

基于混沌搜索策略的鲸鱼优化算法

王坚浩^{1†}, 张亮¹, 史超¹, 车飞¹, 丁刚¹, 武杰²

(1. 空军工程大学 装备管理与无人机工程学院, 西安 710051; 2. 中国人民解放军 94402 部队, 济南 250002)

摘要: 针对鲸鱼优化算法存在探索和开发能力难以协调、易陷入局部最优的不足, 提出一种基于混沌搜索策略的鲸鱼优化算法(CWOA)。首先, 采用混沌反向学习策略产生初始种群, 为全局搜索多样性奠定基础; 其次, 设计收敛因子和惯性权重的非线性混沌扰动协同更新策略以平衡全局探索和局部开发能力; 最后, 将种群进化更新与最优个体的混沌搜索机制相结合, 以减小算法陷入局部最优的概率。对 10 个基准测试函数和 6 个复合测试函数进行优化, 实验结果表明, CWOA 在收敛速度、收敛精度、鲁棒性方面均较对比算法有较大提升。

关键词: 鲸鱼优化算法; 混沌搜索; 反向学习; 收敛因子; 惯性权重

中图分类号: TP18; TP301

文献标志码: A

Whale optimization algorithm based on chaotic search strategy

WANG Jian-hao^{1†}, ZHANG Liang¹, SHI Chao¹, CHE Fei¹, DING Gang¹, WU Jie²

(1. Equipment Management and Unmanned Aerial Vehicles Engineering College, Air Force Engineering University, Xi'an 710051, China; 2. PLA 94402 Troop, Ji'nan 250002, China)

Abstract: A whale optimization algorithm based on the chaotic search strategy (CWOA) is proposed to overcome the drawbacks of being difficult to coordinate the exploration and exploitation ability, and easily trapped into local optimum. In the proposed algorithm, the chaotic opposition-based learning strategy is used to generate initial population, which strengthens the diversity of population in the global searching process. Then, a nonlinearly chaotic disturbance cooperative updating strategy for the convergence factor and inertia weight is designed to balance the exploration and exploitation ability. Finally, the chaotic search strategy for optimum individual is combined with evolutionary population updating to avoid the possibility of being trapped into local optimum. The optimization experiments are conducted on the 10 benchmark functions and 6 composite functions. Simulation results show that the proposed CWOA has fast convergence and more precise convergence than other comparison algorithms.

Keywords: whale optimization algorithm; chaotic search; opposition-based learning; convergence factor; inertia weight

0 引言

优化问题一直是计算机科学、人工智能和管理决策等领域的研究热点。近年来, 以模拟自然界动物群体行为的粒子群算法(PSO)^[1-2]、蚁群算法(ACO)^[3]和蜂群算法(ABC)^[4]等为代表的仿生群体智能优化算法为求解复杂优化问题提供了强有力的工具, 并不断发展壮大, 例如, 模拟蝙蝠利用超声波捕猎行为的蝙蝠算法(BA)^[5]、模拟萤火虫发光行为的萤火虫算法(FA)^[6]、模拟果蝇觅食行为的果蝇算法(FOA)^[7]、模拟灰狼群体捕猎行为和分工等级机制的灰狼算法(GWO)^[8]、模拟蜻蜓觅食和迁移行为的蜻蜓算法(DA)^[9]等。

2016年, 文献[10]仿生了自然界中座头鲸群体狩

猎行为, 并抽象出搜索、包围、攻击 3 种行为, 提出了一种新型仿生群体智能优化算法——鲸鱼优化算法(WOA)。WOA 算法具有原理简单、参数设置少、寻优性能强等特点, 在函数优化方面, 已被证明在求解精度和收敛速度性能上均优于 PSO 算法, 目前已成功应用于大规模优化问题^[11]、神经网络训练^[12]等领域。然而, 基本 WOA 算法仍然存在一些不足, 诸如控制参数调整策略不能完全体现实际优化搜索过程导致算法全局探索和局部开发能力难以协调, 迭代后期群体中所有个体均向最优个体聚集导致群体多样性缺失易使算法陷入局部最优等。

针对 WOA 算法的局限性, 本文提出基于混沌搜索策略的鲸鱼优化算法(CWOA), CWOA 算法采用混

收稿日期: 2018-01-18; 修回日期: 2018-07-21.

基金项目: 国家自然科学基金项目(61503409).

责任编辑: 夏元清.

[†]通讯作者. E-mail: hamilton_wang@sina.com.

沌反向学习策略初始化种群,增强种群的多样性并提高求解效率;受PSO算法启发,提出收敛因子和惯性权重非线性混沌扰动协同更新策略,以协调算法全局探索和局部开发能力;通过对当前最优个体采用具有较好遍历性和随机性的混沌搜索,减小算法陷入局部最优的概率.最后通过10个基准测试函数和6个复合测试函数优化实验验证CWOA算法的有效性和优越性.

1 鲸鱼优化算法

鲸鱼被认为是世界上最大的哺乳动物,成年鲸鱼可长到30米长、180吨重.鲸鱼拥有次声波/超声波听觉,依靠独特的回声定位探寻周围猎物或相互传递信息.在鲸鱼家族中,虎鲸等齿鲸类鲸鱼善于捕猎,而座头鲸等体形巨大的须鲸类鲸鱼,由于缺乏可以咀嚼的牙齿,只能偏向于捕食成群的小型鱼虾,因此进化出一种特殊的觅食行为,即Bubble-net觅食,称为泡泡网觅食或气幕觅食,如图1所示.

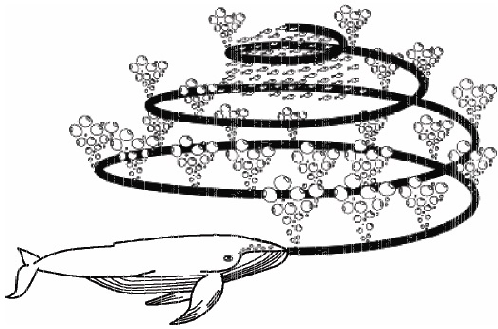


图1 座头鲸的Bubble-net觅食行为

根据座头鲸群体狩猎行为特点,WOA算法抽象出包围猎物、Bubble-net攻击和随机搜索3种行为.

1) 包围猎物.

每只鲸鱼代表一个个体,每个个体在搜索空间的位置代表一个解.鲸鱼能够通过回声定位识别猎物位置并包围猎物,鲸鱼位置更新公式如下:

$$X^{t+1} = X_{\text{gbest}}^t - A \cdot |C \cdot X_{\text{gbest}}^t - X^t|. \quad (1)$$

其中: t 为当前的迭代次数; X^t 为当前的位置向量; $X_{\text{gbest}}^t = (X_{\text{gbest}1}^t, X_{\text{gbest}2}^t, \dots, X_{\text{gbest}D}^t)$ 为当前全局最优位置向量, D 为向量维数; $A \cdot |C \cdot X_{\text{gbest}}^t - X^t|$ 为包围步长,系数向量 A 和 C 定义如下:

$$A = 2a \cdot \text{rand}_1 - a, \quad (2)$$

$$C = 2 \cdot \text{rand}_2. \quad (3)$$

其中: rand_1 和 rand_2 为 $[0, 1]$ 范围内均匀分布产生的随机数; a 为收敛因子,随迭代次数 t 从2线性减小到0,即

$$a = 2 - 2t/t_{\text{max}}, \quad (4)$$

这里 t_{max} 为最大迭代次数.

2) Bubble-net攻击.

从图(1)可以看出,鲸鱼是以向上螺旋式运动并不断收缩包围圈来攻击猎物的,为了从数学上描述Bubble-net觅食行为,设计收缩包围机制和螺旋更新位置两种方法.

收缩包围机制通过式(2)随着收敛因子 a 的减小实现,同时系数向量 A 的波动范围也随着收敛因子 a 减小,即当收敛因子 a 在迭代过程中从2减小到0时,系数向量 A 的波动范围为 $[-a, a]$.当系数向量 A 为 $[-1, 1]$ 内的随机值时,鲸鱼 $t+1$ 时刻的位置可以是 t 时刻的位置与 t 时刻全局最优位置之间的任意位置,即表示鲸鱼始终在收缩包围圈内游动.

在螺旋更新位置方法中,鲸鱼以螺旋运动向猎物游动,其数学模型为

$$X^{t+1} = X_{\text{gbest}}^t + D \cdot e^{bl} \cdot \cos(2\pi l). \quad (5)$$

其中: $D = |X_{\text{gbest}}^t - X^t|$ 为鲸鱼与当前全局最优个体之间的距离, b 为限定对数螺旋形状的常数, l 为 $[-1, 1]$ 之间的随机数.

鲸鱼在猎物收缩圈周围内同步沿着螺旋形路径游动,为了模拟这种同步行为,在优化过程中假设选择收缩包围机制和螺旋更新位置概率均为0.5,同步行为的数学模型为

$$X^{t+1} = \begin{cases} X_{\text{gbest}}^t - A \cdot |C \cdot X_{\text{gbest}}^t - X^t|, & p < 0.5; \\ X_{\text{gbest}}^t + D \cdot e^{bl} \cdot \cos(2\pi l), & p \geq 0.5. \end{cases} \quad (6)$$

其中 p 为 $[0, 1]$ 范围内均匀分布产生的随机数.

3) 随机搜索.

当系数向量 $|A| > 1$ 时,表示鲸鱼在收缩包围圈外游动,此时鲸鱼个体根据彼此位置进行随机搜索,其数学模型为

$$X^{t+1} = X_{\text{rand}}^t - A \cdot |C \cdot X_{\text{rand}}^t - X^t|, \quad (7)$$

其中 X_{rand}^t 为随机选择的鲸鱼个体位置向量.

由上述分析可知,WOA算法主要参数为系数向量 A 和 C ,其中参数 A 对于协调WOA算法全局探索和局部开发能力至关重要.当 $|A| > 1$ 时,鲸鱼种群将扩大搜索范围,以找到更好的候选解,即为WOA算法的全局探索能力;当 $|A| \leq 1$ 时,鲸鱼种群将缩小搜索范围,在局部区域进行精细搜索,即为WOA算法的局部开发能力.而参数 A 的值在很大程度上依赖于收敛因子 a 的变化:较大的收敛因子 a 具有较好的全局搜索能力,避免算法陷入局部最优;较小的收敛因子 a 具有较强的局部搜索能力,以加快算法的收敛速度.

2 基于混沌搜索策略的鲸鱼优化算法

2.1 混沌反向学习初始化策略

研究表明^[13], 初始种群的好坏影响着算法的求解精度和收敛速度, 多样性较好的初始种群对提升算法性能很有帮助. 然而, WOA算法在求解优化问题时通常采用随机方法产生初始种群, 可能使得初始种群分布不均, 导致初始种群多样性较差. 此外, 由于对优化问题的全局最优解没有任何先验知识, 应尽可能使种群均匀分布在搜索空间. 为了增强种群的多样性并提高求解效率, 为算法进行全局搜索奠定多样性基础, CWOA算法采用混沌反向学习初始化策略. 首先利用混沌变量的随机性、遍历性和规律性特征, 产生具有较好多样性的混沌初始种群; 其次通过对混沌初始种群及其反向种群进行排序选择, 将适应度值较优的解作为初始种群, 提高求解效率.

假设种群规模为 N , 采用具有较好遍历均匀性和较快迭代速度的 Tent 混沌映射在 D 维欧氏空间中生成混沌序列 $y = \{y_d, d = 1, 2, \dots, D\}$, $y_d = \{y_{id}, i = 1, 2, \dots, N\}$, Tent 混沌映射函数表达式^[14]为

$$y_{i+1,d} = \begin{cases} 2y_{id}, & y_{id} < 0.5; \\ 2(1 - y_{id}), & y_{id} \geq 0.5. \end{cases} \quad (8)$$

将混沌序列映射到解空间中, 得到种群 $X = \{X_i, i = 1, 2, \dots, N\}$, $X_i = \{X_{id}, d = 1, 2, \dots, D\}$, 种群个体 X_{id} 表示为

$$X_{id} = X_{\min d} + y_{id} \cdot (X_{\max d} - X_{\min d}). \quad (9)$$

其中: X_{id} 为第 i 个种群个体的第 d 维码值, $X_{\min d}$ 和 $X_{\max d}$ 为 X_{id} 的搜索上下界.

由种群 X 计算反向种群 $OX = \{OX_i, i = 1, 2, \dots, N\}$, $OX_i = \{OX_{id}, d = 1, 2, \dots, D\}$, 反向种群个体 OX_{id} 表示为

$$OX_{id} = X_{\min d} + X_{\max d} - X_{id}. \quad (10)$$

种群 X 与反向种群 OX 合并, 得到新种群 $\{X \cup OX\}$. 计算新种群的目标函数值并排序, 选取其中适应度值最好的 N 个个体作为初始种群.

2.2 收敛因子和惯性权重混沌扰动协同更新策略

在 WOA 算法中, 收敛因子 a 随迭代次数的增加由 2 线性减小到 0, 属于线性时变更新策略. 然而, 收敛因子 a 的线性时变更新策略不能完全体现出实际 WOA 算法的优化搜索过程. 文献[11]提出了一种非线性收敛因子, 在搜索前期以较小的值随迭代次数增加而递增, 当增大到一个较大值后, 再快速递减到一个较小值, 最后以较慢的速度再递增, 属于非线性时变更新策略. 本文采用具有较好遍历性的逻辑自映

射函数产生混沌序列^[15], 进而对收敛因子进行混沌扰动, 提出一种带混沌扰动的收敛因子非线性时变更新策略, 收敛因子更新公式为

$$a = a_{\text{initial}} \cdot |y^t| - (a_{\text{initial}} - a_{\text{final}}) \cdot \tan\left(0.875 \cdot \frac{t}{t_{\max}}\right). \quad (11)$$

其中: a_{initial} 和 a_{final} 分别为收敛因子 a 的初值和终值; $y^t = 1 - 2(y^{t-1})^2$, $y^t \in (-1, 1)$ 为自逻辑映射函数产生的混沌序列.

通过施加带混沌扰动的收敛因子非线性时变更新策略, 在一定程度上提升了算法性能. 然而, 当涉及到时变更新策略时, 单独依靠收敛因子并不能在种群全局探索能力与局部开发能力之间进行有效平衡. 因此, CWOA 算法借鉴 PSO 算法, 引入惯性权重配合收敛因子共同调节种群全局探索能力和局部开发能力, 惯性权重更新策略同样选择带混沌扰动的非线性时变更新策略. 惯性权重更新公式为

$$\omega = \omega_{\text{final}} \cdot |y^t| + (\omega_{\text{initial}} - \omega_{\text{final}}) \cdot \left(\frac{t_{\max} - t}{t_{\max}}\right)^2, \quad (12)$$

其中 ω_{initial} 和 ω_{final} 分别为惯性权重 ω 的初值和终值.

通过收敛因子和惯性权重混沌扰动协同更新, 当前个体的位置更新和螺旋更新位置公式分别为

$$X^{t+1} = \omega \cdot X_{\text{gbest}}^t - A \cdot |C \cdot X_{\text{gbest}}^t - X^t|, \quad (13)$$

$$X^{t+1} = \omega \cdot X_{\text{gbest}}^t + D \cdot e^{bl} \cdot \cos(2\pi l). \quad (14)$$

2.3 最优个体混沌搜索策略

WOA 算法迭代后期, 由于群体中所有鲸鱼个体均向最优个体位置聚集, 从而导致群体内多样性缺失, 算法过早收敛到一种非全局最优状态, 出现早熟收敛现象. 为了减少 WOA 算法出现早熟收敛现象的概率, CWOA 算法通过对当前最优鲸鱼个体施加混沌搜索策略, 以当前搜索到最优个体位置向量 $X_{\text{gbest}}^t = (X_{\text{gbest}1}^t, X_{\text{gbest}2}^t, \dots, X_{\text{gbest}D}^t)$ 为基础, 将当前最优个体位置向量线性映射到混沌变量的取值区间, 利用混沌变量的随机性、遍历性和规律性等内在特性进行混沌优化搜索, 最终将获得的混沌优化解线性转换到优化空间, 此时的混沌优化本质上是一种深度局部搜索方法. 混沌搜索主要步骤如下.

Step 1: 按照 $y_{\text{gbest}d} = \frac{2(X_{\text{gbest}d} - X_{\min d})}{X_{\max d} - X_{\min d}} - 1$ 将当前最优个体位置向量的每一维映射到区间 $(-1, 1)$.

Step 2: 将混沌变量加载于待搜索的个体变量 $y_{\text{gbest}d}$, 即将 $y_{\text{gbest}d}$ 代入逻辑自映射函数 $y_{\text{gbest}d}^{t_c} = 1 - 2(y_{\text{gbest}d}^{t_c-1})^2$ 进行迭代, 产生混沌序列 $y_{\text{gbest}d}^{t_c}(t_c =$

1, 2, \dots, t_{c \max}). 其中: t_{c \max} 为混沌搜索的最大迭代次数, y_{g \text{best} d}^t \in (-1, 1).

Step 3: 利用式 X'_{g \text{best} d} = \frac{1}{2}(X_{\max d} - X_{\min d}) \times y_{g \text{best} d} + \frac{1}{2}(X_{\max d} + X_{\min d}) 将 y_{g \text{best} d} 载波到原搜索空间邻域内以产生新最优个体位置向量 X'_{g \text{best}}.

Step 4: 对 X_{g \text{best}} 和 X'_{g \text{best}} 进行适应度函数值评价, 更新当前最优个体位置.

Step 5: 判断是否达到最大混沌搜索次数, 如果是, 则终止混沌搜索, 否则转入 Step 2.

2.4 算法步骤

综上所述, CWOA 算法步骤的伪代码表示如下:

begin

初始化算法参数, 按照 2.1 节所述的混沌反向学习初始化策略产生初始化鲸鱼种群 {X_i, i = 1, 2, \dots, N};

计算群体中每个个体的适应度值 {F(X_i), i = 1, 2, \dots, N}, 记录当前最优个体位置.

while (t < t_{\max}) do

for i = 1 to N do

根据式(11)计算收敛因子 a 的值;

根据式(12)计算惯性权重 \omega 的值.

if (p < 0.5) do

if (|A| < 1) do

按照式(13)更新当前个体位置.

else if (|A| \ge 1) do

按照式(7)更新当前个体位置.

end if

else if (p \ge 0.5) do

按照式(14)更新当前个体位置.

end if

end for

按照 2.3 节所述策略执行最优个体混沌搜索; 更新当前最优个体位置.

t = t + 1.

end while

end

3 数值实验及比较

3.1 基准测试函数及性能指标

为了测试 CWOA 算法性能, 从文献 [10-11] 中选取 10 个不同类型的基准测试函数, 其函数名、表达式、类型、搜索范围、理论最优值和收敛精度如表 1 所示. 在表 1 中: U 表示单峰函数, M 表示多峰函数, S 表示可分函数, N 表示不可分函数, f_{\min} 表示理论最优值.

表 1 基准测试函数

函数名	表达式	类型	搜索范围	f_{\min}	收敛精度
Sphere	f_1(x) = \sum_{i=1}^D x_i^2	US	[-100, 100]	0	1 \times 10^{-08}
Schwefel2.22	f_2(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i	UN	[-10, 10]	0	1 \times 10^{-08}
Schwefel2.21	f_3(x) = \max_i \{ x_i , 1 \le x_i \le D\}	UN	[-100, 100]	0	1 \times 10^{-08}
Rosenbrock	f_4(x) = \sum_{i=1}^D [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]	UN	[-5, 10]	0	1 \times 10^0
Step	f_5(x) = \sum_{i=1}^D ([x_i + 0.5])^2	US	[-100, 100]	0	1 \times 10^{-08}
Quartic	f_6(x) = \sum_{i=1}^D ix_i^4 + \text{random}[0, 1]	US	[-1.28, 1.28]	0	1 \times 10^{-04}
Rastrigin	f_7(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]	MS	[-5.12, 5.12]	0	1 \times 10^{-08}
Ackley	f_8(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e	MN	[-32, 32]	0	1 \times 10^{-08}
Griewank	f_9(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1	MN	[-600, 600]	0	1 \times 10^{-08}
Penalized1	f_{10}(x) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_D - 1)^2 \right\} + \sum_{i=1}^D u(x_i, 10, 100, 4), y_i = 1 + \frac{1}{4}(x_i + 1),	MN	[-600, 600]	0	1 \times 10^{-02}
	u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a; \\ 0, & -a \le x_i \le a; \\ k(-x_i - a)^m, & x_i < -a. \end{cases}				

本文选取两个性能指标评价算法性能. 第1个性能指标是结果精确度(AC), 该指标反映了算法所得结果与问题全局最优解的接近程度. 假设一个问题的全局最优解是 X_{opt} , t_{max} 次迭代后找到的最优解为 P_{best} , 则所求结果的精确度定义为

$$AC = |F(P_{best}) - F(X_{opt})|. \quad (15)$$

第2个性能指标是寻优成功率(SR), 即多次实验中算法收敛到问题最优解的比例, 定义为

$$SR = \frac{z'}{z} \times 100. \quad (16)$$

其中: z 为总的实验次数, z' 为算法收敛到问题全局最优解的实验次数.

3.2 在基准测试函数上的算法性能对比分析

采用CWOA算法对上述10个基准测试函数进行优化求解. 10个函数的维数分别设置为30维、200维、

500维和1000维, 并与基本WOA算法和文献[11]提出的IWOA算法性能进行比较, 通过统计3种算法不同性能测度的结果, 对比分析3种算法的优化性能.

为了比较的公平性, 3种算法均采用相同的实验参数, 即种群规模 $N = 30$, 最大迭代次数 $t_{max} = 500$ (即最大适应度函数评价次数为15000次). 经过反复多次实验, 确定CWOA算法的其他最优参数为: 常数 $b = 1$, 收敛因子初值 $a_{initial} = 2$, 收敛因子终值 $a_{final} = 0$, 惯性权重初值 $\omega_{initial} = 0.9$, 惯性权重终值 $\omega_{final} = 0.2$, 混沌搜索最大迭代次数 $t_c = 50$. 基本WOA算法和IWOA算法其他最优参数按照对应参考文献设置. 对于每个测试函数, 3种算法均独立运行30次, 记录其平均精度、精度标准差和成功率, 结果如表2所示, 其中黑色粗体为比较算法中最好结果.

由表2可知, 函数维数设置为30、200、500和

表2 3种算法对基准测试的寻优性能比较

函数	维数	WOA			IWOA			CWOA		
		平均精度	标准差	成功率/%	平均精度	标准差	成功率/%	平均精度	标准差	成功率/%
$f_1(x)$	30	1.86e-070	1.01e-069	100	1.02e-115	5.07e-115	100	0	0	100
	200	1.03e-072	4.73e-072	100	1.69e-112	5.60e-112	100	0	0	100
	500	1.02e-069	4.45e-069	100	8.89e-110	4.85e-109	100	0	0	100
	1000	3.42e-067	1.41e-066	100	1.76e-111	7.69e-111	100	0	0	100
$f_2(x)$	30	2.25e-049	1.22e-048	100	1.92e-068	6.18e-068	100	4.56e-226	0	100
	200	9.63e-049	4.90e-048	100	4.39e-066	1.80e-065	100	9.53e-243	0	100
	500	8.88e-047	3.65e-046	100	1.25e-065	4.55e-065	100	1.24e-230	0	100
	1000	1.06e-048	3.24e-048	100	1.14e-064	4.50e-064	100	2.14e-238	0	100
$f_3(x)$	30	4.63e+001	3.16e+001	0	1.63e-000	1.55e-006	0	3.60e-265	0	100
	200	7.63e+001	2.37e+001	0	3.07e+001	2.57e+001	0	2.49e-214	0	100
	500	8.50e+001	1.62e+001	0	5.48e+001	3.23e+001	0	1.90e-219	0	100
	1000	7.66e+001	2.20e+001	0	6.82e+001	2.51e+001	0	2.56e-250	0	100
$f_4(x)$	30	2.79e+001	4.70e-001	0	2.82e+001	1.81e-001	0	2.74e+001	5.17e-000	3.33
	200	1.98e+002	2.22e-001	0	1.98e+002	5.43e-002	0	1.97e+002	8.65e-002	0
	500	4.96e+002	4.66e-001	0	4.96e+002	3.78e-001	0	4.94e+002	2.35e-001	0
	1000	9.94e+002	2.77e-001	0	9.92e+002	8.29e-001	0	9.90e+002	4.51e-001	0
$f_5(x)$	30	3.33e-002	1.83e-001	96.67	0	0	100	0	0	100
	200	3.33e-002	1.83e-001	96.67	0	0	100	0	0	100
	500	0	0	100	0	0	100	0	0	100
	1000	0	0	100	0	0	100	0	0	100
$f_6(x)$	30	2.96e-003	3.61e-003	10	2.19e-003	2.14e-003	6.67	3.61e-005	3.73e-005	90
	200	4.63e-003	3.84e-003	0	1.97e-003	2.18e-003	6.67	3.50e-005	3.21e-005	96.67
	500	4.96e-003	4.97e-003	6.67	2.57e-003	3.26e-003	10	4.12e-005	3.56e-005	93.33
	1000	4.55e-003	4.55e-003	0	2.18e-003	2.75e-003	10	3.72e-005	3.48e-005	96.67
$f_7(x)$	30	0	0	100	0	0	100	0	0	100
	200	0	0	100	0	0	100	0	0	100
	500	0	0	100	0	0	100	0	0	100
	1000	0	0	100	0	0	100	0	0	100
$f_8(x)$	30	4.20e-015	2.27e-015	100	4.09e-015	2.35e-015	100	8.88e-016	4.01e-031	100
	200	5.15e-015	1.94e-015	100	8.88e-016	0	100	8.88e-016	4.01e-031	100
	500	5.86e-015	2.97e-015	100	4.44e-015	0	100	8.88e-016	4.01e-031	100
	1000	7.28e-015	3.59e-015	100	5.15e-015	2.89e-015	100	8.88e-016	4.01e-031	100
$f_9(x)$	30	8.87e-003	3.38e-002	93.33	0	0	100	0	0	100
	200	0	0	100	0	0	100	0	0	100
	500	0	0	100	0	0	100	0	0	100
	1000	0	0	100	0	0	100	0	0	100
$f_{10}(x)$	30	1.94e-002	1.47e-002	33.33	1.16e-002	6.07e-003	46.67	3.09e-002	1.37e-002	6.67
	200	6.09e-002	2.18e-002	0	9.14e-002	3.63e-002	0	4.14e-002	2.11e-002	3.33
	500	8.97e-002	4.19e-002	0	1.43e-001	6.97e-002	0	3.28e-002	1.78e-002	16.67
	1000	1.19e-001	5.15e-002	0	1.59e-001	6.19e-002	0	3.46e-002	1.76e-002	3.33

1 000维时,除了函数 $f_4(x)$ 和 $f_{10}(x)$,CWOA算法在其他8个函数30次独立实验中均一致收敛到全局最优解,尤其是函数 $f_1(x)$ 、 $f_5(x)$ 、 $f_7(x)$ 和 $f_9(x)$,CWOA算法均能收敛到理论最优值0,对于函数 $f_2(x)$ 和 $f_3(x)$,CWOA算法获得的最优解非常接近理论最优值0.

与WOA算法和IWOA算法相比,CWOA算法在测试函数 $f_1(x)$ 、 $f_2(x)$ 、 $f_3(x)$ 和 $f_6(x)$ 上均获得了较好的寻优性能;对于函数 $f_4(x)$ 和 $f_8(x)$,CWOA算法寻优性能略优于IWOA算法和WOA算法;对于函数 $f_5(x)$ 、 $f_7(x)$ 和 $f_9(x)$,3种算法得到了相似的寻优性能;对于函数 $f_{10}(x)$,当函数维数为30维时,IWOA算法获得了较好的平均精度和标准差,当函数维数为200、500和1 000时,CWOA算法获得了较好的平均精度和标准差.在寻优成功率方面,除了函数 $f_4(x)$ 、 $f_6(x)$ 和 $f_{10}(x)$,CWOA算法在其他7个函数上的寻优成功率均为100%,在函数 $f_6(x)$ 上的寻优成功率接近100%.函数维数为500和1 000维时,3种算法在求解高维优化问题上均显示出了较强的鲁棒性.

为了更加直观地反映CWOA算法性能,图2~图5给出了3种算法在部分函数维数为200维时的寻优收敛曲线.由图2~图5可知,与WOA算法和IWOA算法相比,CWOA算法具有更高收敛精度和更快收敛速度.

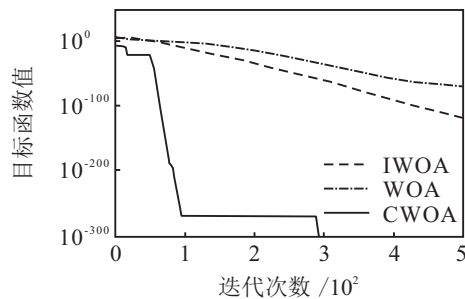


图2 3种算法对Sphere函数(200维)的收敛曲线

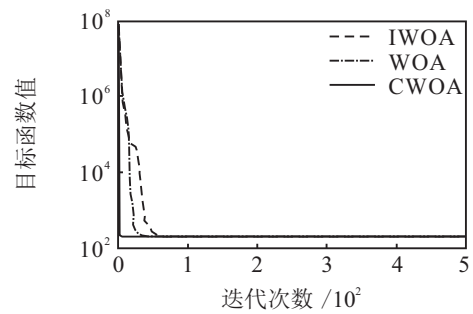


图3 3种算法对Rosenbrock函数(200维)的收敛曲线

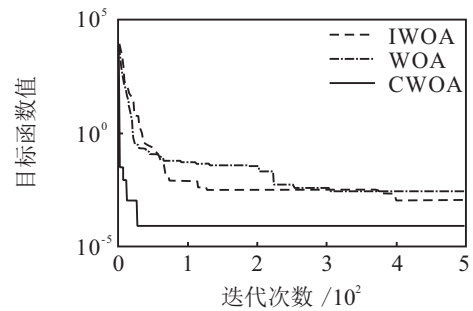


图4 3种算法对Quartic函数(200维)的收敛曲线

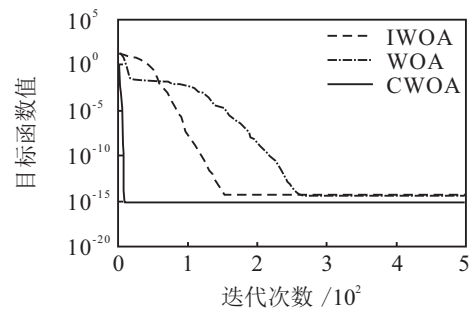


图5 3种算法对Ackley函数(200维)的收敛曲线

3.3 算法改进策略有效性分析

为了分析3种混沌搜索策略对CWOA算法性能的有效性,对表1中的10个基准测试函数进行优化求解.将CWOA算法与仅采用混沌反向学习初始化策略的WOA算法(记为CWOA-1)、仅采用收敛因子和

表3 4种算法对10个基准测试函数(200维)的寻优性能比较

函数	CWOA-1		CWOA-2		CWOA-3		CWOA	
	平均精度	标准差	平均精度	标准差	平均精度	标准差	平均精度	标准差
$f_1(x)$	6.31e-078	1.40e-077	0	0	8.23e-077	1.48e-076	0	0
$f_2(x)$	1.75e-053	2.53e-053	2.41e-233	0	2.07e-051	3.11e-051	9.53e-243	0
$f_3(x)$	7.54e+001	2.61e+001	1.48e-247	0	7.97e+001	2.24e+001	2.49e-214	0
$f_4(x)$	1.98e+002	1.38e-001	1.97e+002	1.01e-001	1.98e+002	1.20e-001	1.97e+002	8.65e-002
$f_5(x)$	0	0	0	0	0	0	0	0
$f_6(x)$	2.85e-003	2.62e-003	5.80e-005	8.19e-005	1.51e-003	1.21e-003	3.50e-005	3.21e-005
$f_7(x)$	0	0	0	0	0	0	0	0
$f_8(x)$	4.56e-015	2.55e-015	8.88e-016	4.01e-031	3.49e-015	2.27e-015	8.88e-016	4.01e-031
$f_9(x)$	0	0	0	0	0	0	0	0
$f_{10}(x)$	5.84e-002	2.11e-002	3.79e-002	2.00e-002	7.29e-002	4.04e-002	4.14e-002	2.11e-002

惯性权重混沌扰动协同更新策略的WOA算法(记为CWOA-2)、仅采用最优个体混沌搜索策略的WOA算法(记为CWOA-3)对比,4种算法的参数设置与3.2节相同.表3给出了4种算法对函数维数为200维的寻优性能比较.

由表3可知,仅采用混沌反向学习初始化策略(CWOA-1)和仅采用最优个体混沌搜索策略(CWOA-3)对WOA算法性能的改进有限,而采用收敛因子和惯性权重混沌扰动协同更新策略(CWOA-2)对WOA算法性能有较大改善,从而说明收敛因子和惯性权重对WOA算法的寻优性能有较大影响,进一步表明收敛因子和惯性权重混沌扰动协同更新策略比仅采用收敛因子非线性时变更新策略更有效.

3.4 在复合测试函数上的算法性能对比分析

为了进一步验证CWOA算法性能和有效性,对文献[16]提出的CF₁ ~ CF₆多峰复合测试函数进行优化求解,并与基本WOA算法和主流PSO算法、ABC算法和文献[17]提出的基于引力搜索机制的花朵授粉算法(GSFPA)性能进行比较.在对比实验中,为了

比较的公平性,CF₁ ~ CF₆多峰复合测试函数的维数均设置为10维,5种算法均采用相同的实验参数,即种群规模N = 20,最大迭代次数t_{max} = 500(即最大适应度函数评价次数为10000次).基本WOA算法和CWOA算法的其他参数同3.2节;PSO算法、ABC算法和GSFPA算法的其他最优参数按照文献[17]设置.

对于每个复合测试函数,5种算法均独立运行30次,记录其平均精度和标准差,结果如表4所示.

由表4可知:基本WOA算法和CWOA算法相比于其他3种算法,在复合测试函数CF₁ ~ CF₄和CF₆上获得了较好的寻优性能,在CF₅函数上,基本WOA算法和CWOA算法性能与其他3种算法相当;在函数CF₁、CF₂、CF₄和CF₆上,CWOA算法性能要优于基本WOA算法;在函数CF₃和CF₅上,CWOA算法性能虽劣于基本WOA算法,但优于其他3种算法.因此,总体上CWOA算法在复合测试函数上的寻优性能要优于其他4种算法,从而说明将混沌搜索策略融入基本WOA算法是行之有效的.

表4 5种算法对复合测试函数的寻优性能比较

函数	WOA		PSO		ABC		GSFPA		CWOA	
	平均精度	标准差	平均精度	标准差	平均精度	标准差	平均精度	标准差	平均精度	标准差
CF ₁	6.01e-001	4.37e-001	2.00e+002	1.05e+002	1.29e+002	8.86e+001	1.28e+002	3.50e+001	5.46e-001	4.59e-001
CF ₂	9.53e+001	5.27e+001	1.76e+002	8.59e+001	1.49e+002	4.94e+001	1.19e+002	5.26e+001	7.85e+001	4.91e+001
CF ₃	7.46e+001	3.16e+001	6.27e+002	2.32e+002	2.90e+002	7.53e+001	4.35e+002	6.10e+001	8.03e+001	3.37e+001
CF ₄	7.25e+001	4.38e+001	5.67e+002	7.01e+001	4.98e+002	9.10e+001	4.95e+002	3.58e+001	6.48e+001	5.22e+001
CF ₅	1.07e+002	3.85e+001	1.75e+002	1.78e+002	1.20e+002	1.19e+002	1.29e+002	6.19e+001	1.24e+002	3.37e+001
CF ₆	8.93e+001	3.61e+001	9.02e+002	8.42e-001	7.80e+002	1.82e+002	6.63e+002	1.64e+002	7.54e+001	4.25e+001

4 结论

本文提出了一种基于混沌搜索策略的WOA算法,即CWOA算法.所提算法采用混沌反向学习初始化策略产生初始种群,增强了种群的多样性,提高了求解效率;受PSO算法启发,通过收敛因子和惯性权重混沌扰动协同更新,以平衡算法全局探索和局部开发能力;为了减少WOA算法出现早熟收敛现象的概率,设计当前鲸鱼最优个体的混沌搜索策略.通过对10个基准测试函数和6个复合测试函数进行数值实验表明,本文提出的CWOA算法具有较高的收敛精度、较快的收敛速度和较好的鲁棒性,且能有效处理高维函数优化问题.下一步研究内容是将CWOA算法应用于约束优化问题、多目标优化和实际工程问题中.

参考文献(References)

- [1] Kennedy J, Eberhart R. Particle swarm optimization[C]. Proc of the 1995 IEEE Int Conf on Neural Networks. Piscataway: IEEE Press, 1995: 1942-1948.
- [2] Shi Y, Eberhart R. A modified particle swarm optimizer[C]. Proc of the 1998 IEEE Int Conf on Evolutionary Computation. Piscataway: IEEE Press, 1998: 69-73.
- [3] Dorigo M, Maniezzo V, Colorni A. Ant system: Optimization by a colony of cooperating agents[J]. IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics, 1996, 26(1): 29-41.
- [4] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm[J]. J of Global Optimization, 2007, 39(3): 457-471.

- [5] Yang X S. A new metaheuristic bat-inspired algorithm[C]. *Nature Inspired Cooperative Strategies for Optimization*. Berlin: Springer, 2010: 65-74.
- [6] Yang X S. Firefly algorithm, stochastic test functions and design optimization[J]. *Int J of Bio-Inspired Computation*, 2010, 2(2): 78-84.
- [7] Pan W T. A new fruit fly optimization algorithm: taking the financial distress model as an example[J]. *Knowledge-Based Systems*, 2012(26): 69-74.
- [8] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. *Advances in Engineering Software*, 2014, 69(7): 46-61.
- [9] Mirjalili S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems[J]. *Neural Computing & Applications*, 2016, 27(4): 1053-1073.
- [10] Mirjalili S, Lewis A. The whale optimization algorithm[J]. *Advances in Engineering Software*, 2016(95): 51-67.
- [11] 龙文, 蔡绍洪, 焦建军, 等. 求解大规模优化问题的改进鲸鱼优化算法[J]. *系统工程理论与实践*, 2017, 37(11): 2983-2994.
(Long W, Cai S H, Jiao J J, et al. Improved whale optimization algorithm for large scale optimization problems[J]. *Systems Engineering—Theory & Applications*, 2017, 37(11): 2983-2994.)
- [12] 牛培峰, 吴志良, 马云鹏, 等. 基于鲸鱼优化算法的汽轮机热耗率模型预测[J]. *化工学报*, 2017, 68(3): 1049-1057.
(Niu P F, Wu Z L, Ma Y P, et al. Prediction of steam turbine heat consumption rate based on whale optimization algorithm[J]. *CIESC J*, 2017, 68(3): 1049-1057.)
- [13] Haupt R, Haupt S. Practical genetic algorithm[M]. New York: John Wiley and Sons, 2004: 51-65.
- [14] Jordehi A R. Chaotic bat swarm optimisation (CBSO)[J]. *Applied Soft Computing*, 2015(26): 523-530.
- [15] 刘长平, 叶春明. 基于逻辑自映射的变尺度混沌粒子群优化算法[J]. *计算机应用研究*, 2011, 28(8): 2825-2827.
(Liu C P, Ye C M. Mutative scale chaos particle swarm optimization algorithm based on self logical mapping function[J]. *Application Research of Computer*, 2011, 28(8): 2825-2827.)
- [16] Liang J J, Suganthan P N, Deb K. Novel composition test functions for numerical global optimization[C]. *Proc of the 2005 IEEE Swarm Intelligence Symposium*. Pasadena: IEEE, 2005: 68-75.
- [17] 肖辉辉, 万常选, 段艳明, 等. 基于引力搜索机制的花朵授粉算法[J]. *自动化学报*, 2017, 43(4): 576-594.
(Xiao H H, Wan C X, Duan Y M, et al. Flower pollination algorithm based on gravity search mechanism[J]. *Acta Automatica Sinica*, 2017, 43(4): 576-594.)

作者简介

王坚浩(1982—), 男, 讲师, 博士, 从事装备保障任务规划与资源调度、智能控制、计算与优化的研究, E-mail: hamilton_wang@sina.com;

张亮(1982—), 男, 副教授, 博士, 从事装备综合保障工程等研究, E-mail: 179108235@qq.com;

史超(1978—), 男, 副教授, 博士, 从事装备综合保障工程等研究, E-mail: sc78115@sohu.com;

车飞(1983—), 男, 讲师, 博士, 从事装备综合保障工程的研究, E-mail: cfei_1983@163.com;

丁刚(1983—), 男, 讲师, 硕士, 从事装备综合保障工程的研究, E-mail: baffulo@sina.com;

武杰(1985—), 男, 工程师, 博士, 从事装备综合保障工程等研究, E-mail: wujiekgd@163.com.

(责任编辑: 闫妍)