

引用格式:何庆,魏康园,徐钦帅. 求解函数优化问题的改进鲸鱼优化算法[J]. 微电子学与计算机,2019,36(4):72-77. [HE Q, WEI K Y, XU Q S. An enhanced whale optimization algorithm for the problems of function optimization [J]. Microelectron-ics & Computer,2019,36(4): 72-77.]

求解函数优化问题的改进鲸鱼优化算法

何 庆^{1,2},魏康园¹,徐钦帅¹

(1 贵州大学 大数据与信息工程学院,贵州 贵阳 550025;

2 贵州大学 贵州省公共大数据重点实验室,贵州 贵阳 550025)

摘 要:针对鲸鱼优化算法(WOA)易陷入局部最优、寻优精度低等问题,提出一种改进的鲸鱼优化算法(EWOA).首先,将自适应策略引入鲸鱼位置更新公式中,以便平衡算法全局探索和局部开发能力的同时,加快算法收敛速度、提高算法的寻优精度;然后,引入差分变异思想,对较优的鲸鱼位置进行变异操作以避免算法陷入局部最优,防止早熟收敛现象;最后,通过对 9 个基准测试函数在固定参数和不同维度的实验表明,改进算法在寻优精度和收敛速度比传统算法均有显著提高,尤其在高维函数的优化问题中表现出更好的收敛性能。

关键词:鲸鱼优化算法;自适应策略;差分变异;函数优化

中图分类号: TP301.6

文献标识码: A

文章编号: 1000—7180(2019)04—0072—06

An Enhanced Whale Optimization Algorithm for the Problems of Function Optimization

HE Qing^{1,2}, WEI Kang-yuan¹, XU Qin-shuai¹

(1 College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China;

2 Guizhou Provincial Key Laboratory of Public Big Data,Guizhou University, Guiyang 550025, China)

Abstract: To resolve the problem that the whale optimization algorithm (WOA) is easy to fall into local optimum and low precision, an enhanced whale optimization algorithm (EWOA) is proposed. Firstly, the adaptive strategy was introduced into the whale's position to balance the global exploration and local exploitation capabilities of the algorithm, speed up the convergence of the algorithm, and improve the optimization accuracy of the algorithm. Then, to avoid the algorithm falling into local optimum and prevent premature convergence, the idea of differential mutation was introduced to mutate the better whale's position. Finally, the experimental results on nine test functions under fixed parameters and different dimensions show that the improved algorithm has significantly improved search precision and convergence speed compared with the traditional WOA. Especially in the optimization problem of high-dimensional functions, the improved algorithm shows better convergence performance than the traditional WOA and its variants.

Key words: whale optimization algorithm; adaptive strategy; differential mutation; function optimization

收稿日期: 2018—08—03; 修回日期: 2018—10—10

基金项目: 贵州省科技计划项目重大专项(黔科合重大专项字[2018]3002); 贵州省公共大数据重点实验室开放课题(2017BDKFJJ004); 贵州省教育厅青年科技人才成长项目(黔科合 KY 字[2016]124); 贵州大学培育项目(黔科合平台人才[2017]5788); 贵州省科技计划项目重大专项(黔科合重大专项字[2016]3022)

1 引言

鲸鱼优化算法^[1] (Whale Optimization Algorithm, WOA)是由 Mirjalili 等于 2016 年提出的基于鲸鱼捕食行为的一种优化算法. 研究发现,与粒子群优化算法(Particle Swarm Optimization, PSO)和引力搜索算法(Gravitational Search Algorithms, GSA)等优化算法相比,传统 WOA 算法具有明显较优的寻优精度. 但是,与其他元启发式优化算法相类似,传统 WOA 算法同样存在收敛速度慢、早熟收敛、易陷入局部最优等问题,降低了算法的寻优能力. 基于此,近年来国内外学者实现了许多有效的改进 WOA 算法. 如 Kaur 等^[2]将混沌理论引入 WOA 算法中来提供算法收敛速度,同时利用混沌 WOA 算法调整参数 p ,平衡算法探索和开发能力,提出一种 CWOA 算法,并在基准测试函数的实验验证了算法的有效性;闫旭等^[3]将量子计算思想引入 WOA 算法,提出一种 QWOA 算法,并通过计算复杂度、全局收敛性的仿真实验,验证该算法具有较好的寻优精度和全局搜搜能力;张永等^[4]首先利用分段 Logistic 混沌映射对种群进行初始化,然后引入分段非线性自适应权重平衡算法全局探索和局部开发能力,提出了 MWOA 算法,并在 6 个基准测试函数的测试,验证了算法具有较好的收敛速度和更优的寻优精度;龙文等^[5]首先利用对立学习策略进行种群位置初始化,然后提出非线性收敛因子,协调算法探索和开发能力,最后引入多样性变异操作,提出一种 IWOA 算法,应用于大规模复杂优化问题中具有较好的性能;郭振洲等^[6]首先引入柯西逆累积分布函数对鲸鱼位置进行变异,同时引入自适应权重对鲸鱼局部搜索能力进行改进,提出一种 WOAWC 算法,通过实验表明,该算法的收敛精度和算法稳定性具有较好的改善. 然而,尽管已有的研究对传统 WOA 算法的寻优结果有所改进,但对于算法的收敛速度慢和寻优精度不高问题仍需要深入研究.

本文针对鲸鱼优化算法寻优精度低、易陷入局部最优等问题,提出一种改进的鲸鱼优化算法(Enhanced Whale Optimization Algorithm, EWOA).

2 鲸鱼优化算法

鲸鱼优化算法是一种元启发式优化算法,其主要模拟鲸鱼捕食行为,即泡泡网捕食方法^[7]. 其

原理为:鲸鱼发现猎物后,沿螺旋路径制造泡泡网向上游去进行捕食,分为包围猎物、泡泡网攻击、搜寻猎物三个阶段. 其中在 WOA 算法中,假设鲸鱼种群规模表示为 N ,搜索空间的维度为 Dim ,则第 i 只鲸鱼在第 d 维的位置可以表示为: $X_i = (X_i^1, X_i^2, \dots, X_i^{Dim})$, $i = 1, 2, \dots, N$, 猎物的位置代表问题的全局最优解.

(1) 包围猎物

在 WOA 算法中,鲸鱼识别猎物的位置,并将它们包围起来,然而鲸鱼无法提前获知猎物的位置. 因此,假设当前的最优位置是目标猎物,群体中的其他个体均向最优位置移动,利用如下公式更新位置:

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

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

式中, t 表示为当前迭代次数; $\vec{X}^*(t) = (X_1^*, X_2^*, \dots, X_{Dim}^*)$ 表示猎物位置; $\vec{X}(t)$ 表示当前鲸鱼位置; $\vec{A} \cdot \vec{D}$ 表示包围步长,其中 $\vec{A}$ 和 $\vec{C}$ 定义如下:

$$\vec{A} = 2\vec{a} \cdot \text{rand} - \vec{a} \quad (3)$$

$$\vec{C} = 2 \cdot \text{rand} \quad (4)$$

式中, rand 表示 $[0, 1]$ 之间的随机数; $\vec{a}$ 表示随着迭代次数的增加从 2 线性递减到 0 的收敛因子,表示为:

$$a = \left(2 - \frac{2t}{T_{\max}}\right) \quad (5)$$

式中, t 表示为当前迭代次数; $T_{\max}$ 表示为最大迭代次数.

(2) 泡泡网攻击

在 WOA 算法中,设计两种方法来描述鲸鱼的捕食行为,分别为:收缩包围机制、螺旋更新位置.

a. 收缩包围机制:通过减小公式(3)和(5)中的收敛因子 a 来实现.

b. 螺旋更新位置:首先计算鲸鱼个体与当前最优位置之间的距离,然后模拟鲸鱼以螺旋的方式捕获食物,其数学模型可以表示为:

$$\vec{X}(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \quad (6)$$

式中, $\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)|$ 表示第 i 只鲸鱼和当前最优位置之间的距离; b 是用来限定对数螺旋形式的常量系数,文本取 1; l 表示 $[-1, 1]$ 之间的随机数.

值得注意的是,鲸鱼以螺旋形式包围猎物的同

时,还需收缩包围圈,因此,为了实现这种同步模型,则选择相同概率 p 进行收缩包围机制和螺旋位置更新,其数学模型表示如下:

$$\vec{X}(t+1)=$$

$$\begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D} & \text{if } p < 0.5 \\ \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) & \text{if } p \geq 0.5 \end{cases} \quad (7)$$

(3) 搜寻猎物

当 $|A| \geq 1$ 时,随机选择鲸鱼迫使其远离参考鲸鱼,来找到一个更优的猎物,以便增强算法的全局探索能力,其数学模型表示如下:

$$\vec{D} = |\vec{C} \cdot \vec{X}_{\text{rand}} - \vec{X}(t)| \quad (8)$$

$$\vec{X}(t+1) = \vec{X}_{\text{rand}}(t) - \vec{A} \cdot \vec{D} \quad (9)$$

式中, $\vec{X}_{\text{rand}}(t)$ 表示随机选取鲸鱼的位置向量。

3 改进的鲸鱼优化算法

3.1 自适应非线性位置更新

与其他元启发式优化算法相似,在 WOA 中如何平衡算法的全局探索与局部开发能力至关重要。通过对传统 WOA 算法分析可知, $\vec{A} \cdot \vec{D}$ 为包围步长,其变化主要取决于收敛因子 $\vec{a}$,但在传统 WOA 算法中收敛因子 $\vec{a}$ 是随着迭代次数的增加从 2 线性递减到 0,这将使得算法收敛速度过慢。而且,传统 WOA 算法的位置更新公式(2)、(6)中均有 $\vec{X}^*(t)$ 表示猎物位置,即当前最优解;公式(9)中有 $\vec{X}_{\text{rand}}(t)$ 表示随机选取鲸鱼的位置,随着迭代次数的增加均未被充分利用。因此,结合粒子群优化算法中自适应惯性权重引导种群寻优思想^[8],本文将自适应策略引入 WOA 算法的位置更新公式中,加快算法收敛速度、提高算法的寻优精度。具体公式如下:

$$\vec{X}(t+1) = S1 * \vec{X}^*(t) - S2 * \vec{A} \cdot \vec{D}, \quad |A| < 1 \quad p < 0.5 \quad (10)$$

$$\vec{X}(t+1) = S1 * \vec{X}_{\text{rand}}(t) - S2 * \vec{A} \cdot \vec{D}, \quad |A| \geq 1 \quad p < 0.5 \quad (11)$$

$$\vec{X}(t+1) = S1 * \vec{X}^*(t) + \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l), \quad p \geq 0.5 \quad (12)$$

式中, $S1$ 和 $S2$ 分别为当前最优位置和包围步长的自适应调整系数,具体公式表示为:

$$S1 = -\gamma \cdot \left[\cos\left(\pi \cdot \frac{t}{T}\right) - \lambda \right] \quad (13)$$

$$S2 = \gamma \cdot \left[\cos\left(\pi \cdot \frac{t}{T}\right) + \lambda \right] \quad (14)$$

式中, γ 表示 $S1$ 和 $S2$ 变化取值范围; λ 表示 $S1$ 和 $S2$ 取值步长,其取值分别为 $\gamma = \frac{1}{2}, \lambda = 1$ 。

从式(13)、(14)可以看出,自适应调整系数 $S1$ 随着算法迭代进化呈非线性增大趋势,使种群能够充分朝向精英猎物位置移动;而 $S2$ 随着算法迭代进化而不断减小,使种群在进化后期具有较小的包围步长而加快收敛速度,从而实现算法全局探索与局部搜索能力的平衡,同时加快算法收敛速度,提高算法的寻优精度。

3.2 引入差分变异操作

通过对传统 WOA 算法的原理分析可知,随着算法的迭代进化,鲸鱼个体将逐渐向种群中适应度较优个体靠拢,易使得鲸鱼个体分布收缩,从而导致算法陷入局部最优。因此,本文针对这一问题,引入差分变异思想,改善算法易陷入局部最优及出现早熟收敛现象。

差分进化算法^[9] (Differential Evolution, DE) 由 Storn 等于 1997 年提出的一种模拟生物进化的优化算法,具有良好的全局收敛性、参数少、寻优结果稳定等优点^[10]。本文引入差分变异思想具体描述如下:

(1) 变异操作

选取当前种群中较优的鲸鱼个体进行变异,能够有效扩大算法的搜索域,避免算法陷入局部最优。具体操作如下:

$$\vec{V}_i(t+1) = \vec{X}_i(t) + F(\vec{X}_{r1}(t) - \vec{X}_{r2}(t)) \quad (15)$$

式中, $\vec{V}_i(t+1)$ 表示变异后的第 i 只鲸鱼的位置; F 表示缩放比例因子; $\vec{X}_{r1}(t) - \vec{X}_{r2}(t)$ 表示当代鲸鱼位置的差异向量。

(2) 选择操作

选择操作是将变异后的新个体与原始个体进行比较,判断其适应度值是否较优,如果较优,则保留,否则舍弃。差分进化算法中通常采用贪婪选择,具体公式如下:

$$\vec{X}_i(t+1) = \begin{cases} \vec{V}_i(t+1) & \text{fit}(\vec{V}_i(t+1)) < \text{fit}(\vec{X}_i(t)) \\ \vec{X}_i(t) & \text{fit}(\vec{V}_i(t+1)) \geq \text{fit}(\vec{X}_i(t)) \end{cases} \quad (16)$$

改进的鲸鱼优化算法流程图如图 1 所示。

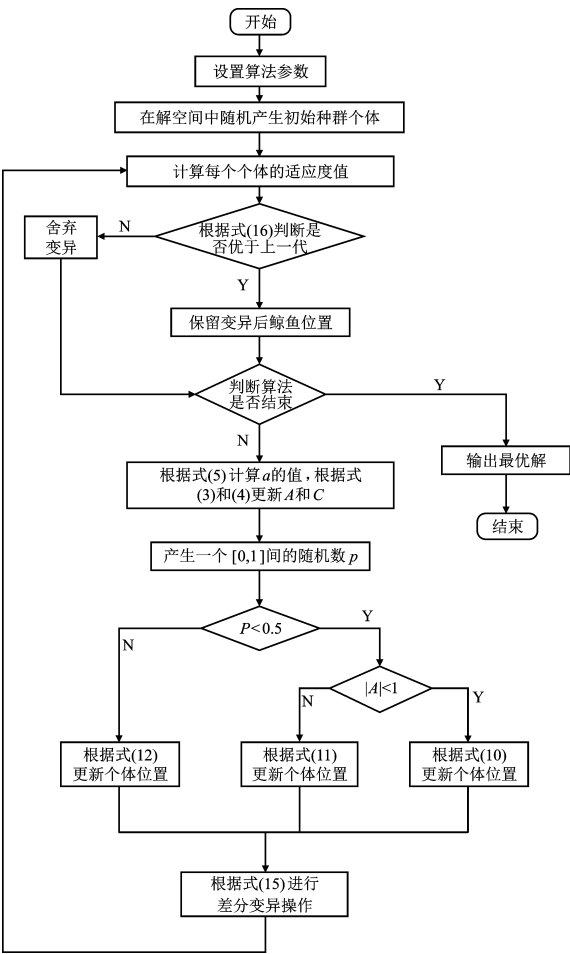


图 1 EWOA 算法流程图

4 实验结果与分析

为了测试本文提出的 EWOA 算法中改进策略的有效性,选取 9 个基准测试函数进行数值仿真实验,并与 GSA 算法、传统 WOA 算法、IWOA^[5]算法和 WOAWC^[6]算法进行对比. 本文所采用的 9 个基准测试函数如表 1 所示,其中 F_1 — F_5 表示单模态函数, F_6 — F_9 表示复杂的非线性多模态函数,形态各异的测试函数能有效检验算法的优化性能.

本文从三个方面对算法进行函数优化测试:(1)当种群规模、迭代次数及维度等参数固定时对算法进行测试,并与 GSA 算法、传统 WOA 算法进行比较,观察算法的收敛速度和寻优精度;(2)当其他参数固定时,比较 EWOA 算法在不同维度下的优化性能;(3)将 EWOA 算法与参考文献[5]和[6]中的算法进行比较,以检验算法的性能.

4.1 参数固定时算法性能测试

通过对比 EWOA 算法与传统 WOA 算法、GSA 算法的寻优精度和收敛速度,以测试算法性能. 参比算法的相关参数设置如表 2 所示. 表 3 展示了三种算法分别对选用的 9 个基准测试函数分别独立运行 30 次之后,取得的平均值 Mean、最小值 Best 和标准差 Std. Dev. 图 2 展示了三种算法优化基准测试函数的收敛曲线图,可以更加直观地反映算法的收敛过程.

表 1 基准测试函数

函数名	表达式	维度	范围	理论最优
Sphere	$F_1 = \sum_{i=1}^{Dim} x_i^2$	10/30/100/200	$[-100, 100]$	0
Schwefel 2, 22	$F_2 = \sum_{i=1}^{Dim} x_i + \prod_{i=1}^{Dim} x_i $	10/30/100/200	$[-10, 10]$	0
Schwefel 1. 2	$F_3 = \sum_{i=1}^{Dim} (\sum_{j=1}^i x_j)^2$	10/30/100/200	$[-100, 100]$	0
Schwefel 2. 21	$F_4 = \max_i \{ x_i , 1 \leq i \leq D \}$	10/30/100/200	$[-100, 100]$	0
Quartic	$F_5 = \sum_{i=1}^{Dim} i \cdot x_i^4 + \text{random}[0, 1)$	10/30/100/200	$[-1. 28, 1. 28]$	0
Schwefel 2. 26	$F_6 = \sum_{i=1}^{Dim} -x_i \sin(\sqrt{x_i})$	10/30/100/200	$[-500, 500]$	-1. 2569. 5
Rastrigin	$F_7 = \sum_{i=1}^{Dim} [x_i^2 - 10 \cos(2\pi x_i) + 10]$	10/30/100/200	$[-5. 12, 5. 12]$	0
Ackley	$F_8 = -20 \exp\left(-0. 2 \sqrt{\frac{1}{Dim} \sum_{i=1}^{Dim} x_i^2}\right) - \exp\left(\frac{1}{Dim} \sum_{i=1}^{Dim} \cos(2\pi x_i)\right) + 20 + e$	10/30/100/200	$[-32, 32]$	0
Griewank	$F_9 = \frac{1}{4000} \sum_{i=1}^{Dim} x_i^2 - \prod_{i=1}^{Dim} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	10/30/100/200	$[-600, 600]$	0

表 2 算法参数设置

算法	参数设置
WOA	$N=30, T_{\max}=500, Dim=30$
GSA	$N=30, T_{\max}=500, Dim=30, G_0=100, \alpha=30$
EWOA	$N=30, T_{\max}=500, Dim=30, \gamma=\frac{1}{2}, \lambda=1, F=0.5$

表 3 EWOA 与 WOA、GSA 算法性能测试对比

函数	算法	Mean	Best	Std. Dev
F_1	WOA	4.53E-071	2.10E-088	2.18E-070
	GSA	2.45E-016	1.32E-016	1.29E-016
	EWOA	0.00E+000	0.00E+000	0.00E+000
F_2	WOA	1.72E-050	4.05E-057	6.73E-050
	GSA	8.12E-008	4.92E-008	3.34E-008
	EWOA	6.41E-246	0.00E+000	0.00E+000
F_3	WOA	4.33E+004	1.36E+004	1.27E+004
	GSA	1.12E+003	6.51E+002	5.00E+002
	EWOA	0.00E+000	0.00E+000	0.00E+000
F_4	WOA	4.33E+001	2.88E+000	3.02E+001
	GSA	7.19E+000	2.60E+000	2.43E+000
	EWOA	1.06E-242	0.00E+000	0.00E+000
F_5	WOA	4.40E-003	1.80E-004	5.40E-003
	GSA	8.77E-002	3.08E-002	4.62E-002
	EWOA	6.23E-005	1.57E-006	7.15E-005
F_6	WOA	-1.04E+004	-1.26E+004	1.79E+003
	GSA	-2.59E+003	-3.56E+003	4.24E+003
	EWOA	-1.17E+004	-1.26E+004	1.38E+003
F_7	WOA	0.00E+000	0.00E+000	0.00E+000
	GSA	2.96E+001	1.09E+001	9.17E+000
	EWOA	0.00E+000	0.00E+000	0.00E+000
F_8	WOA	4.32E-015	8.88E-016	2.72E-015
	GSA	1.14E-008	7.43E-009	2.71E-009
	EWOA	8.88E-016	8.88E-016	0.00E+000
F_9	WOA	1.26E-002	0.00E+000	5.23E-002
	GSA	3.06E+001	1.72E+001	5.32E+000
	EWOA	0.00E+000	0.00E+000	0.00E+000

从表 3 可以看出,针对本文选取的 9 个基准测试函数,EWOA 算法的寻优精度均远远高于 GSA 算法.而对于传统 WOA 算法,通过观察发现,对多模态函数 F_7 ,EWOA 算法和传统 WOA 算法均可取到理论最优解,但由图 2(g)可以看出,传统 WOA 算法是当迭代到达 248 次时取得最优解 0,而 EWOA 算法在第 7 次迭代便已取到最优解 0,表明本文提出的改进策略提高了算法的优化性能.

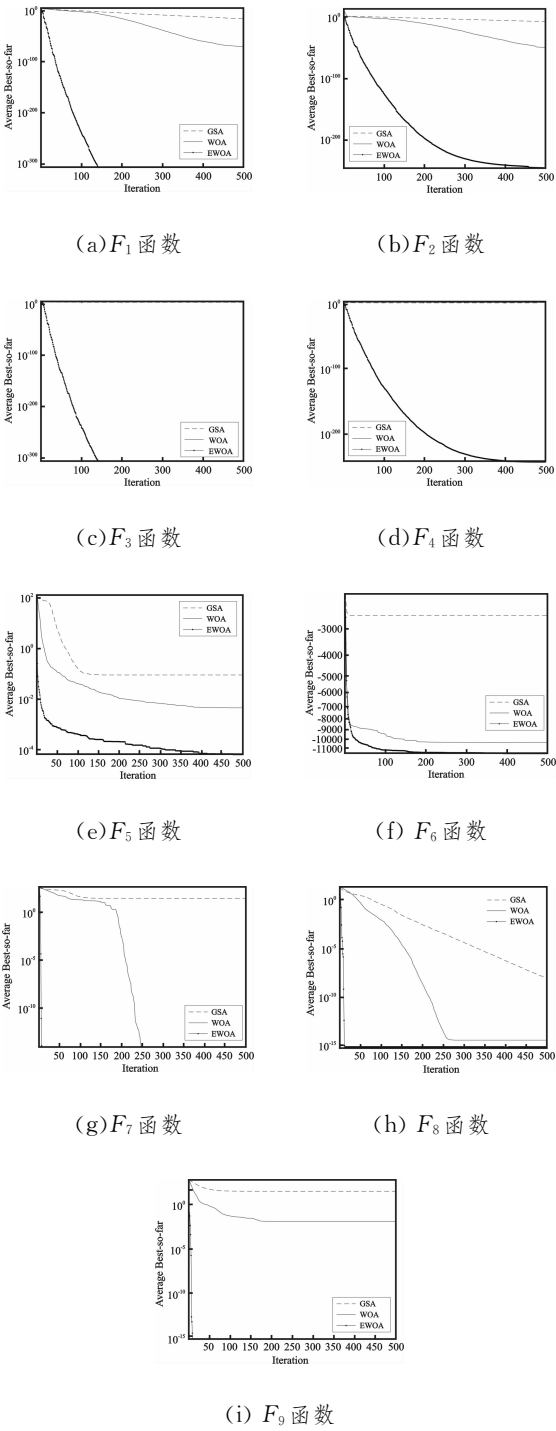


图 2 算法优化基准测试函数的收敛曲线

同时,对于单模态函数 F_1 、 F_3 和多模态函数 F_8 、 F_9 分别在迭代到达 144、144、12、12 次时取到其理论最优解;对于函数 F_2 、 F_4 的寻优精度也具有较大提高;然而,对于函数 F_5 、 F_6 的寻优精度改善不够明显.但通过图 2 观察发现,EWOA 算法的收敛速度明显优于其他算法,表明本文提出的算法能够有效加快算法收敛速度,提高算法寻优精度.

4.2 不同维度下算法性能测试

设置算法其他参数不变的情况下,分别比较 EWOA 算法和传统 WOA 算法在 10 维、100 维和 200 维下的寻优精度,实验看出,对于测试维度分别为 100 和 200 时,传统 WOA 算法对函数 F_9 能取到其理论最优解,且对函数 F_8 和 F_9 的最小值均可取到其理论最优解,但对于其他函数,EWOA 算法的优化结果明显优于传统 WOA 算法,表明 EWOA 算法具有较好的高维函数优化性能;对于 Dim=10 而言,EWOA 算法对本文选取的 9 个基准测试函数的优化结果均远远优于传统 WOA 算法,且函数 F_1 、 F_3 、 F_7 、 F_8 、 F_9 均取得其理论最优值。

4.3 与参考文献的算法性能对比

为了更好地验证 EWOA 算法的优化性能,将其与参考文献[5]和[6]中的算法进行比较,实验结果如表 4 所示。

从表 4 可以看出,EWOA 算法与传统 WOA 算

法相比,EWOA 算法明显具有更优的性能. 对于函数 F_7 ,所有的算法均可以使其取得理论最优解,且 4.1 节的实验结果表明了 EWOA 算法的改进策略对于函数 F_7 的优化行之有效;对于函数 F_9 ,EWOA 算法、IWOA 算法和 WOAWC 算法都能够使得其取得最优解,但对于其他函数的优化结果,EWOA 算法明显优于 IWOA 算法。

对比本文算法和文献[6]的算法 WOAWC,两种算法在对函数 F_1 、 F_3 、 F_7 、 F_8 、 F_9 均取到理论最优解;但当 WOAWC 算法优化函数 F_1 、 F_7 、 F_8 使得其取得最优解时,到达的迭代进化次数分别约为 460、130、225 次,而本文算法 EWOA 算法仅到 144、7、12 次时便取得其最优解,表明 EWOA 算法的收敛速度具有显著提高. 同时,对于其他函数的优化,EWOA 算法都表现出比 WOAWC 算法更优的效果,表明了 EWOA 算法具有更优的寻优能力。

表 4 EWOA 算法与参考文献中算法性能对比

函数	WOA		IWOA ^[5]		WOAWC ^[6]		EWOA	
	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev
F_1	4.53E-071	2.18E-070	6.54E-125	6.80E-125	0.00E+000	0.00E+000	0.00E+000	0.00E+000
F_2	1.72E-050	6.73E-050	2.15E-073	3.64E-073	3.32E-178	0.00E+000	6.41E-246	0.00E+000
F_3	4.33E+004	1.2726e+04	1.56E-023	1.81E-023	0.00E+000	0.00E+000	0.00E+000	0.00E+000
F_4	4.33E+001	3.02E+001	7.06E-007	2.18E-006	1.19E-180	0.00E+000	1.06E-242	0.00E+000
F_5	4.40E-003	5.40E-003	2.42E-004	4.41E-004	1.30E-004	1.00E-004	6.23E-005	7.15E-005
F_6	-1.04E+004	1.79E+003	-1.14E+004	1.13E+002	-1.06E+004	2.25E+003	-1.17E+004	1.38E+003
F_7	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000
F_8	4.32E-015	2.72E-015	3.02E-015	1.95E-015	8.88E-016	0.00E+000	8.88E-016	0.00E+000
F_9	1.26E-002	5.23E-002	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000	0.00E+000

5 结束语

针对传统鲸鱼优化算法寻优精度低、易陷入局部最优等问题,提出了一种改进的鲸鱼优化算法(EWOA). 首先,本文在寻优过程中将自适应策略引入鲸鱼位置更新公式中,以便使得较好的猎物位置随迭代进化不断增加,同时使包围步长随迭代进化不断减小,以便平衡算法的全局探索和局部开发能力、加快算法收敛速度、提高算法的寻优精度;然后,引入差分变异思想,将较优的鲸鱼位置进行变异后,并通过贪婪选择判断变异后的优劣性,使得算法能够有效跳出局部最优,避免算法出现早熟收敛现象;最后,通过对 9 个基准测试函数在固定参数和不同维度条件下进行仿真实验,同时与参考文献中的

算法进行对比,所有实验结果表明,本文算法具有较好的有效性和优越性.

参考文献:

[1] 钟明辉, 龙文. 一种随机调整控制参数的鲸鱼优化算法[J]. 科学技术与工程, 2017, 17(12):68-73.
[2] KAUR G, ARORA S. Chaotic Whale Optimization Algorithm[J]. Journal of Computational Design & Engineering, 2018, 5(3):275-284.
[3] 闫旭, 叶春明, 姚远远. 量子鲸鱼优化算法求解作业车间调度问题[J/OL]. 计算机应用研究, 2019, 36(4):1-2[2018-02-07]. <http://kns.cnki.net/kcms/detail/51.1196.TP.20180209.1230.160.html>.

参考文献:

[1] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69(3):46-61.

[2] 龙文,赵东泉,徐松金. 求解约束优化问题的改进灰狼优化算法[J]. 计算机应用, 2015, 35(9):2590-2595.

[3] 马卫,孙正兴,李俊楼. 基于 Powell 局部搜索策略的全局优化布谷鸟算法[J]. 计算机应用研究, 2015, 32(6):1667-1675.

[4] 张新明,涂强,康强,程金凤. 灰狼优化与差分进化的混合算法及函数优化[J]. 计算机科学, 2017, 44(9): 93-98.

[5] 张贾奎,崔利杰,郭庆,陈浩然. 基于 Tent 混沌序列的灰狼优化算法[J]. 微电子学与计算机, 2018, 35(6): 11-16.

[6] 张悦,孙惠香,魏政磊,韩博. 具有自适应调整策略的混沌灰狼优化算法 [J]. 计 算 机 科学, 2017, 44(S2):

119-122.

[7] SAREMI S, MIRJALILI S Z, MIRJALILI S M, et al. Evolutionary population dynamics and grey wolf optimizer[J]. Neural Computing & Applications, 2015, 26(5):1257-1263.

作者简介:

莫艳红 女, (1994-), 硕士研究生. 研究方向为最优化方法研究. E-mail: 577463887@qq. com.

聂 慧 女, (1972-), 博士, 副研究员. 研究方向为人力资源管理.

刘振丙 男, (1980-), 博士, 教授. 研究方向为图像处理、机器学习.

杨辉华 男, (1972-), 博士, 教授. 研究方向为机器学习、大数据分析.

(上接第 77 页)

[4] 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.

[5] 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.

[6] 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.

[7] WATKINS W A, SCHEVILL W E. Aerial Observation of Feeding Behavior in Four Baleen Whales: Eubalaena glacialis, Balaenoptera borealis, Megaptera novaeangliae, and Balaenoptera physalus[J]. Journal of Mammalogy, 1979, 60(1):155-163.

[8] 陈占伟, 李骞. 一种自适应惯性权重的粒子群优化算法[J]. 微电子学与计算机, 2011, 28(3): 27-30.

[9] STRON R, PRICE K. Differential evolution — A simple and efficient heuristic for global optimization over continuous spaces[J]. J of Global Optimization, 1997, 11(4): 341-359.

[10] NERI F, TIRRONEN V. Recent advances in differential evolution: a survey and experimental analysis[J]. Artificial Intelligence Review, 2010, 33(1-2):61-106.

作者简介:

魏康园 男, (1991-), 硕士研究生. 研究方向为数据挖掘、进化计算.

何 庆(通讯作者) 男, (1982-), 博士, 副教授. 研究方向为大数据应用、人工智能. E-mail: qhe@gzu. edu. cn.

徐钦帅 男, (1994-), 硕士研究生. 研究方向为机器学习、进化计算.