5.58E-05
UABC ^[5]	0.012 665 0	0.012 683 0	—	3.31E-05
ITLBO ^[26]	0.012 665 2	0.012 666 2	0.012 673 4	2.12E-06
IFOA ^[21]	0.012 665 2	0.012 665 2	0.012 665 2	1.13E-11
DGCOA	0.012 665 2	0.012 665 2	0.012 665 2	1.01E-16

3) Pressure Vessel 问题

本文 DGCOA 算法运行结果和其他算法在 Pressure Vessel 问题上比较如表5所示。可以看出 ICTLBO、COMDE 和 DGCOA 在每次运行中均能获得最优解,且 DGCOA 取得了最小的标准差,即 DGCOA 稳定性更好。

表5 不同算法在 Pressure Vessel 问题上的结果比较
Table 5 Comparison of results of different algorithms on the Pressure Vessel problem

算法	Best	Worst	Mean	Std
CVI-PSO ^[4]	6 059.714 300	6 820.410 100	6 292.123 100	2.88E+02
UABC ^[5]	6 059.714 335	6 192.116 200	—	2.04E+02
ICTLBO ^[4]	6 059.714 335	6 059.714 335	6 059.714 335	9.28E-13
COMDE ^[26]	6 059.714 335	6 059.714 335	6 059.714 335	3.62E-10
DGCOA	6 059.714 335	6 059.714 335	6 059.714 335	9.26E-13

5 结束语

本文提出一种动态调整成长方式的郊狼优化算法求解复杂优化问题。通过引入变异交叉策略增加种群多样性,协助算法获得全局最优解,并给出一种新的成长更新策略,加入全局最优影响因子,以提高算法的收敛速度。此外设计两种不同的成长方式,通过各个子种群中郊狼个体的相似度来对郊狼成长更新方式进行动态选择,有效平衡算法的全局探索与局部开发,同时弥补基础 COA 算法中子种群内郊狼个体信息共享不足的问题。最后将改进的郊狼算法与自适应约束处理技术相结合,实现对约束优化问题的有效求解。实验结果表明,相比 COA、ICTLBO、ODPSO 等算法,DGCOA 算法具有较高的收敛精度和稳定性。虽然本文算法在求解高维复杂约束优化问题时能够找到可行解,但是求解精度还有待提高。下一步将通过与更先进的方法结合来改进郊狼算法的寻优性能,提升其在求解高维优化问题时解的质量。

参考文献

- [1] SELVANAYAGAM C, DUONG P L T, WILKERSON B, et al. Inverse design for low warpage ultra-thin packages using constrained particle swarm optimization[J]. IEEE Access, 2021, 9: 64043-64053.
- [2] 梁静, 葛士磊, 瞿博阳, 等. 求解电力系统经济调度问题的改进粒子群优化算法[J]. 控制与决策, 2020, 35(8): 1813-1822.
LIANG J, GE S L, QU B Y, et al. Improved particle swarm optimization algorithm for solving power system economic dispatch problem[J]. Control and Decision, 2020, 35(8): 1813-1822. (in Chinese)
- [3] XU B, ZHANG Z H. Constrained optimization based on ensemble differential evolution and two-level-based epsilon method[J]. IEEE Access, 2020, 8: 213981-213997.
- [4] YU K J, WANG X, WANG Z L. Constrained optimization based on improved teaching-learning-based optimization algorithm[J]. Information Sciences, 2016, 352/353: 61-78.
- [5] 刘三阳, 靳安钊. 求解约束优化问题的协同进化教与学优化算法[J]. 自动化学报, 2018, 44(9): 1690-1697.
LIU S Y, JIN A Z. A co-evolutionary teaching-learning-based optimization algorithm for constrained optimization problems[J]. Acta Automatica Sinica, 2018, 44(9): 1690-1697. (in Chinese)
- [6] BANSAL J C, JOSHI S K, SHARMA H. Modified global best artificial bee colony for constrained optimization problems[J]. Computers & Electrical Engineering, 2018, 67: 365-382.
- [7] 张永, 陈锋. 一种改进的鲸鱼优化算法[J]. 计算机工程, 2018, 44(3): 208-213, 219.
ZHANG Y, CHEN F. A modified whale optimization algorithm[J]. Computer Engineering, 2018, 44(3): 208-213, 219. (in Chinese)
- [8] 顾启元, 王俊祥. 求解约束优化问题改进的水波优化算法[J]. 计算机工程与设计, 2020, 41(5): 1320-1326.
GU Q Y, WANG J X. Improved water wave optimization for constrained optimization problems[J]. Computer Engineering and Design, 2020, 41(5): 1320-1326. (in Chinese)
- [9] WANG Y, LI J P, XUE X H, et al. Utilizing the correlation between constraints and objective function for constrained evolutionary optimization [J]. IEEE Transactions on Evolutionary Computation, 2020, 24(1): 29-43.
- [10] TIAN Y, ZHANG Y, SU Y, et al. Balancing objective optimization and constraint satisfaction in constrained evolutionary multi-objective optimization [J]. IEEE Transactions on Cybernetics, 2021, 23(5): 2230-2241.
- [11] PIEREZAN J, DOS SANTOS COELHO L. Coyote optimization algorithm: a new metaheuristic for global optimization problems[C]//Proceedings of IEEE Congress on Evolutionary Computation. Washington D. C., USA: IEEE Press, 2018: 1-8.
- [12] MOSTAFA H H, IBRAHIM A M. Performance investigation for tracking GMPP of photovoltaic system under partial shading condition using coyote algorithm[C]//Proceedings of the 21st International Middle East Power Systems Conference. Washington D. C., USA: IEEE Press, 2019: 34-40.
- [13] FATHY A, AL-DHAIFALLAH M, REZK H. Recent coyote algorithm-based energy management strategy for enhancing fuel economy of hybrid FC/battery/SC system[J]. IEEE Access, 2019, 7: 179409-179419.
- [14] ABDELGHAFAR S, GODA E, DARWISH A, et al. Satellite lithium-ion battery remaining useful life estimation by coyote optimization algorithm[C]//Proceedings of the 9th International Conference on Intelligent Computing and Information Systems. Washington D. C., USA: IEEE Press, 2019: 124-129.
- [15] CARBALLAL A, IVÁN PAZOS-PÉREZ R, RODRIGUEZ-FERNANDEZ N, et al. A point-based redesign algorithm for designing geometrically complex surfaces [J]. IET Image Processing, 2020, 14(12): 2948-2956.
- [16] MOHAMED A, SABER W, ELNAHRY I, et al. Coyote optimization based on a fuzzy logic algorithm for energy-efficiency in wireless sensor networks[J]. IEEE Access, 2020, 8: 185816-185829.
- [17] 张新明, 王豆豆, 陈海燕, 等. 强化最优和最差狼的郊狼优化算法及其二次指派问题应用[J]. 计算机应用, 2019, 39(10): 2985-2991.
ZHANG X M, WANG D D, CHEN H Y, et al. Best and worst coyotes strengthened coyote optimization algorithm and its application to quadratic assignment problem [J]. Journal of Computer Applications, 2019, 39(10): 2985-2991. (in Chinese)
- [18] 张新明, 付子豪, 陈海燕, 等. 全局引导和相互作用的郊狼优化算法及其应用[J]. 计算机应用研究, 2020, 37(9): 2711-2717.
ZHANG X M, FU Z H, CHEN H Y, et al. Coyote optimization algorithm with global guidance and coyote interaction and its application [J]. Application Research of Computers, 2020, 37(9): 2711-2717. (in Chinese)
- [19] 张新明, 李双倩, 刘艳, 等. 信息共享模型和组外贪心策略的郊狼优化算法[J]. 计算机学报, 2020, 47(5): 217-224.
ZHANG X M, LI S Q, LIU Y, et al. Coyote optimization algorithm based on information sharing and static greed selection [J]. Computer Science, 2020, 47(5): 217-224. (in Chinese)
- [20] PIEREZAN J, MAIDL G, MASSASHI YAMAO E, et al. Cultural coyote optimization algorithm applied to a heavy duty gas turbine operation [J]. Energy Conversion and Management, 2019, 199: 111932.
- [21] 石建平, 李培生, 刘国平, 等. 求解约束优化问题的改进果蝇优化算法及其工程应用[J]. 控制与决策, 2021, 36(2): 314-324.
SHI J P, LI P S, LIU G P, et al. Improved fruit fly optimization algorithm for solving constrained optimization problems and engineering applications [J]. Control and Decision, 2021, 36(2): 314-324. (in Chinese)
- [22] WANG Y, CAI Z X. A dynamic hybrid framework for constrained evolutionary optimization [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2012, 42(1): 203-217.
- [23] JIA G B, WANG Y, CAI Z X, et al. An improved ($\mu + \lambda$)-constrained differential evolution for constrained optimization [J]. Information Sciences, 2013, 222: 302-322.
- [24] COELLO C A C. Constraint-handling techniques used with evolutionary algorithms [J]. Applied Soft Computing, 2018, 72(11): 14-29.
- [25] 龙文, 蔡绍洪, 焦建军, 等. 求解约束优化问题的萤火虫算法及其工程应用[J]. 中南大学学报(自然科学版), 2015, 46(4): 1260-1267.
LONG W, CAI S H, JIAO J J, et al. Firefly algorithm for solving constrained optimization problems and engineering applications [J]. Journal of Central South University (Science and Technology), 2015, 46(4): 1260-1267. (in Chinese)
- [26] YU K J, WANG X, WANG Z L. An improved teaching-learning-based optimization algorithm for numerical and engineering optimization problems [J]. Journal of Intelligent Manufacturing, 2016, 27(4): 831-843.