控制与决策
Control and Decision

基于余弦控制因子和多项式变异的鲸鱼优化算法

黄清宝， 李俊兴， 宋春宁， 徐辰华， 林小峰

文章编号: 1001-0920(2020)03-0559-10 DOI: 10.13195/j.kzyjc.2018.0463

基于余弦控制因子和多项式变异的鲸鱼优化算法

黄清宝†, 李俊兴, 宋春宁, 徐辰华, 林小峰

(广西大学 电气工程学院，南宁 530004)

摘 要: 针对基本鲸鱼优化算法(Whale optimization algorithm, WOA)在求解最优解不在原点附近的目标函数时存在收敛精度低、 易陷入局部最优解的缺陷, 提出一种基于余弦控制因子和多项式变异的鲸鱼优化算法(CPWOA). 所提算法中控制参数按照余弦曲线变化, 并加入同步余弦惯性权值, 使得在迭代前期减缓收敛速度以进行充分的全局探索, 而在迭代后期加速收敛以提高算法精度; 同时, 对最佳鲸鱼位置引入多项式变异, 以增强算法跳出局部最优解的能力. 将所提算法对多个shifted单峰、多峰和固定维测试函数进行求解, 实验结果表明, 与基本WOA、EHO、GWO、SCA、MBO以及其他改进型WOA算法相比, CPWOA对绝大多数测试函数的求解有更高的精度和稳定性. 用非参数估计方法对计算结果进行差异显著性统计检验, 表明CPWOA算法的显著性更优.

关键词: 余弦因子；多项式变异；鲸鱼优化算法；全局优化；偏移型测试函数；统计检验

中图分类号: TP18 **文献标志码:** A

Whale optimization algorithm based on cosine control factor and polynomial mutation

HUANG Qing-bao†, LI Jun-xing, SONG Chun-ning, XU Chen-hua, LIN Xiao-feng

(School of Electrical Engineering，Guangxi University，Nanning 530004，China)

Abstract: The basic whale optimization algorithm (WOA) has the defects of low convergence, getting easily trapped into local optima, and being difficult to keep balance in exploration and exploitation when solving the shifted functions whose optimum are not at the near origin. A whale optimization algorithm based on cosine control factors and polynomial mutation (CPWOA) is proposed to solve the mentioned defects. In this algorithm, the control parameter is changed as a cosine curve, and a synchronous cosine inertia weight is added to slow down the convergence speed early in the iteration of the algorithme thus improve exploration, and to accelerate the convergence in the later iteration thus improve the accuracy of the exploitation. And polynomial mutation is joined in the optimum whale location to enhance the ability of jumping out of local optimal solutions. By the experiments on multiple shifted benchmark functions such as unimodal, multimodal, and fixed-dimension multimodal, the proposed strategy outperforms the WOA, the EHO, the GWO, the SCA, the MBO and other improved WOAs on solution accuracy and stability. The non-parametric statistical test is carried out to show the significance of the difference of the proposed method.

Keywords: cosine factor；polynomial mutation；whale optimization algorithm；global optimization；shifted benchmark functions；non-parametric statistical test

0 引 言

鲸鱼优化算法(WOA)是由Mirjalili等[1]提出的一种新型启发式搜索算法, 该算法通过模仿座头鲸群体捕食行为来寻找目标问题的最优解. 与现有的启发式算法相比, WOA操作简单, 调整的参数少, 且由于它的最优个体及运动方式依照概率更新, 具有更大的随机性和更快的收敛速度, 在工程实践的初步应用中取得了良好的效果. Jangir等[2]利用WOA训练神经网络中的多层感知器, 获得了更高的训练效率和精度; Khaled等[3]将WOA应用到电力系统, 进行无功功率最优调度; Prakash等[4]将WOA应用于径向配电网电容器的最佳选址中. 然而, 在求解更复杂工程的目标函数时, WOA与其他智能算法类似, 存在依赖初始种群、收敛精度低和易陷入局部最优的问题. 牛培峰等[5]提出了一种反向学习自适应的鲸鱼优化算法(AWOA), 利用反向学习提高了算法的种群初始化质量, 并引入自适应惯性权值, 提高了算法的收敛精度和收敛速度; Oliva等[6]提出了一种混沌鲸鱼优化

收稿日期: 2018-04-15；**修回日期:** 2018-10-23.

基金项目: 国家自然科学基金项目(51767005)；广西自然科学基金项目(2017GXNSFAA198225).

责任编委: 巩敦卫.

†通讯作者. E-mail: qingbaohuang@163.com.

算法(CWOA),利用混沌变异提高了算法跳出局部最优解的能力. 这些改进后的WOA对于求解全局最优解在原点附近或者低维目标函数时,具有非常高的收敛精度和稳定性,但在求解全局最优解远离原点的目标函数时,往往收敛精度低,容易陷入局部最优解. WOA的这个倾向性限制了它在更多实际问题上的应用.

针对上述问题,本文提出基于余弦控制因子和多项式变异的鲸鱼优化算法(CPWOA). 首先,使控制参数按照余弦曲线变化,并加入同步余弦惯性权值,使得算法在迭代前期减缓收敛速度以进行充分的全局探索,而在迭代后期加速收敛以提高搜索精度;然后,对最佳鲸鱼位置引入多项式变异,以增强算法跳出局部最优解的能力;最后,采用多个位置偏移的测试函数进行实验及非参数统计检验,结果表明,所提出的算法对最优解偏离原点的目标函数的求解具有更高的精度和稳定性,检验结果显示了这种算法与其他算法的差异显著性.

1 鲸鱼优化算法

鲸鱼优化算法是一种模仿座头鲸狩猎策略的启发式搜索算法. 它包括3个环节:包围猎物、气泡网攻击和寻找猎物.

1.1 包围猎物

在包围猎物环节,距离猎物最近的鲸鱼的位置相当于一个局部最优解,其他鲸鱼根据最佳鲸鱼位置进行更新,包围猎物. 数学模型如下所示:

$$D=|C\cdot X^*(t)-X(t)|, \tag{1}$$

$$X(t+1)=X^*(t)-A\cdot D. \tag{2}$$

其中:t是迭代次数;X^*是到目前为止获得的最佳解;X是当前解;A和C是矩阵系数,分别表示为

$$A=2a\cdot r-a, \tag{3}$$

$$C=2\cdot r. \tag{4}$$

r是[0, 1]中的随机向量;随着t的增加,a从2线性减小到0,表示为

$$a=2-2\cdot\frac{t}{t_{\rm MaxIter}}, \tag{5}$$

$t_{\rm MaxIter}$是最大迭代次数.

1.2 气泡网攻击

鲸鱼吐出气泡,并以螺线运动轨迹向目标猎物游去,个体位置更新的数学模型为

$$X(t+1)=D\cdot{\rm e}^{bl}\cdot\cos(2\pi l)+X^*(t). \tag{6}$$

其中:D同式(1),b为常数,l为[0,1]中的随机向量.

1.3 寻找猎物

在寻找猎物阶段,鲸鱼随机游走觅食,进行全局探索,其数学模型为

$$D=|C\cdot X_{\rm rand}(t)-X(t)|, \tag{7}$$

$$X(t+1)=X_{\rm rand}(t)-A\cdot D, \tag{8}$$

其中$X_{\rm rand}$表示鲸鱼从种群中随机选择某一个体的位置作为目标位置.

在WOA中,当控制参数$|A|<1$时,进行局部最优解搜索,此时鲸鱼以概率P^*进行包围猎物,以概率$1-P^*$进行螺线运动;当控制参数$|A|\geqslant 1$时,算法进行全局最优解探索.

2 鲸鱼优化算法的改进

2.1 控制参数a余弦变化

在基本鲸鱼优化算法中,式(8)描述算法的全局探索,式(2)和(6)描述算法的局部搜索. 参数a决定了控制参数A的变化,从而协调算法全局探索与局部搜索. a越大,算法的全局探索能力越强;a越小,算法的局部搜索能力越强. 式(5)描述的参数a是线性减小的,即算法全局探索能力线性减小. 而在处理高维复杂函数的优化问题时,通常需要增强算法的全局探索能力避免算法早熟[7],故需对式(5)进行调整. 本文根据余弦曲线变化特点,将控制因子a的变化改为如下所示:

$$a=2\cdot\cos\Big(\frac{\pi}{2}\cdot\frac{t}{t_{\rm MaxIter}}\Big). \tag{9}$$

在算法迭代前期,a较大并缓慢减小以充分进行全局探索;在算法迭代后期,a急速减小以进行局部搜索.

2.2 权值因子w余弦变化

文献[8]指出,较大的惯性权值有利于算法的全局探索,而较小的惯性权值有利于算法的局部搜索. 为提高算法的收敛速度和精度,文献[9-10]在WOA算法局部搜索更新中引入了自适应权值,如下所示:

$$X(t+1)=\begin{cases}w(t)\cdot X^*(t)-A\cdot D,\ P<P^*;\\ w(t)\cdot X^*(t)+D\cdot{\rm e}^{bl}\cdot\cos(2\pi l),\ P\geqslant P^*.\end{cases} \tag{10}$$

式中权值w最终收敛到0,算法在原点对所有个体具有很强的吸引作用,对于最优解在原点附近的目标函数的求解具有加速作用. 然而,当最优解远离原点时,往往得不到问题的全局最优解.

结合前面控制参数a的余弦变化,本文对鲸鱼优化算法的位置更新引入一个与a同步变化的非线性

惯性权值w,进一步增强算法的全局探索能力. 同时借鉴式(10),将权值位置调整为

$$X(t+1)=\begin{cases}X^*(t)-w(t)\cdot A\cdot D,\ P<P^*;\\ X^*(t)+w(t)\cdot D\cdot \mathrm{e}^{bl}\cdot\cos(2\pi l),\ P\geqslant P^*.\end{cases}\tag{11}$$

权值w的表达式为

$$w(t)=2\cdot\cos\left(\frac{\pi}{2}\cdot\frac{t}{t_{\mathrm{MaxIter}}}\right).\tag{12}$$

当$t\to t_{\mathrm{MaxIter}}$时,$w\to 0$,算法的调整步长变小,全局搜索能力逐渐减弱,局部搜索能力逐渐增强,当w很小时,种群个体的位置进行的是微调而不是趋向原点.

2.3 最优位置多项式变异

在群体智能优化算法中,更加充分地进行全局探索有利于减小算法陷入局部最优解的概率,而在算法中引入变异也将进一步提升算法跳出局部最优解的能力,因为通过引入变异算子能够使算法具有一定的局部随机搜索能力,一方面,在求解的后期,加速向最优解收敛,另一方面也维持解的多样性[11]. 多项式变异是学者们研究多目标优化时常用的一种变异算子,由于单个目标函数优化是多个目标函数优化的特殊情况,将多项式变异用于单个目标函数优化同样适用. 因此,本文对鲸鱼优化算法中的最佳鲸鱼位置引入多项式变异.

多项式变异算子形式为

$$v_{k+1}=v_k+\delta\cdot(u_k-l_k).\tag{13}$$

其中

$$\delta=\begin{cases}[2u+(1-2u)(1-\delta_1)^{\eta_m+1}]^{\frac{1}{\eta_m+1}}-1,\\ \qquad u\leqslant 0.5;\\ 1-[2(1-u)+2(u-0.5)(1-\delta_2)^{\eta_m+1}]^{\frac{1}{\eta_m+1}},\\ \qquad u>0.5.\end{cases}$$

u表示[0,1]上的随机数,η_m表示分布指数,$\delta_1=(v_k-l_k)/(u_k-l_k)$,$\delta_2=(u_k-v_k)/(u_k-l_k)$,$v_k$表示原来最优个体位置. v_{k+1}表示变异后最优个体位置,u_k表示位置上限,l_k表示位置下限.

通过数值实验发现,若变异在算法迭代中以某一概率发生或一直发生,则不利于算法的收敛,尤其会明显减慢单峰问题求解的收敛速度. 为此,本文改变了变异的触发方式,即,当第i次鲸鱼位置更新后的最佳适应度大于或等于第$i-1$次的最佳适应度时,对最佳鲸鱼位置进行多项式变异,若变异后的最佳适应度更优,则变异成功,否则变异失败并保留第i次最佳鲸鱼位置. 这样既能在算法求解停滞时触发变异,又避免了不利的变异.

2.4 算法步骤

CPWOA对鲸鱼个体位置的每一维度进行更新前都先对变量A和C更新. 当目标函数最优解的各维度值差别很大时,则更有利于个体位置的各维度以不同的步长和方向进行调整. CPWOA算法执行过程的伪代码如下所示.

1) 设置种群规模N,最大评价次数t_{MaxFEs},目标函数维数Dim,评价次数$t=0$

2) 随机产生初始群体$\{X_i, i=1,2,\cdots,N\}$,并计算个体适应度值$\{f_{\mathrm{obj}}(X_i), i=1,2,\cdots,N\}$和目前为止最优个体适应度值Leader_score及其位置Leader_pos, $t=N$

3) while $t<t_{\mathrm{MaxFEs}}$ do

4) 　根据式(9)更新a

5) 　for $i=1$ to N do

6) 　　根据式(12)更新权值因子w,更新概率p

7) 　　for $j=1$ to Dim do

8) 　　　if $p<0.5$

9) 　　　　根据式(3)和(4)分别更新A和C

10) 　　　　if $|A|<1$

11) 　　　　　根据式(1)更新D

12) 　　　　　根据式(11)更新种群个体位置X_{ij}

13) 　　　　else if $|A|\geqslant 1$

14) 　　　　　选择随机个体X_{rand}作为最优个体

15) 　　　　　根据式(7)和(8)更新群体位置X_{ij}

16) 　　　　end if

17) 　　　else if $p\geqslant 0.5$

18) 　　　　更新随机变量l

19) 　　　　根据式(11)更新种群个体位置X_{ij}

20) 　　　end if

21) 　　end for

22) 　end for

23) 　计算鲸鱼个体的适应度$f_{\mathrm{obj}}(X_i)$,得到目前为止最优个体适应度值Leader_score及其位置Leader_pos

24) 　if Leader_score ⩾ Leader_score_last

25) 　　按照式(13)进行位置的多项式变异

26) 　　if Leader_score ⩾ Leader_score_last

27) 　　　变异失败,则保留原最优个体

```
        else 变异成功,取新的最优个体
        end if
    t = t + 1
    end if
    t = t + Dim
end while
输出最优值和最佳位置.
```

3 数值实验及分析

3.1 测试函数及性能指标

为了测试本文所提出CPWOA算法的有效性,选取15个标准测试函数进行仿真实验,并将最优解在原点或原点附近的测试函数进行随机位移. 测试函数表达式、自变量每个维度值的搜索范围和最优值如表1～表3所示. 其中: $f_1 \sim f_5$为不定维单峰函数, $f_6 \sim f_{10}$为不定维多峰函数, $f_{11} \sim f_{15}$为固定维多峰函数. 单峰函数主要用于测试算法的收敛速度和求解精度,多峰函数主要用于测试算法全局探索能力. 因为最优解是优化算法的搜索目标,故本文用算法对各测试函数最佳求解结果的平均值和标准差来评价优化算法的性能. 其中,平均值反映算法求解精度,标准差反映算法求解的稳定性.

表1 单峰测试函数(n维)

函数名称	函数表达式	搜索范围	最优值
Shifted Sphere	$f_1(x)=\sum_{i=1}^{n}(x_i-o_i)^2$	$[-100,100]$	0
Shifted Schwefel 2.21	$f_2(x)=\max\{\lvert(x_i-o_i)\rvert, 1\leqslant i\leqslant n\}$	$[-10,10]$	0
Shifted Schwefel 1.2	$f_3(x)=\sum_{i=1}^{n}\Big[\sum_{j=1}^{i}(x_j-o_j)\Big]^2$	$[-100,100]$	0
Shifted Schwefel 2.22	$f_4(x)=\sum_{i=1}^{n}\lvert x_i-o_i\rvert+\prod_{i=1}^{n}\lvert x_i-o_i\rvert$	$[-10,10]$	0
Shifted Quartic with noise	$f_5(x)=\sum_{i=1}^{n}i(x_i-o_i)^4+\mathrm{random}[0,1)$	$[-1.28,1.28]$	0

表2 多峰测试函数(n维)

函数名称	函数表达式	搜索范围	最优值
Shifted Rosenbrock	$f_6=\sum_{i=1}^{n-1}[100(z_i^2-z_{i+1})^2+(z_i-1)^2], z_i=x_i-o_i+1$	$[-100,100]$	0
Shifted Ackley	$f_7(x)=-20\exp\Big(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}z_i^2}\Big)-\exp\Big(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi z_i)\Big)+20+\mathrm{e}, z=x-o$	$[-32,32]$	0
Shifted Griewank	$f_8(x)=\frac{1}{4\,000}\sum_{i=1}^{n}z_i^2-\prod_{i=1}^{n}\cos\Big(\frac{z_i}{\sqrt{i}}\Big)+1, z=x-o$	$[-600,600]$	0
Shifted Rastrigin	$f_9(x)=\sum_{i=1}^{n}(z_i^2-10\cos(2\pi z_i)+10), z=x-o$	$[-5,5]$	0
Shifted Zakharov	$f_{10}(x)=\sum_{i=1}^{n}z_i^2+\Big(\sum_{i=1}^{n}0.5iz_i\Big)^2+\Big(\sum_{i=1}^{n}0.5iz_i\Big)^4, z=x-o$	$[-5,10]$	0

表3 固定维测试函数

函数名称	函数表达式	维数	搜索范围	最小值
Shekel's Foxholes	$f_{11}(x)=\Big(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\Big)^{-1}$	2	$[-65,65]$	0.998
Kowalik	$f_{12}(x)=\sum_{i=1}^{11}\Big[a_i-\frac{x_i(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\Big]^2$	4	$[-5,5]$	0.000 307 5
Branin	$f_{13}(x)=\Big(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\Big)^2+10\Big(1-\frac{1}{8\pi}\Big)\cos x_1+10$	2	$[-5,5]$	0.398
Easom	$f_{14}(x)=-\cos x_1\cdot\cos x_2\cdot\exp(-(x_1-\pi)^2+(x_2-\pi)^2)$	2	$[-100,100]$	-1
Hartman	$f_{15}(x)=-\sum_{i=1}^{4}c_i\exp\Big[-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2\Big]$	6	$[0,1]$	-3.32

3.2　参数设置及实验结果

采用本文提出的CPWOA对表1～表3的测试函数进行数值实验,并与基本WOA和其他一些较新的方法——象群优化算法(EHO)[12]、大红斑蝶优化算法(MBO)[13]、灰狼优化算法(GWO)[14]、正余弦算法(SCA)[15]进行对比. 算法的实验参数设置如表4所示.

表 4　算法的实验参数设置

设置对象	参数值
共同部分	种群规模 $N = 50$,最大评价次数 $\mathrm{Max\_FEs} = 5\mathrm{e}{+}04$,位移 $o_i(i = 1, 2, \cdots, n)$ 在变量范围内随机产生
MBO	$\mathrm{Keep} = 2$, $\mathrm{MaxStepsize} = 1$,种群百分比为5/12
EHO	$\mathrm{Keep} = 2, \alpha = 0.5, \beta = 0.5, n\mathrm{Clan} = 5$
SCA	r_1 由2到0线性减小, $r_2 = 2\pi\mathrm{rand}$, $r_3 = 2\mathrm{rand}$, $r_4 = \mathrm{rand}$
GWO	a 由2到0线性减小
WOA	a 由2到0线性减小, $b = 1$, $P^* = 0.5$
CPWOA	a 由2到0非线性减小, $\eta_m = 2, b = 1, P^* = 0.5$

为研究算法在不同维度下的性能, $f_1 \sim f_{10}$ 在实验中分别设置 $n = 10, 30, 50$. 由于CPWOA算法存在变异,进化一代的适应度评价次数可能会超过种群的个体数量,且各算法初始化适应度评价次数不完全相同,为了公平比较,选择评价次数作为比较的横坐标. 6种算法分别对每个测试函数独立运行30次,记录算法求得的最优函数值的平均值和标准差,结果如表5～表7所示. 在算法迭代过程中,记录每一次评价的最佳适应度,求出每一次评价的平均最优值并绘制曲线. 该曲线反映了算法的收敛趋势,部分收敛曲线如图1和图2所示. 所有仿真实验在Intel Core Quad, CPU i5-8250U, 8 G内存, 1.6 GHz主频的PC上运行,程序采用Matlab 2014a的M脚本语言实现.

由表5～表6可知,总体上, GWO的求解结果比WOA的小,但CPWOA的求解结果最小. 尤其是对于 f_1、f_2 和 f_6,在其他算法求解精度低或无法求解的情况下, CPWOA仍具有较高的求解精度和鲁棒性. 当维度 n 由10增加到30和50时,各算法对目标函数的求解精度和稳定性有所下降,这是由于目标函数更复杂,算法求解需做更多调节.

表 5　测试函数的求解结果(单峰)

函数	n	指标	MBO	EHO	SCA	GWO	WOA	CPWOA
f_1	10	Ave	1.63e+03	8.63e+02	6.40e+02	1.62e+01	1.97e-01	**9.08e-08**
		Std	1.38e+03	6.85e+02	2.75e+02	4.33e+01	2.22e-01	**1.39e-07**
	30	Ave	5.35e+04	3.68e+04	1.80e+04	1.10e+03	1.75e+02	**2.82e-02**
		Std	2.01e+04	1.35e+04	3.28e+03	1.29e+03	9.42e+01	**2.39e-02**
	50	Ave	1.38e+05	1.00e+05	6.78e+04	5.61e+03	1.94e+03	**1.21e+00**
		Std	4.53e+04	2.43e+04	6.28e+03	3.32e+03	5.33e+02	**9.35e-01**
f_2	10	Ave	3.56e+00	3.73e+00	1.67e+00	1.06e-02	2.48e+00	**3.49e-04**
		Std	8.78e-01	1.40e+00	3.10e-01	2.61e-03	1.12e+00	**1.79e-04**
	30	Ave	1.03e+01	9.33e+00	6.78e+00	1.91e+00	9.08e+00	**9.03e-02**
		Std	1.69e+00	3.00e-01	5.16e-01	8.41e-01	4.91e-01	**2.19e-02**
	50	Ave	1.35e+01	9.42e+00	8.74e+00	3.59e+00	9.72e+00	**6.65e-01**
		Std	1.62e+00	1.94e-01	3.83e-01	3.59e-01	6.15e-01	**1.85e-01**
f_3	10	Ave	3.95e+03	2.08e+03	8.55e+02	2.43e+02	4.43e+03	**4.16e-03**
		Std	2.91e+03	9.92e+02	2.96e+02	2.51e+02	1.74e+03	**4.64e-03**
	30	Ave	1.10e+05	5.35e+04	3.55e+04	1.25e+04	1.29e+05	**7.11e+02**
		Std	6.02e+04	1.33e+04	6.53e+03	5.75e+03	3.24e+04	**3.45e+02**
	50	Ave	4.11e+05	1.35e+05	1.17e+05	4.11e+04	3.65e+05	**1.24e+04**
		Std	1.90e+05	2.95e+04	2.23e+04	8.90e+03	1.05e+05	**3.75e+03**
f_4	10	Ave	6.20e+00	1.16e+01	6.80e+00	9.47e-02	4.21e+00	**1.15e-04**
		Std	1.97e+00	5.26e+00	1.43e+00	3.51e-01	1.15e+01	**8.25e-05**
	30	Ave	1.86e+02	4.39e+07	3.62e+03	2.98e+01	3.21e+06	**6.03e-01**
		Std	3.61e+02	1.25e+08	1.18e+04	1.58e+01	1.71e+07	**2.74e+00**
	50	Ave	1.31e+18	3.59e+17	2.30e+14	9.00e+01	3.70e+16	**5.07e+01**
		Std	5.52e+18	1.69e+18	6.68e+14	**2.29e+01**	1.97e+17	9.86e+01
f_5	10	Ave	1.35e-01	6.97e-01	7.00e-02	5.25e-03	5.54e-02	**3.12e-03**
		Std	1.09e-01	7.44e-01	2.92e-02	4.62e-03	2.77e-02	**1.76e-03**
	30	Ave	9.25e+01	6.26e+01	1.13e+01	4.67e-01	1.53e+00	**3.02e-02**
		Std	4.65e+01	3.85e+01	4.46e+00	3.65e-01	5.60e-01	**1.00e-02**
	50	Ave	7.49e+02	3.46e+02	9.59e+01	4.00e+00	6.68e+00	**1.30e-01**
		Std	3.39e+02	1.35e+02	2.88e+01	2.45e+00	2.89e+00	**5.25e-02**

表 6 测试函数的求解结果(多峰)

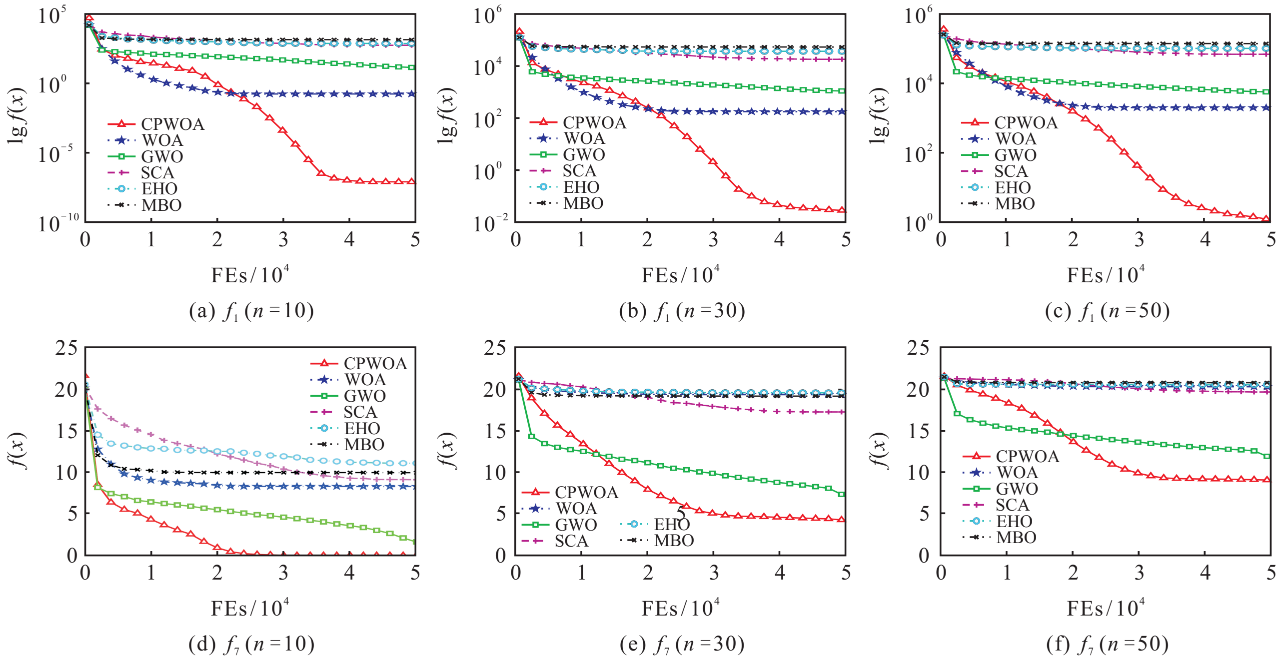
函数	n	指标	MBO	EHO	SCA	GWO	WOA	CPWOA
f_6	10	Ave	3.12e+07	1.90e+07	5.94e+06	3.67e+05	4.36e+05	**1.24e+01**
		Std	4.82e+07	3.10e+07	3.40e+06	5.48e+05	5.17e+05	**2.46e+01**
	30	Ave	2.64e+10	1.47e+10	2.48e+09	8.13e+07	8.00e+06	**5.21e+02**
		Std	2.26e+10	1.01e+10	1.14e+09	1.87e+08	5.71e+06	**7.03e+02**
	50	Ave	1.35e+11	6.10e+10	2.85e+10	5.42e+08	1.87e+08	**1.68e+03**
		Std	7.35e+10	2.01e+10	8.08e+09	7.29e+08	1.76e+08	**2.01e+03**
f_7	10	Ave	9.89e+00	1.10e+01	9.06e+00	1.62e+00	8.25e+00	**1.53e-04**
		Std	1.85e+00	3.78e+00	9.51e-01	1.94e+00	5.05e+00	**9.75e-05**
	30	Ave	1.92e+01	1.96e+01	1.73e+01	7.39e+00	1.94e+01	**4.32e+00**
		Std	8.94e-01	9.41e-01	**5.87e-01**	2.11e+00	7.27e-01	7.12e+00
	50	Ave	2.07e+01	2.05e+01	1.96e+01	1.19e+01	2.03e+01	**9.05e+00**
		Std	3.28e-01	**1.39e-01**	5.66e-01	1.95e+00	2.83e-01	8.31e+00
f_8	10	Ave	2.29e+01	2.64e+01	7.43e+00	6.28e-01	9.64e-01	**1.69e-01**
		Std	1.18e+01	2.67e+01	2.08e+00	2.38e-01	4.49e-01	**1.01e-01**
	30	Ave	5.48e+02	3.71e+02	1.85e+02	1.04e+01	2.89e+00	**4.23e-01**
		Std	1.37e+02	1.42e+02	3.94e+01	1.06e+01	1.61e+00	**1.17e-01**
	50	Ave	1.11e+03	9.09e+02	5.16e+02	6.02e+01	1.60e+01	**6.64e-01**
		Std	2.59e+02	1.82e+02	6.08e+01	3.85e+01	4.61e+00	**1.66e-01**
f_9	10	Ave	2.50e+01	5.08e+01	3.78e+01	9.65e+00	4.75e+01	**4.81e+00**
		Std	6.28e+00	1.38e+01	6.07e+00	5.86e+00	1.80e+01	**2.45e+00**
	30	Ave	2.71e+02	3.52e+02	2.79e+02	9.35e+01	2.61e+02	**5.76e+01**
		Std	4.87e+01	3.73e+01	1.86e+01	3.02e+01	4.66e+01	**1.40e+01**
	50	Ave	7.26e+02	6.86e+02	5.77e+02	2.15e+02	4.90e+02	**1.69e+02**
		Std	9.51e+01	5.16e+01	3.79e+01	4.28e+01	6.67e+01	**2.69e+01**
f_{10}	10	Ave	9.21e-01	4.71e+00	2.82e+00	1.60e-01	2.94e+00	**5.28e-03**
		Std	3.39e-01	1.60e+00	7.75e-01	2.04e-01	1.95e+00	**7.11e-03**
	30	Ave	4.34e+01	4.55e+01	3.57e+01	3.64e+00	3.64e+01	**2.97e+00**
		Std	9.87e+00	7.95e+00	3.34e+00	**1.70e+00**	8.96e+00	1.85e+00
	50	Ave	1.04e+02	8.70e+01	8.12e+01	1.25e+01	7.42e+01	**1.07e+01**
		Std	1.93e+01	1.18e+01	7.73e+00	4.80e+00	1.75e+01	**4.36e+00**

表 7 测试函数的求解结果(固定维)

函数	指标	MBO	EHO	SCA	GWO	WOA	CPWOA
f_{11}	Ave	1.03e+00	1.20e+00	1.33e+00	3.48e+00	1.59e+00	**9.98e-01**
	Std	1.81e-01	4.49e-01	7.52e-01	3.62e+00	1.86e+00	**3.13e-16**
f_{12}	Ave	1.98e-03	4.11e-04	8.35e-04	2.98e-03	6.49e-04	**3.44e-04**
	Std	5.12e-03	2.16e-04	4.12e-04	6.93e-03	3.31e-04	**1.67e-04**
f_{13}	Ave	**3.98e-01**	**3.98e-01**	**3.98e-01**	**3.98e-01**	**3.98e-01**	**3.98e-01**
	Std	7.49e-05	7.63e-04	3.44e-04	2.14e-07	1.04e-07	**0.00e+00**
f_{14}	Ave	−9.67e-01	−6.54e-01	**−1.00e+00**	**−1.00e+00**	**−1.00e+00**	**−1.00e+00**
	Std	1.83e-01	3.36e-01	4.08e-04	1.12e-07	1.89e-08	**0.00e+00**
f_{15}	Ave	**−3.29e+00**	−3.25e+00	−2.92e+00	−3.28e+00	−3.26e+00	**−3.29e+00**
	Std	5.55e-02	6.58e-02	4.35e-01	7.59e-02	7.49e-02	**5.54e-02**

由图1可知:求解 f_1 和 f_7 时,EHO、MBO、SCA和WOA均过早收敛到局部最优解,而CPWOA随着迭代进行继续向目标解收敛,且收敛速度比GWO快.这主要是因为控制因子 a 和权值 w 按照余弦曲线在第一象限非线性变化,先减小了算法的收敛速度以进行充分全局探索,然后再加速收敛,最后由于 w 很小,算法在当前最优解附近进行微调.

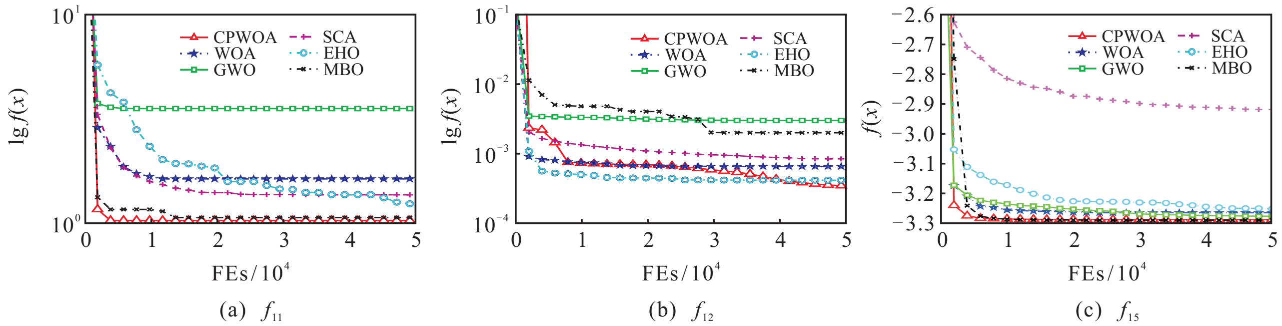
由表7和图2可知:由于这些固定维函数的解需要调整的维度少,各算法求解精度和稳定性相对较高.其中:各算法对于 f_{13} 的求解均能收敛到最优点,



(a) f_1 ($n=10$)　(b) f_1 ($n=30$)　(c) f_1 ($n=50$)

(d) f_7 ($n=10$)　(e) f_7 ($n=30$)　(f) f_7 ($n=50$)

图 1　单峰、多峰函数求解收敛曲线



(a) f_{11}　(b) f_{12}　(c) f_{15}

图 2　固定维函数求解收敛曲线

但CPWOA的标准差更小,鲁棒性更好;在求解f_{11}时,GWO、MBO、EHO、SCA和WOA均频繁陷入局部最优解导致收敛曲线偏高;CPWOA通过合理控制收敛速度以及变异,提高了算法跳出局部最优解的能力.

3.3　实验结果的统计分析

为验证本文提出的CPWOA算法与其他算法的显著性差异,采用一种非参数估计方法——Wilcoxon秩和检验,进行统计分析. 对于每个测试函数,将CPWOA的30次求解结果分别与MBO、EHO、SCA、GWO和WOA的30次求解结果进行假设检验. 其中:假设H_0时两种算法的平均值之间没有明显差异,H_1时两种算法的平均值之间有明显差异. 设置零假设检验显著性水平阈值$\alpha = 0.05$. 当检验结果$P < 0.05$时,拒绝零假设,则两种算法的实验结果具有显著性差异;当检验结果$P > 0.05$时,接受零假设,则两种算法的实验结果没有显著性差异.

表8展示了$n = 10$的高维测试函数$f_1 \sim f_{10}$及固定维测试函数$f_{11} \sim f_{15}$的检验结果. 其中:对小于显著性水平阈值0.05的概率P值加粗显示;"W"为显著性检验判断结果;结合表5～表7,用"+"表示CPWOA相对其他算法求解精度更高并具有显著性;用"-"表示CPWOA相对其他算法求解精度更低并具有显著性;用"="表示CPWOA算法与其他算法求解精度无显著差异;"N/A"表示无法进行显著性判断. 从检验结果来看,绝大多数检验P值小于0.05且"W"为"+",拒绝零假设. 此外,对$n = 30$和$n = 50$下$f_1 \sim f_{10}$的计算结果进行统计检验,绝大多数结果均为拒绝零假设. 因此,总体上,CPWOA与MBO、EHO、SCA、GWO、WOA的计算结果之间具有显著差异,且CPWOA显著更优.

3.4　与其他改进型WOA的比较

为进一步体现CPWOA算法的优越性,将它与AWOA[5]、WOAWC[9]、MWOA[10]和LWOA[16]进行对比. 为了公平比较,采用表1～表3测试函数作数值实验,实验参数设置参照表4,各算法的其他参数采用其文献中的实验设定值.

表 8 Wilcoxon秩和检验结果

测试函数	CPWOA vs MBO		CPWOA vs EHO		CPWOA vs SCA		CPWOA vs GWO		CPWOA vs WOA	
	P	W	P	W	P	W	P	W	P	W
$f_1\,(n=10)$	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+
$f_2\,(n=10)$	**2.63e-11**	+	**3.02e-11**	+	**2.95e-11**	+	**3.02e-11**	+	**3.02e-11**	+
$f_3\,(n=10)$	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+	**3.69e-11**	+	**3.02e-11**	+
$f_4\,(n=10)$	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+
$f_5\,(n=10)$	**3.02e-11**	+	**3.34e-11**	+	**3.02e-11**	+	4.12e-01	=	**3.02e-11**	+
$f_6\,(n=10)$	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+	**1.47e-07**	+	**6.07e-11**	+
$f_7\,(n=10)$	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+
$f_8\,(n=10)$	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+	**1.69e-09**	+	**4.98e-11**	+
$f_9\,(n=10)$	**2.95e-11**	+	**2.95e-11**	+	**2.95e-11**	+	**2.74e-05**	+	**3.26e-11**	+
$f_{10}\,(n=10)$	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+	**4.11e-07**	+	**3.02e-11**	+
f_{11}	3.34e-01	=	**1.91e-10**	+	**3.41e-07**	+	**1.90e-06**	+	**2.15e-02**	+
f_{12}	**8.33e-05**	+	**1.45e-07**	+	**2.82e-10**	+	**6.26e-05**	+	**1.41e-08**	+
f_{13}	**2.92e-05**	+	**1.65e-11**	+	**1.20e-12**	+	N/A	N/A	N/A	N/A
f_{14}	1.60e-01	=	**1.21e-12**	+	**1.21e-12**	+	N/A	N/A	N/A	N/A
f_{15}	**2.52e-04**	+	**4.08e-04**	+	**1.01e-11**	+	5.16e-01	=	**1.55e-04**	+

表 9 CPWOA算法与其他改进型WOA算法的计算结果比较

函数	指标	WOA	MWOA	AWOA	WOAWC	LWOA	CPWOA
$f_1\,(n=10)$	Ave	1.97e-01	5.85e+03	6.75e+03	6.63e+02	2.18e-03	**9.08e-08**
	Std	2.22e-01	1.30e+03	2.20e+03	1.10e+03	7.76e-04	**1.39e-07**
$f_2\,(n=10)$	Ave	2.48e+00	7.11e+00	7.21e+00	5.50e+00	2.94e-02	**3.49e-04**
	Std	1.12e+00	9.74e-01	1.09e+00	2.01e+00	2.05e-02	**1.79e-04**
$f_3\,(n=10)$	Ave	4.43e+03	6.80e+03	6.80e+03	7.00e+03	1.33e-01	**4.16e-03**
	Std	1.74e+03	1.39e+03	1.55e+03	2.09e+03	8.21e-02	**4.64e-03**
$f_4\,(n=10)$	Ave	4.21e+00	4.23e+01	4.62e+01	4.27e+01	1.04e+00	**1.15e-04**
	Std	1.15e+01	1.01e+01	1.79e+01	1.62e+01	2.05e+00	**8.25e-05**
$f_5\,(n=10)$	Ave	5.54e-02	2.81e+00	4.24e+00	1.46e+00	8.94e-03	**3.12e-03**
	Std	2.77e-02	1.47e+00	1.72e+00	1.36e+00	5.99e-03	**1.76e-03**
$f_6\,(n=10)$	Ave	4.36e+05	2.20e+08	2.21e+08	8.38e+06	**5.94e+00**	1.24e+01
	Std	5.17e+05	4.26e+07	4.77e+07	1.50e+07	**2.97e+00**	2.46e+01
$f_7\,(n=10)$	Ave	8.25e+00	1.72e+01	1.74e+01	1.69e+01	1.96e+00	**1.53e-04**
	Std	5.05e+00	6.07e-01	5.25e-01	2.01e+00	3.49e+00	**9.75e-05**
$f_8\,(n=10)$	Ave	9.64e-01	1.24e+02	1.47e+02	1.19e+01	3.94e-01	**1.69e-01**
	Std	4.49e-01	3.14e+01	3.87e+01	8.46e+00	2.24e-01	**1.01e-01**
$f_9\,(n=10)$	Ave	4.75e+01	6.62e+01	6.76e+01	6.50e+01	8.37e+00	**4.81e+00**
	Std	1.80e+01	1.31e+01	1.14e+01	2.05e+01	3.33e+00	**2.45e+00**
$f_{10}\,(n=10)$	Ave	2.94e+00	5.95e+00	5.49e+00	4.75e+00	1.66e+00	**5.28e-03**
	Std	1.95e+00	1.81e+00	1.89e+00	2.24e+00	1.16e+00	**7.11e-03**
f_{11}	Ave	1.59e+00	1.16e+00	1.06e+00	1.62e+00	**9.98e-01**	**9.98e-01**
	Std	1.86e+00	5.27e-01	2.52e-01	1.28e+00	3.31e-12	**3.13e-16**
f_{12}	Ave	6.49e-04	3.49e-04	**3.37e-04**	4.43e-04	6.38e-04	3.44e-04
	Std	3.31e-04	3.69e-05	**3.31e-05**	1.63e-04	3.68e-04	1.67e-04
f_{13}	Ave	**3.98e-01**	**3.98e-01**	**3.98e-01**	**3.98e-01**	**3.98e-01**	**3.98e-01**
	Std	1.04e-07	1.13e-04	2.19e-04	6.32e-07	2.13e-06	**0.00e+00**
f_{14}	Ave	**−1.00e+00**	-8.00e-01	**−1.00e+00**	**−1.00e+00**	**−1.00e+00**	**−1.00e+00**
	Std	1.89e-08	4.07e-01	4.87e-04	3.32e-05	1.91e-07	**0.00e+00**
f_{15}	Ave	−3.26e+00	−3.25e+00	−3.16e+00	−3.13e+00	−3.16e+00	**−3.29e+00**
	Std	7.49e-02	6.26e-02	1.18e-01	1.24e-01	1.28e-01	**5.54e-02**

表9展示了各改进型WOA对$n = 10$下的高维测试函数及固定维测试函数的30次计算结果的平均值和标准差. 除了f_6和f_{12}以外, CPWOA对其余测试函数计算结果精度及稳定性最高. MWOA、AWOA和WOAWC对$f_1 \sim f_5$的计算结果均比WOA的更差. 此外, 对$n = 30$和$n = 50$进行实验, 结果显示, 除了f_8以外, 其余计算结果均为CPWOA最优.

图3展示了各改进型WOA的计算收敛情况. 对于f_1和f_7, MWOA、AWOA、WOAWC因在原点附近具有强烈吸引, 均过早收敛而陷入局部最优解, 且比原始WOA差, 而CPWOA的局部搜索策略不同, 避免了原点的吸引. LWOA和CPWOA比WOA收敛精度更高, 且CPWOA更优. 对于较低维度的f_{15}, 各算法均较早收敛, 但CPWOA收敛精度更高.

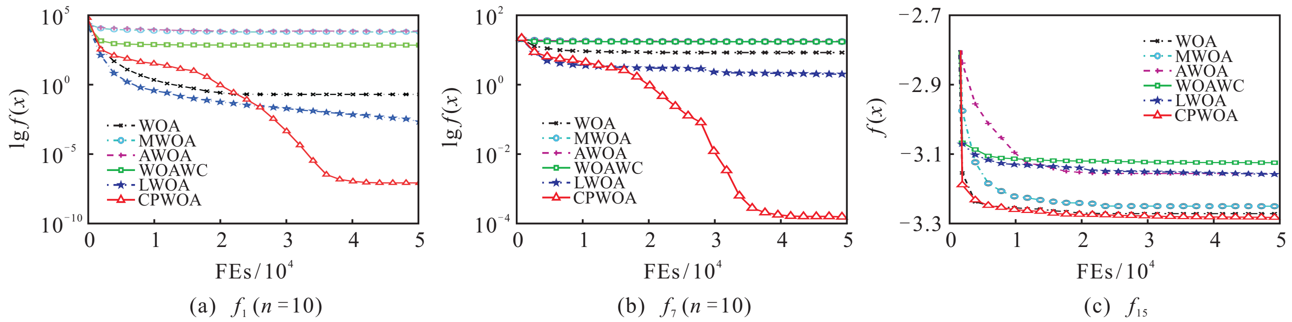


图 3　改进型WOA对测试函数求解收敛图

为比较各改进型WOA算法差异的显著性, 按照3.3节的方法进行统计检验, 结果如表10所示. LWOA对f_6的计算结果平均值和标准差更优, 但Wilcoxon秩和检验结果表明这种差异是不显著的. 对于固定维测试函数, 除少部分计算结果无显著差异或无法检验以外, 其余大部分计算结果均具有显著差异. 此外, 对$n = 30$和$n = 50$下$f_1 \sim f_{10}$的计算结果进行统计检验, 绝大多数结果均为拒绝零假设. 结合比较各改进型WOA的平均值和标准差来看, CPWOA的绝大多数计算结果显著更优.

表 10　Wilcoxon秩和检验结果(与其他改进型WOA比较)

测试函数	CPWOA vs NWOA		CPWOA vs AWOA		CPWOA vs WOAWC		CPWOA vs LWOA	
	P	W	P	W	P	W	P	W
$f_1\ (n = 10)$	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+
$f_2\ (n = 10)$	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+
$f_3\ (n = 10)$	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+
$f_4\ (n = 10)$	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+
$f_5\ (n = 10)$	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+	**9.83e-08**	+
$f_6\ (n = 10)$	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+	2.71e-01	=
$f_7\ (n = 10)$	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+
$f_8\ (n = 10)$	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+	**1.49e-04**	+
$f_9\ (n = 10)$	**2.95e-11**	+	**2.90e-11**	+	**2.95e-11**	+	**5.79e-06**	+
$f_{10}\ (n = 10)$	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+	**3.02e-11**	+
f_{11}	8.15e-02	=	1.61e-01	=	**5.20e-06**	+	N/A	N/A
f_{12}	**1.25e-04**	+	**1.12e-02**	-	**3.83e-06**	+	**6.04e-07**	+
f_{13}	**3.41e-07**	+	**6.18e-10**	+	N/A	N/A	3.34e-01	=
f_{14}	**1.66e-08**	+	**1.64e-11**	+	**8.76e-07**	+	N/A	N/A
f_{15}	**4.96e-05**	+	**4.31e-07**	+	**1.09e-07**	+	**9.18e-08**	+

4　结　论

鲸鱼优化算法(WOA)作为一种新型群体智能优化算法, 在提高收敛精度和跳出局部最优解的性能方面仍具有较大的优化空间. 本文提出了一种基于余弦控制因子和多项式变异的鲸鱼优化算法(CPWOA), 对控制参数按照余弦规律变化作动态调整, 对种群位置更新加入与控制参数同步变化的惯性因子, 从而较合理地协调算法的全局探索与局部搜索. 在算法中还对最优解进行多项式变异, 从而进一步提升了其跳出局部最优的能力. 实验显示, 所提出的CPWOA算法对绝大多数最优解偏离原点的目标函数的求解结果, 比MBO、EHO、SCA、GWO、WOA以及其他改进型WOA的求解精度和稳定性都更高, 且统计检验表明了这种差异的显著性. 对于鲸鱼优

化算法对特定问题求解能力的进一步提高,以及针对工程实际的更好应用将是下一步工作的主要内容.

参考文献(References)

[1] Mirjalili S, Lewis A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95(5): 51-67.

[2] Jangir P, Trivedi I N, Bhesdadiya R H, et al. Training multi-layer perceptron in neural network using whale optimization algorithm[J]. Indian J of Science & Technology, 2016, 9(19): 28-36.

[3] Khaled M, Samir S, Abdelghani B. Whale optimization algorithm based optimal reactive power dispatch: A case study of the Algerian power system[J]. Electric Power Systems Research, 2017, 163: 696-705.

[4] Prakash D B, Lakshminarayana C. Optimal siting of capacitors in radial distribution network using whale optimization algorithm[J]. Alexandria Engineering J, 2017, 56(4): 499-509.

[5] 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.)

[6] Oliva D, Abd El A M, Ella H A. Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm[J]. Applied Energy, 2017, 200: 141-154.

[7] Long W, Jiao J, Liang X, et al. An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization[J]. Engineering Applications of Artificial Intelligence, 2018, 68(C): 63-80.

[8] 张顶学, 关治洪, 刘新芝. 一种动态改变惯性权重的自适应粒子群算法[J]. 控制与决策, 2008, 23(11): 1253-1257.
(Zhang D X, Guan Z H, Liu X Z. Adaptive particle swarm optimization algorithm with dynamically changing inertia weight[J]. Control and Decision, 2008, 23(11): 1253-1257.)

[9] 郭振洲, 王平, 马云峰, 等. 基于自适应权重和柯西变异的鲸鱼优化算法[J]. 微电子学与计算机, 2017, 34(9): 20-25.
(Guo Z Z, Wang P, Ma Y F, et al. Whale optimization algorithm based on adaptive weight and cauchy mutation[J]. Micro Electronics & Computer, 2017, 34(9): 20-25.)

[10] 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.
(Zhang Y, Chen F. A modified whale optimization algorithm[J]. Computer Engineering, 2018, 44(3): 208-213.)

[11] Zeng G Q, Chen J, Li L M, et al. An improved multi-objective population-based extremal optimization algorithm with polynomial mutation[J]. Information Sciences, 2016, 330(C): 49-73.

[12] Wang G G, Suash D, Gao X Z, et al. A new metaheuristic optimisation algorithm motivated by elephant herding behaviour[J]. Int J of Bio-Inspired Computation, 2017, 8(6): 394-409.

[13] Wang G G, Suash D, Cui Z. Monarch butterfly optimization[J]. Neural Computing and Applications, 2015, DOI:10.1007/s00521-015-1923-y.

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

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

[16] Ling Y, Zhou Y, Luo Q. Lévy flight trajectory-based whale optimization algorithm for global optimization[J]. IEEE Access, 2017, 5(99): 6168-6186.

作者简介

黄清宝(1979−), 男, 副教授, 博士生, 从事智能计算与机器学习的研究, E-mail: qingbaohuang@163.com;

李俊兴(1990−), 男, 硕士生, 从事嵌入式控制系统与智能计算的研究, E-mail: 1220415874@qq.com;

宋春宁(1969−), 男, 副教授, 博士, 从事嵌入式控制系统等研究, E-mail: scn206@gux.edu.cn;

徐辰华(1975−), 女, 副教授, 博士, 从事复杂系统建模与智能优化算法等研究, E-mail: xchhelen@163.com;

林小峰(1955−), 男, 教授, 博士生导师, 从事智能控制等研究, E-mail: gxulinxf@163.com.

(责任编辑:齐　霁)