

引用格式:莫艳红,聂慧,刘振丙,等. 基于莱维飞行的灰狼优化算法 [J]. 微电子学与计算机,2019,36(4):78-83. [MO Y H, NIE H, LIU Z B, et al. Grey wolf optimization algorithm based on levy flight [J]. Microelectronics & Computer, 2019, 36(4): 78-83.]

基于莱维飞行的灰狼优化算法

莫艳红¹, 聂慧¹, 刘振丙¹, 杨辉华²

(1 桂林电子科技大学, 广西 桂林 541004; 2 北京邮电大学, 北京 100876)

摘 要: 针对灰狼优化算法在求解比较复杂的非线性优化问题时容易早熟收敛, 陷入局部极值以及进化后期精度低的情况, 在原始的灰狼优化算法(GWO)引入了一种属于随机游动的莱维飞行, 提出了一种基于莱维飞行的灰狼优化算法(LGWO). 在头狼位置进行更新时, 利用莱维飞行对头狼进行全局搜索, 防止狼群丧失多样性, 易陷入局部最优解. 通过仿真, 与经典的布谷鸟搜索算法和粒子群算法等群智能算法进行对比, 结果表明基于莱维飞行的灰狼优化算法能够有效地提高解的精度并加快收敛速度, 寻优效果更优.

关键词: 灰狼优化算法; 莱维飞行; 群智能算法

中图分类号: TP301

文献标识码: A

文章编号: 1000-7180(2019)04-0078-06

Grey Wolf Optimization Algorithm Based on Levy Flight

MO Yan-hong¹, NIE Hui¹, LIU Zhen-bing¹, YANG Hui-hua²

(1 Guilin University of Electronic Technology, Guilin 541004, China;

2 Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: The grey wolf optimization algorithm (GWO) has some demerits on solving more complex nonlinear optimization problems, such as relapsing into local extremum, slow convergence velocity and low convergence precision in the late evolutionary. this paper proposed a kind GWO algorithm based on levy flight (LGWO) via introducing of levy flight that is random walk. When updating the wolf's position, levy flight was used to conduct a global search on the wolf to prevent the wolves from losing diversity and easily falling into a local optimal solution. Through simulation, compared with the classic cuckoo search algorithm, particle swarm optimization algorithm and other group intelligence algorithms, the results show that the grey wolf optimization algorithm based on levy flight can effectively improve the accuracy of the solution and convergence speed, and the optimization effect is better.

Key words: grey wolf optimization algorithm; levy flight; group intelligence algorithms

1 引言

灰狼优化算法(GWO)是由 Mirjalili 等人^[1]通过模拟狼群觅食行为提出的仿真智能优化算法. 由于 GWO 算法原理简单、参数少、易于编程实现、并具有支持分布式并行计算和较强的全局搜索能力等特点, 被广泛应用于计算机科学、工程科学、管理科学等领域的全局优化问题.

近年来, 很多学者为了提高 GWO 算法的性能, 对它做了很多改进. 龙文等人^[2]在灰狼优化算法中引入 Powell 局部搜索^[3], 有效提高了算法局部搜索能力. 张新明等人^[4]在灰狼优化算法中引入自适应调节参数的差分进化策略, 提高了算法的性能. 张贾奎等人^[5]在灰狼优化算法中引入 Tent 混沌序列的局部搜索策略, 提高了算法的局部搜索能力. 张悦等人^[6]在灰狼个体的适应度中引入了自适应调整策略

收稿日期: 2018-07-04; 修回日期: 2018-09-13

基金项目: 国家自然科学基金(21365008, 61105004)

和位置更新策略,提升了算法运行速度. Saremi 等人^[7]提出了一种混合 GWO 算法,该算法能够有效的提高种群的质量. 从以上的研究可以发现,对于灰狼优化算法的改进通常是增加各种异动、扰动等操作,这些方法的确能够在一定程度上提升

算法的收敛精度和速度,但是难以克服 GWO 算法易陷入局部最优的固有弱点. 因此,本文提出了基于莱维飞行的灰狼优化算法(LGWO),增加种群的多样性,减少陷入局部极值的次数,有效平衡算法全局与局部搜索能力.

2 基本灰狼优化算法

2.1 灰狼群体捕食行为

灰狼是群居动物,通常每个群体有十几只灰狼,构建了严格的灰狼金字塔等级制度,如图 1 所示. 金字塔第一层为种群中的头狼称为 α ,主要负责群体各项决策事务;金字塔第二层称为 β ,协助 α 做出管理决策;金字塔第三层为 δ , δ 听从 α 及 β 的指令;金字塔的最底层是 ω ,主要用于协助捕食.

灰狼群体依靠这种等级制度进行觅食活动. 捕食开始时, α 、 β 、 δ 狼最靠近猎物, ω 狼群跟随这三者搜索、追踪、包围猎物,当包围圈足够小时,开始进攻袭击并捕获猎物.

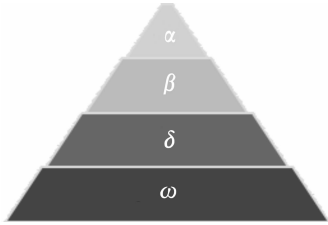


图 1 灰狼种群等级金字塔示意图

2.2 GWO 算法描述

利用 GWO 算法求解函数优化问题时,假设灰狼种群规模为 N ,空间搜索维度为 d 维, $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ 用来表示第 i 只灰狼在 d 维空间的位置,则群体内当前最优解为 α 狼的位置;次优解为 β 狼的位置,第三优解为 δ 狼的位置,其余的为 ω 狼的位置.

(1) 搜索猎物

灰狼群体的围捕行为是从搜索猎物开始的,它们对猎物的搜索行为可以通过如下公式进行描述:

$$D = |C \cdot X_p(t) - X(t)| \quad (1)$$

$$X(t+1) = X_p(t) - A \cdot D \quad (2)$$

式中, $X_p(t)$ 表示第 t 代时猎物的位置; $X(t)$ 表示第

t 代时灰狼个体的位置;常数 C 为摆动因子,由式(3)决定, A 为收敛因子,由式(4)决定:

$$C = 2r_1 \quad (3)$$

$$A = 2ar_2 - a \quad (4)$$

式中, r_1 和 r_2 是取值在 $[0,1]$ 之间的随机向量, a 在迭代过程中从 2 线性地减小到 0.

(2) 包围猎物

在 α 狼的带领下, β 狼和 δ 狼逐渐靠近猎物,可以依靠这三者来估计猎物的位置,尽量靠近这三只灰狼,也就尽可能离目标值最近. 对于每一只 ω 狼,首先根据式(5)~(10)计算它们与 α 、 β 、 δ 的距离,然后由式(11)综合判断出灰狼个体向猎物移动的方向.

$$D_\alpha = |C_1 \cdot X_\alpha(t) - X(t)| \quad (5)$$

$$D_\beta = |C_2 \cdot X_\beta(t) - X(t)| \quad (6)$$

$$D_\delta = |C_3 \cdot X_\delta(t) - X(t)| \quad (7)$$

$$X_1 = X_\alpha - A_1 \cdot D_\alpha \quad (8)$$

$$X_2 = X_\beta - A_2 \cdot D_\beta \quad (9)$$

$$X_3 = X_\delta - A_3 \cdot D_\delta \quad (10)$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \quad (11)$$

式中, X_α 表示 α 狼的位置, X_β 表示 β 狼的位置, X_δ 表示 δ 狼的位置, C_1, C_2, C_3 是随机向量.

(3) 袭击猎物

当猎物停止移动时,灰狼群体袭击猎物,至此灰狼群体的捕猎行为结束. 这个过程可以通过收敛因子 A 在迭代过程中从 2 线性地减少到 0 来模拟. 当 $|A| \leq 1$ 时,灰狼开始袭击猎物. 灰狼的袭击行为模拟使灰狼算法具有良好的局部精细搜索能力.

3 基于莱维飞行的灰狼优化算法

3.1 LGWO 算法原理

在标准的灰狼优化算法(GWO)中, α 狼的位置代表最优解,由于精英保存策略,在 GWO 算法后期,群体中所有的灰狼个体均向最优个体 α 狼靠近,导致种群丧失多样性,易陷入局部收敛,出现早熟收敛现象. 针对这些弊端,本文利用 Levy 飞行对群体中的灰狼个体 α 狼进行全局搜索. Levy 飞行属于随机游动,是一种很好的搜索策略,能够扩大搜索范围,其飞行步长满足一个重尾的稳定分布. 通过 Levy 飞行,新一代的 α 狼的计算公式如下:

$$X_\alpha(t+1) = X_\alpha(t) - a \oplus \text{Levy}(\beta) \quad (12)$$

式中, $X(t)$ 表示第 t 代时 α 狼个体的位置; $\oplus$ 为点对点乘法; a 表示 α 狼个体位置的随机数,由式(13)决

定;Levy(β) 代表随机搜索路径,由式(14) 决定.

$$a = \text{random}(\text{size}(\alpha_position)) \quad (13)$$

$$\text{Levy}(\beta) \sim 0.01 \frac{u}{|v|^{1/\beta}} (X_\alpha(t) - X_{\alpha\text{best}}) \quad (14)$$

式中, β 的取值区间一般为 $1 < \beta < 3$,在本文中, $\beta = 1.5$; $X_{\alpha\text{best}}$ 表示历史最优 α 狼的位置; u 和 v 服从式(15) 所示的正态分布:

$$u \sim N(0, \sigma_u^2) \quad v \sim N(0, \sigma_v^2) \quad (15)$$

σ_u 和 σ_v 取值如下:

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta) \sin(\frac{\pi\beta}{2})}{\Gamma[\frac{1+\beta}{2}] \beta 2^{(\beta-1)/2}} \right\}^{1/\beta}, \sigma_v = 1 \quad (16)$$

3.2 LGWO 算法流程

算法流程如下:

Step1 数值初始化. 初始化灰狼个体的位置和目标函数值.

Step2 选出头狼. 选取最优 α 狼个体为头狼.

Step3 更新头狼位置. 根据式(1)和式(2)更新 α 狼位置,同时利用式(12)~(16)对 α 狼进行全局搜索. 计算对每一次迭代计算相应的 a 、 A 、 C 等参数的值.

Step4 包围猎物. 按照式(5)~(11)对参与围攻行为的 α 狼、 β 狼、 δ 狼位置进行更新,执行对猎物的围攻. 当 $|A| < 1$ 时,进行下一步,袭击猎物.

Step5 若达到迭代终止条件则输出头狼的位置,否则转 Step2.

4 实验仿真与分析

4.1 测试函数

为了检测本文提出的 IGWO 算法的寻优性能,选取了 9 个标准测试函数进行仿真实验,测试函数的表达式如表 1 所示. 其中 $f_1 \sim f_3$ 为单峰函数, $f_4 \sim f_6$ 为多峰函数, $f_7 \sim f_9$ 为固定维度多峰函数.

4.2 与标准 GWO 和 LGWO 算法的比较

利用 LGWO 算法对 9 个标准测试函数进行求解,并与标准的 GWO 算法进行比较. 2 种算法参数设置如下:种群规模 $N=30$,总迭代数 $T=1000$. 基于上述的参数设置,采用 LGWO 算法和 GWO 算法分别独立运行 50 次后取最优值、最差值、平均值和标准差作为最终的测试结果,测试结果如表 2 所示,寻优曲线图如图 2 所示. 本文所提到的算法的仿真实验在 MATLAB R2013a、CPU Intel Core2@2.0 GHz、RAM 2 GB、Window10 环境下进行.

由表 2 的所有 9 个测试函数结果来看, LGWO

算法在整体性能上是明显超越了 GWO 算法的. 单峰函数测试结果表明 LGWO 算法无论是在最优值、最差值、平均值或均方差上,均获得了较好的解,表明 LGWO 算法对于单峰函数有着更高的优化精度. 从 $f_4 \sim f_6$ 可以看出 LGWO 算法获得了较好的寻优结果,特别是测试函数 f_5 ,两种算法均得到了相似的最优值 0,而 LGWO 算法获得了较好的平均值、最差值和标准差. $f_7 \sim f_9$ 测试函数的结果显示, GWO 算法在固定维度多峰函数的最优化求解方面存在较大的困难,容易陷入局部最优.

图 2 记录了 LGWO 算法和 GWO 算法在每一个测试函数的收敛结果,从这些图中可以看出, LGWO 算法明显高于 GWO 算法的收敛精度. 所以对于 GWO 算法所存在的后期容易陷入局部最优的问题, LGWO 算法已经有了很好的改进,不仅如此, LGWO 算法的收敛速度在 GWO 算法的基础上也得到了显著的提高. 由上述比较得知,与标准 GWO 算法相比,不论是收敛精度,还是收敛稳定性方面, LGWO 算法的寻优性能有了明显提高.

4.3 与其他算法的比较

为了进一步验证 LGWO 算法的有效性,将其与 CS 算法和 PSO 算法的结果进行比较. 实验过程中,各算法参数设置详情为:CS 算法中,种群规模 $N=30$,总迭代数 $T=1000$,鸟巢被发现的概率取值 $P_a=0.25$; PSO 算法中,惯性权重 $w_{\max}=0.9$, $w_{\min}=0.2$, $c_1=c_2=2.0$. 测试结果如表 3 所示.

从表 3 可知,与 PSO 算法相比, LGWO 算法在单峰函数和多峰函数上获得了较好的结果,在 f_7 测试函数上得到了相似的结果,在 f_8 测试函数上得到了较差的寻优结果,在 f_9 测试函数上, PSO 表现出较差的优化性. 与 CS 算法相比, LGWO 算法的解的精度较高,特别是 f_5 和 f_9 测试函数的求解中, LGWO 算法体现了其改进的优势. 由上述可以看出, LGWO 算法较 PSO 算法和 CS 算法具有更好的测试效果,这说明通过引入莱维飞增加算法的跳跃灵活性、提高算法性能是有效的.

5 结束语

本文在基本灰狼优化算法上引入具有跳跃特性的 Levy 飞行,使算法能够在搜索后期摆脱局部收敛,并且在收敛速度快的同时能更好的提高收敛精度,达到了改进的目的. 对 9 个基本函数进行了仿真测试,仿真结果表明 LGWO 算法能够有效的跳出局部最优值,找到更好的解,且结果鲁棒性

更强. 如何将改进的灰狼优化算法同其他智能算法相结合来解决生活中的实际问题,将是下一步研究的重点工作.

表 1 标准测试函数

函数	维度	范围	最小值
$f_1(x)=\sum_{i=1}^n x_i^2 +\prod_{i=1}^n x_i $	30	$[-10,10]$	0
$f_2(x)=\sum_{i=1}^n(\sum_{j=1}^nx_j)^2$	30	$[-100,100]$	0
$f_2(x)=\sum_{i=1}^nix_i^4+random(0,1)$	30	$[-1.28,1.28]$	0
$f_4(x)=\sum_{i=1}^n-x_i\sin(\sqrt{ x_i })$	30	$[-500,500]$	0
$f_5(x)=\frac{1}{4000}\sum_{i=1}^nx_i^2-\prod_{i=1}^2\cos(\frac{x_i}{\sqrt{i}})+1$	30	$[-600,600]$	0
$f_6(x)=0.1\{\sin^2(3\pi x_1)+\sum_{i=1}^n(x_i-1)^2[1+\sin^2(3\pi x_i+1)]+(x_n-1)^2[1+\sin^2(2\pi x_n)]\}+\sum_{i=1}^nu(x_i,5,100,4)$	30	$[-50,50]$	0
$f_7(x)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi}\cos x_1)+10$	2	$[-5,5]$	0.398
$f_8(x)=\sum_{i=1}^nc_i\exp(-\sum_{j=1}^3a_{ij}(x_j-p_{ij})^2)$	3	$[-5,5]$	3.86
$f_9(x)=\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	$[0,10]$	-10.5363

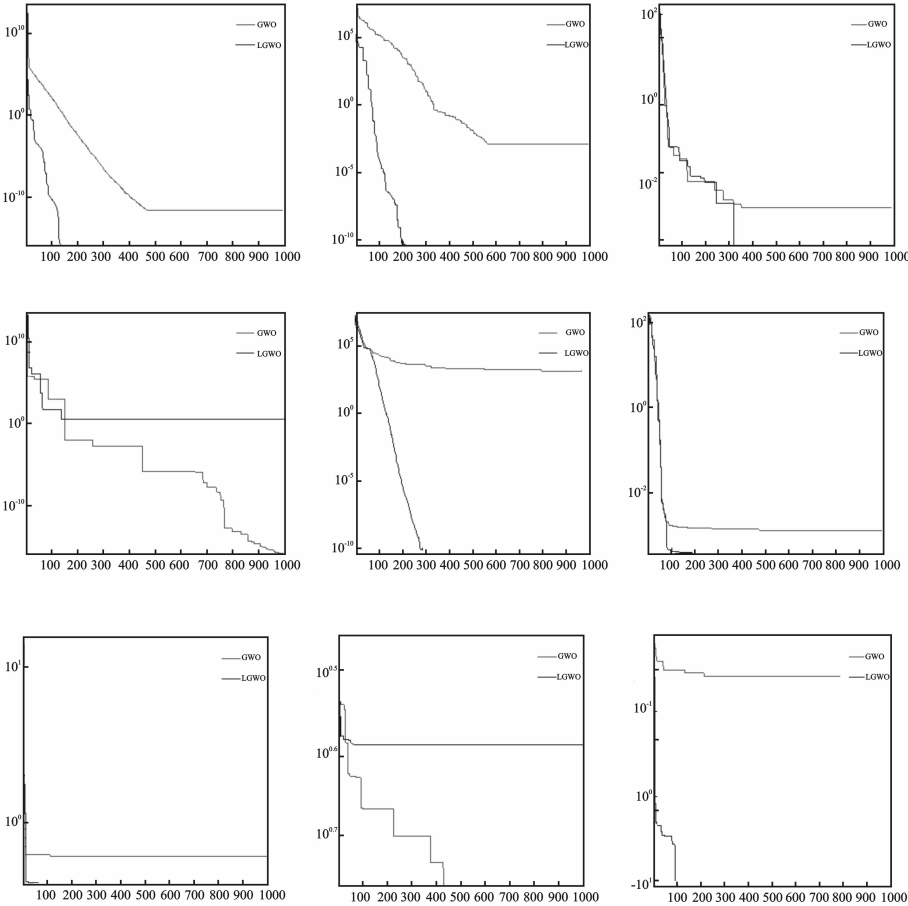


图 2 GWO 和 LGWO 算法寻优曲线对照图

表 2 GWO 和 LGWO 算法测试结果对比

F	目标值	极值类型	算法	Δ_{best}	Δ_{worst}	Δ_{avg}	Δ_{std}
f_1	0	极小值	LGWO	1.820 7e-17	5.761 5e-16	1.516e-16	6.683e-16
			GWO	1.507e-13	7.237e-11	2.381 2-12	6.163 7e-11
f_2	0	极小值	LGWO	1.159 6e-10	7.550 3e-09	5.103 7e-10	1.790 7e-09
			GWO	3.442 1e-04	0.1480	0.017 9	0.010 3
f_3	0	极小值	LGWO	3.007 1e-04	7.822 5e-04	4.948 5e-04	1.483 2e-03
			GWO	0.001 9	0.012 2	0.007 1	0.001 0
f_4	-418.9829*5	极小值	LGWO	-455.316 6*5	-630.542 1*5	556.000 2*5	17.8175
			GWO	-719.998 0*5	-1587.232 4*5	878.561 4*5	53.597 3
f_5	0	极小值	LGWO	0	0	0	0
			GWO	0	0.056 7	0.002 6	3.051 6e-03
f_6	0	极小值	LGWO	0.101 8	0.174 7	0.142 6	0.000 8
			GWO	1.300 3	1.854 8	1.525 7	0.051 3
f_7	0.398	极小值	LGWO	0.398 5	0.409 9	0.405 7	0.003 6
			GWO	0.600 1	0.618 4	0.609 6	0.001 9
f_8	-3.86	极小值	LGWO	-3.862 2	-3.832 8	-3.849 1	0.003 2
			GWO	-5.782 4	-5.793 4	-5.787 8	0.002 6
f_9	-10.5363	极小值	LGWO	-10.536 3	-10.536 3	-10.536 3	0
			GWO	-0.773 0	-0.045 8	-0.415 9	0.109 6

表 3 其他算法和 LGWO 算法测试结果对比

F	目标值	极值类型	算法	Δ_{best}	Δ_{worst}	Δ_{avg}	Δ_{std}
f_1	0	极小值	LGWO	1.8207e-17	5.7615e-16	1.516e-16	6.683e-16
			PSO	1.3757e-09	6.1273e-09	3.8275e-09	7.6641e-09
			CS	1.4230e-04	3.9176e-03	5.9343e-04	2.4256e-03
f_2	0	极小值	LGWO	1.1596e-10	7.5503e-09	5.1037e-10	1.7907e-09
			PSO	0.007 3	0.026 1	0.016 3	0.004 7
			CS	1.4654e-04	0.002 4	0.001 1	0.000 3
f_3	0	极小值	LGWO	3.0071e-04	7.8225e-04	4.9485e-04	1.4832e-03
			PSO	0.125 5	0.224 9	0.168 4	0.021 4
			CS	20.606 5	28.554 4	24.800 3	2.578 7
f_4	-418.982 9*5	极小值	LGWO	-455.3166*5	-630.542 1*5	-556.000 2*5	17.817 5
			PSO	-600.655 8*5	-890.363 5*5	-747.518 1*5	27.2183
			CS	-225.106 6*5	-485.55 42	-334.017 6*5	19.513 6
f_5	0	极小值	LGWO	0	0	0	0
			PSO	2.6914e-10	2.3159e-06	1.2276e-08	5.4184e-06
			CS	0.052 0	0.109 1	0.073 9	0.004 3
f_6	0	极小值	LGWO	0.101 8	0.174 7	0.142 6	0.000 8
			PSO	0.164 6	44.662 4	28.320 5	6.1542
			CS	0.169 2	0.261 1	0.190 5	0.001 3
f_7	0.398	极小值	LGWO	0.398 5	0.409 9	0.405 7	0.003 6
			PSO	0.397 8	0.397 8	0.397 8	0
			CS	6.009 5	72.316 1	39.640 8	7.603 4
f_8	-3.86	极小值	LGWO	-3.862 2	-3.832 8	-3.849 1	0.003 2
			PSO	-3.862 8	-3.862 7	-3.862 8	4.8559e-05
			CS	-2.224 6	-0.060 8	-1.012 1	0.536 2
f_9	-10.536 3	极小值	LGWO	-10.536 3	-10.536 3	-10.536 3	0
			PSO	-9.574 8	-18.421 7	-11.754 7	0.301 0
			CS	-0.697 72	-0.149 62	-0.390 7	0.059 5

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.