第 56 卷　第 1 期
2020 年 1 月

南京大学学报(自然科学)
JOURNAL OF NANJING UNIVERSITY
(NATURAL SCIENCE)

Vol. 56, No. 1
Jan., 2020

DOI:10.13232/j.cnki.jnju.2020.01.005

基于改进蝗虫优化算法的特征选择方法

刘　亮[1,2],何　庆[1,2*]

(1.贵州大学大数据与信息工程学院,贵阳,550025;2.贵州省公共大数据重点实验室,贵州大学,贵阳,550025)

摘　要:针对传统蝗虫优化算法寻优精度低和收敛速度慢的问题,提出一种基于非线性调整策略的改进蝗虫优化算法.首先,利用非线性参数代替传统蝗虫算法中的递减系数,协调算法全局探索和局部开发能力,加快算法收敛速度;其次,引入自适应权重系数改变蝗虫位置更新方式,提高算法寻优精度;然后,结合limit阈值思想,利用非线性参数对种群中部分个体进行扰动,避免算法陷入局部最优.通过六个基准测试函数的仿真结果表明,改进算法的收敛速度和寻优精度均有明显提高.最后将改进算法应用于特征选择问题中,通过在七个数据集上的实验结果表明,基于改进算法的特征选择方法能够有效地进行特征选择,提高分类准确率.

关键词:蝗虫优化算法,非线性参数,自适应权重,limit阈值,特征选择

中图分类号:TP301.6　　**文献标识码:**A

An feature selection method based on improved grasshopper optimization algorithm

Liu Liang[1,2], He Qing[1,2*]

(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: Focused on the issue of low search precision and slow convergence speed of traditional grasshopper optimization algorithm, an improved grasshopper optimization algorithm based on non-linear adjustment was proposed. Firstly, non-linear parameters were used to replace the decline coefficient of traditional grasshopper optimization algorithm, which coordinated the exploration and exploitation ability, and improved the convergence speed. Secondly, the adaptive weight coefficient was introduced to change the grasshopper position renewal modes to improve the search precision. Then, in order to avoid premature convergence, the algorithm combined limit threshold idea and used non-linear parameters to disturb some individuals in the population. The simulation results on six benchmark functions show that the improved algorithm has significant improvement in convergence speed and search precision. Finally, the improved algorithm was applied to the feature selection problem. The experimental results on seven datasets show that the feature selection method based on the improved algorithm can effectively select features and improve the classification accuracy.

Key words: grasshopper optimization algorithm, non-linear parameter, adaptive weight, limit threshold, feature selection

基金项目:贵州省科技计划重大专项(黔科合重大专项字[2018]3002,黔科合重大专项字[2016]3022),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)

收稿日期:2019－08－20

*通讯联系人,E-mail:qhe@gzu.edu.cn

特征选择作为机器学习中数据预处理的关键环节,不仅能够降低数据维度、提高算法的学习效率[1],还能从数据集中筛选出对分类器分类性能最有用的特征[2],提高分类准确率. 常见的特征选择方法大致可分为过滤式、包裹式以及嵌入式三种[3],其中包裹式特征选择将学习器的性能优劣作为特征子集的评价标准[4],因此该方式也最有利于为学习器选择最佳特征子集. 但当数据中包含大量特征时,采用包裹式方法对特征子集进行穷举搜索是很难实现的,因此,如何进行有效的特征选择成了一个难题. 近年来,许多学者使用群智能优化算法的搜索方式作为包裹式特征选择的搜索机制,包括粒子群优化算法[5](Particle Swarm optimization,PSO)、蚁狮优化算法[6](Ant Lion Optimizer, ALO)以及鲸鱼优化算法[7](Whale Optimization Algorithm,WOA)等较为经典的群智能算. 蝗虫优化算法(Grasshopper Optimisation Algorithm, GOA)是 Saremi et al[8]于2017年提出的一种模拟自然界中蝗虫的群体行为来解决优化问题的新型群智能优化算法,实验证明该算法在实际问题的求解中有显著效果,但与其他群智能优化算法类似,GOA存在收敛速度慢、易陷入局部最优的问题. 为提高算法收敛速度和寻优精度,许多学者利用不同的方法对GOA进行改进. Ewees et al[9]引入对立学习的思想来更新GOA中每次迭代后的种群,提出OBLGOA(Opposition-based Learning Grasshopper Optimization Algorithm)算法,并通过实验证明该算法的性能比GOA有明显提升. Luo et al[10]利用高斯变异增加种群多样性,并且引入Lévy飞行策略增加搜索的随机性,提高算法的全局寻优能力,实验结果显示,该方法能显著提升算法的收敛速度和寻优精度. Arora and Anand[11]将混沌理论引入GOA中,利用混沌映射来平衡算法的全局探索和局部开发能力,并通过基准函数对算法进行测试,证明该方法的有效性和优越性. 李洋州和顾磊[12]提出利用曲线自适应代替GOA中的关键参数,并引入模拟退火机制,实验证明,和传统GOA算法相比,该方法寻优精度更高,收敛速度更快. 可见,为解决GOA的问题目前已有许多研究成果,但如何以更有效更简便的方式进一步提高GOA的寻优精度和收敛速度依然值得深入研究.

本文针对GOA寻优精度低、收敛速度慢的问题,采用非线性参数代替GOA中的下降系数,能更好地平衡算法在迭代过程中的全局探索和局部开发能力,提高算法的收敛速度;通过引入自适应权重系数改变蝗虫位置的更新方式,提高算法的寻优精度;为避免算法陷入局部最优,结合limit阈值思想,对种群中的个体进行随机扰动,提高种群多样性,增强算法的全局寻优能力. 通过六个基准测试函数的仿真实验证明本文采用的改进策略能有效地提高GOA的寻优精度和收敛速度. 最后,将改进算法应用于特征选择问题,提出一种基于改进GOA算法的包裹式特征选择方法,通过在七个UCI数据集上的实验结果表明,该方法能有效地选取最佳特征,提高分类准确率.

1 蝗虫优化算法

蝗虫常以大规模聚集的方式进行捕食和迁徙,幼虫时期的蝗虫移动范围小且速度缓慢,成年蝗虫则能在大范围空间内进行快速移动. GOA算法即是通过模拟蝗虫的群体特点而衍生出的一种新型群智能优化算法,其数学模型如式(1)所示:

$$X_i = S_i + G_i + A_i \tag{1}$$

其中,X_i表示蝗虫群体中第i只蝗虫的位置,S_i为第i只蝗虫与种群中其他个体之间的相互影响力,G_i为第i只蝗虫所受重力,A_i为第i只蝗虫所受风力. 考虑随机因素的影响可将式(1)改写为:

$$X_i = r_1 \cdot S_i + r_2 \cdot G_i + r_3 \cdot A_i \tag{2}$$

其中,r_1, r_2, r_3为[0,1]之间的随机数,S_i的表达式如下:

$$S_i = \sum_{j=1, j\neq i}^{N} s(d_{ij})\hat{d}_{ij} \tag{3}$$

其中,d_{ij}为第i个个体与第j个个体之间的距离,$\hat{d}_{ij}$为第i个个体到第j个个体的单位向量,且$\hat{d}_{ij} = \frac{x_j - x_i}{d_{ij}}$,$N$为种群中的个体数,$s$为定义个体间相互作用力的函数:

$$s(r) = fe^{\frac{-r}{l}} - e^{-r} \tag{4}$$

其中，f表示吸引力强度，l为吸引力尺度范围．本文取$f=0.5$，$l=1.5$，通过s可将蝗虫个体所在空间划分为吸引区、排斥区与舒适区．但当个体之间距离大于10后，函数s的值接近0，此时不再对该个体产生作用力，因此本文限制种群中个体的位置均在[1，4]范围之内．当种群中的个体都处于舒适区时，个体不再进行位置更新，此时种群中个体围绕在最优解周围，而非全部聚集于最优解所在的位置，因此式(2)的模型不能直接用于求解优化问题，可改写为：

$$X_i^d=c\cdot\left(\sum_{j=1,j\neq1}^{N}c\cdot\frac{ub_d-lb_d}{2}\cdot s\left(\left|x_j^d-x_i^d\right|\right)\cdot\frac{x_j-x_i}{d_{ij}}\right)+\hat{T}_d \tag{5}$$

其中，ub_d与lb_d分别为D维搜索空间的上下界，$\hat{T}_d$为当前种群中的最优个体，不考虑重力影响且假设风向总是指向最优解所在位置，c为线性递减系数，其表达式如下：

$$c=c_{\max}-l\cdot\frac{c_{\max}-c_{\min}}{L} \tag{6}$$

其中，l表示算法当前迭代次数，L表示最大迭代次数，本文取$c_{\max}=1$，$c_{\min}=0.00001$.

基于上述模型，蝗虫优化算法主要步骤如下：

(1)种群及参数初始化．

(2)选择当前种群中适应度最高的最优解．

(3)根据式(6)更新参数c.

(4)调整种群个体间距离，根据式(5)进行位置更新．

(5)检查更新后个体是否超出搜索边界，若超出则返回更新前的位置．

(6)更新种群中的最优解．

(7)判断是否到达最大迭代次数：否，则循环步骤(3)至步骤(6)；是，则算法结束返回最优解．

2　基于非线性调整策略的改进蝗虫优化算法

为提高蝗虫优化算法的寻优精度及收敛速度，本文引入非线性参数作为GOA递减系数以及位置权重系数，并结合limit阈值思想对GOA算法进行改进，提出一种基于非线性调整策略的改进蝗虫优化算法(Improved Grasshopper Optimization Algorithm，IGOA)．

2.1　非线性递减系数　GOA通过递减系数c来调整蝗虫个体间吸引力、排斥力以及个体搜索范围的大小．如式(5)所示，作用于括号内部的参数c有助于减少与算法迭代次数成比例的个体间的排斥力及吸引力，作用于括号外部的参数c则可随迭代次数的增加而降低个体的搜索覆盖范围．因此，GOA通过参数c来协调算法迭代过程中的全局探索和局部开发能力，但由式(6)可知，c随迭代次数增加而线性递减，会使算法的收敛速度变慢，易陷入局部最优．因此，本文采用非线性参数代替原始线性递减系数，将式(6)改写为：

$$c=\left[1-\sin\left(\frac{1}{2}\cdot\pi\cdot\sqrt{\frac{l}{L}}\right)\right]\cdot\left[c_{\max}-l\cdot\frac{c_{\max}-c_{\min}}{L}\right] \tag{7}$$

其中，l表示当前迭代次数，L为最大迭代次数．非线性变化的递减系数c能在算法迭代前期以更快的速率下降，使种群中的蝗虫个体迅速向目标靠近，提升算法收敛速度；而在算法迭代后期，c的递减速度减缓，使个体能对周围空间进行仔细搜索，避免算法陷入局部最优．由此，利用非线性变化的递减系数能更好地平衡算法在不同迭代时期的全局探索和局部开发能力．

2.2　自适应权重系数　在GOA中，当种群中所有个体都处于舒适区后将不再进行位置更新，此时种群中的个体没有聚集在最优解的位置，而是分布在其周围，这样会使算法容易出现早熟收敛的现象．由式(5)可知，蝗虫个体位置的更新不仅取决于种群中的其他个体，还依赖当前种群中的最优解，因此最优解的位置对其他个体的移动有重要影响．考虑到在算法迭代的不同时期，为寻找全局最优解所在的位置，种群中的个体对当前群体中最优解的依赖程度不同，本文引入非线性参数作为当前种群最优解的权重系数，定义如下：

$$\omega=1-\sin\left(\frac{\pi}{2}\cdot\sqrt{\frac{l}{L}}\right) \tag{8}$$

并将式(5)改写为：

$$X_i^d = c \cdot \left(\sum_{j=1, j \neq i}^{N} c \cdot \frac{ub_d - lb_d}{2} \cdot s\left(\left| x_j^d - x_i^d \right| \right) \cdot \frac{x_j - x_i}{d_{ij}} \right) + \omega \cdot \hat{T}_d \tag{9}$$

其中,l表示当前迭代次数,L为最大迭代次数.ω为本文所定义的权重系数,ω值随迭代次数增加呈非线性递减趋势,即,随着算法迭代,种群中最优解对于其他个体位置更新的影响也随之改变.算法迭代初期ω值较大,个体根据最优解及个体间相互的位置信息进行位置更新.而随着算法迭代的进行,为避免种群中的个体逐渐处于自身舒适区,应以较低的ω值降低种群中的个体对最优解位置的依赖,使种群中的个体能在最优解附近进行移动,从而避免所有个体全部停留在最优解周围的问题,这样就可以增强算法的局部开发能力,提高算法的寻优精度.

2.3 limit 阈值 为避免算法出现早熟收敛现象,受杨菊蜻等[13]的启发,本文引入limit阈值来判断算法是否陷入局部最优,通过设置limit阈值限定种群中最优解的停滞次数.当停滞次数达到所设阈值时,在种群中随机选取n个个体,利用非线性递减系数c对其进行扰动,改变个体所处位置,提高种群多样性,使算法跳出局部最优.limit阈值的设置需根据具体的问题来决定:阈值过高会无法及时使算法跳出局部最优;阈值过低则会频繁地对种群中的个体进行随机扰动,影响种群的平均适应度.本文多次改变limit阈值进行试验来选择试验效果最好的limit阈值,最终设置limit阈值为15.此外,对于随机选取的个体数n,由于较高的n值会使多数个体的位置发生改变,不利于种群进化的稳定性;而较低的n值则无法为种群提供足够的多样性.经多次测试,本文取$n=N/3$,N为种群中的个体数.为保证种群中的个体随算法迭代不断向最优解靠近,在进行limit判定前对位置更新后种群中的个体进行择优保留.

综上所述,IGOA算法流程如图1所示.

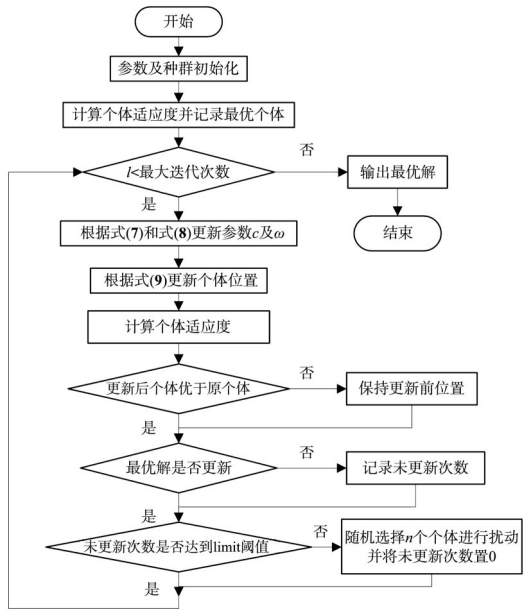

图1 IGOA算法流程图

Fig. 1 Flow chat of IGOA algorithm

3 基于IGOA的特征选择方法

特征选择问题可理解为一个多目标优化问题,即选择尽可能少的特征数使分类器获得尽可能高的分类准确率.本文利用IGOA算法来解决这一实际优化问题,提出一种基于IGOA的特征选择方法,具体算法流程如图2所示.

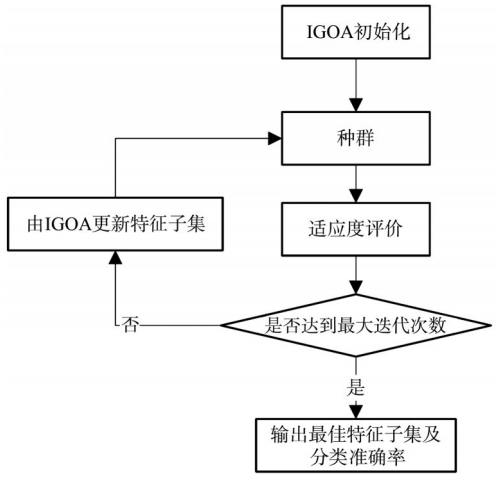

图2 基于IGOA的特征选择流程图

Fig. 2 Flow chart of feature selection based on IGOA

在特征选择问题中,IGOA种群中的每个个体都代表数据集中的一组特征组合,也即是所谓的特征子集.个体维度则由数据集中的原始特征数决定,并且每个个体向量均由0和1组成,1表示选取了对应的特征属性,0则表示该特征属性未被选取.在IGOA种群初始化时个体各维度的

取值为[0,1]的随机数,因此为使种群中的个体向量均为0和1组成,本文取个体各维度值大于0.65的值为1,其余值置0,得到由0和1组成的个体向量.为了以尽可能少的特征数获得尽可能高的分类准确率,评价个体好坏的适应度函数需同时考虑这两个因素,因此本文采用的适应度函数定义如下[14]:

$$Fitness=\alpha\cdot\gamma_R^D(D)+\beta\cdot\frac{|R|}{|N|} \tag{10}$$

其中,$\gamma_R(D)$为分类器错误率(本文采用KNN分类算法来评价特征子集的优劣(取K=5)),$|R|$为当前个体所包含特征数,$|N|$为数据集中原始特征数,α和β为平衡分类准确率及特征子集长度的协调参数,且$\beta=1-\alpha$,$\alpha\in[0,1]$,本文取α=0.99.

为了评价基于改进蝗虫优化算法的特征选择方法的优劣,本文选用分类器准确率、特征选择个数及特征选择率作为衡量指标.

分类准确率的定义如式(11)所示:

$$Accuracy=\frac{TP+TN}{P+N} \tag{11}$$

其中,TP,TN,P,N分别表示真正例、真负例、正和负样本数.

特征选择率的定义如式(12)所示:

$$FsRatio=\frac{1}{M}\sum_{i=1}^{M}\frac{size(\hat{g}^i)}{D} \tag{12}$$

其中,M为特征选择算法运行次数,D为数据集中原始特征数,$\hat{g}^i$为算法每次运行得到的最优特征子集,$size(x)$是向量x中元素1的个数.

4　实验结果及分析

仿真测试环境:Intel(R) Core(TM) i5-6500 CPU 3.2 GHz内存8 GB Windows7(64位)操作系统,所有算法均采用Matlab R2015b实现.为证明本文所提出的IGOA算法比传统蝗虫优化算法的寻优精度及收敛速度都有所提升,引入如表1所示的六个基准测试函数进行测试.为证明基于IGOA的特征选择方法的有效性,在七个UCI数据集上对算法进行测试.

表1　基准测试函数

Table 1　Benchmark functions

函数名	表达式	维度(Dim)	搜索空间	理论最优值
Sphere	$F_1=\sum_{i=1}^{Dim}x_i^2$	5/30	[−100,100]	0
Schwefel 2.22	$F_2=\sum_{i=1}^{Dim}\|x_i\|+\prod_{i=1}^{Dim}\|x_i\|$	5/30	[−10,10]	0
Schwefel 1.2	$F_3=\sum_{i=1}^{Dim}\left(\sum_{j-1}^{i}x_j\right)^2$	5/30	[−100,100]	0
Schwefel 2.21	$F_4=\max_i\{\|x_i\|,1\leqslant i\leqslant D\}$	5/30	[−100,100]	0
Rastrigin	$F_5=\left[x_i^2-10\cos(2\pi x_i)+10\right]$	5/30	[−5.12,5.12]	0
Ackley	$F_6=-20\exp\left(-0.2\sqrt{\frac{1}{Dim}\sum_{i=1}^{Dim}x_i^2}\right)-\exp\left(\frac{1}{Dim}\sum_{i=1}^{Dim}\cos(2\pi x_i)\right)+20+e$	5/30	[−32,32]	0

4.1　IGOA算法性能测试　本文在不同维度(Dim=5,30)的搜索空间中,利用六个基准测试函数测试IGOA算法性能,设置种群规模为30,最大迭代次数为500次.为获得更为客观真实数据,取算法独立运行30次后得到的最优解的均值和方差,与相同条件下的传统蝗虫优化算法以及

李洋州和顾磊[12]提出的最新的改进蝗虫优化算法(CAGOA2,SA-CAGOA2)进行对比,测试结果如表2所示(表中黑体字表示对比算法得到的最优值). 为证明IGOA算法比传统蝗虫优化算法的收敛速度更快,通过对比两种算法的收敛曲线,验证本文提出的IGOA算法的有效性及优越性,实验结果如图3和图4所示.

由表2可知,IGOA算法无论是在5维或是30维的搜索空间中,针对六个基准测试函数,算法的寻优精度及稳定性都明显优于传统蝗虫优化算法. 通过与李洋州和顾磊[12]提出的最新的改进蝗虫优化算法(包含CAGOA2与SA-CAGOA2两种方法,本文所对比的方法为原文中在$D=5$及$D=30$条件下对应的最优方法)相比较,IGOA同样能在函数F_5取到全局最优解,而且对函数F_6所求最优解的标准差也取到了0;而对于其余函数,IGOA所取得的最优解的均值及方差都明显优于该改进算法,也证明本文所提出的IGOA算法的有效性及优越性. 此外,由图3和图4可知,IGOA算法仅在五维搜索空间中对于函数F_5的收敛速度提升不太明显,而在其余不同维度的搜索空间中,IGOA在六个基准测试函数上的收敛速度都明显优于传统蝗虫优化算法.

表2 算法寻优性能对比

Table 2 Optimization performance of IGOA and other algorithms

函数	*Dim*		GOA	文献[12]	IGOA
F_1	5	Mean	1.74E−008	2.55E−013	**2.03E−035**
		Std. Dev	1.97E−008	5.68E−013	**8.82E−036**
	30	Mean	3.86E+001	6.62E−019	**1.21E−034**
		Std. Dev	2.97E+001	8.53E−019	**2.86E−035**
F_2	5	Mean	2.36E+000	1.49E+000	**3.85E−019**
		Std. Dev	2.88E+000	2.06E+000	**5.83E−020**
	30	Mean	1.68E+001	3.64E−010	**2.70E−018**
		Std. Dev	1.91E+001	3.63E−010	**4.66E−019**
F_3	5	Mean	8.27E−006	7.17E−008	**4.96E−035**
		Std. Dev	2.51E−005	2.38E−007	**4.58E−035**
	30	Mean	2.60E+003	6.78E−016	**1.02E−033**
		Std. Dev	1.67E+003	9.99E−016	**1.17E−033**
F_4	5	Mean	1.71E−004	1.17E−006	**2.82E−018**
		Std. Dev	2.71E−004	3.92E−006	**8.14E−019**
	30	Mean	1.50E+001	1.93E−010	**3.89E−018**
		Std. Dev	4.05E+000	2.15E−010	**4.29E−019**
F_5	5	Mean	1.11E+001	7.85E+000	**0.00E+000**
		Std. Dev	7.57E+000	5.19E+000	**0.00E+000**
	30	Mean	9.45E+001	**0.00E+000**	**0.00E+000**
		Std. Dev	3.30E+001	**0.00E+000**	**0.00E+000**
F_6	5	Mean	1.04E+000	7.42E−001	**8.88E−016**
		Std. Dev	2.52E+000	1.08E+000	**0.00E+000**
	30	Mean	5.50E+000	2.06E−010	**8.88E−016**
		Std. Dev	1.76E+000	2.03E−010	**0.00E+000**

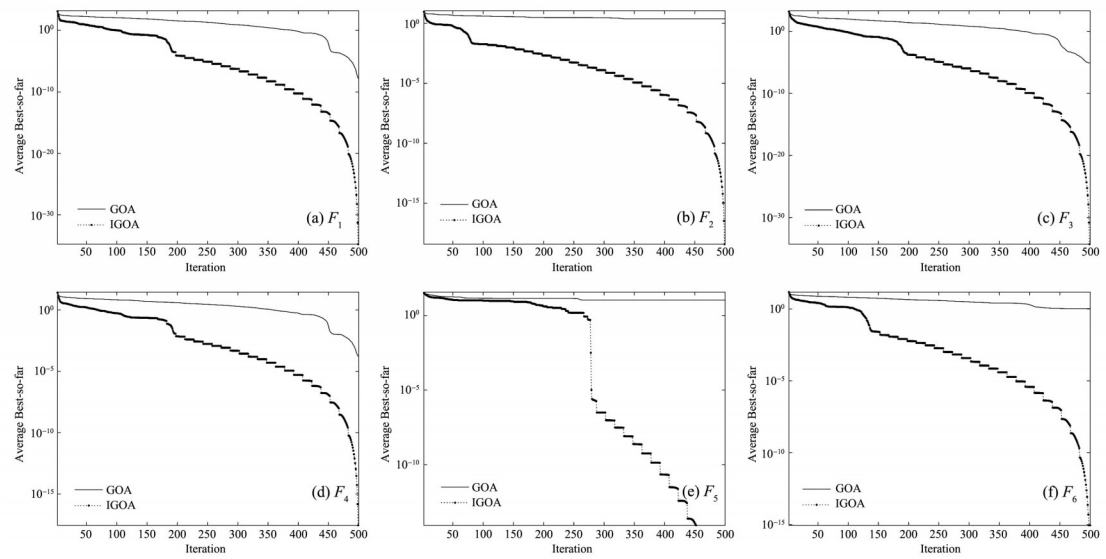

图3　IGOA和GOA算法的收敛曲线(*Dim*=5)

Fig.3　Convergence curve of IGOA and GOA algorithm (*Dim*=5)

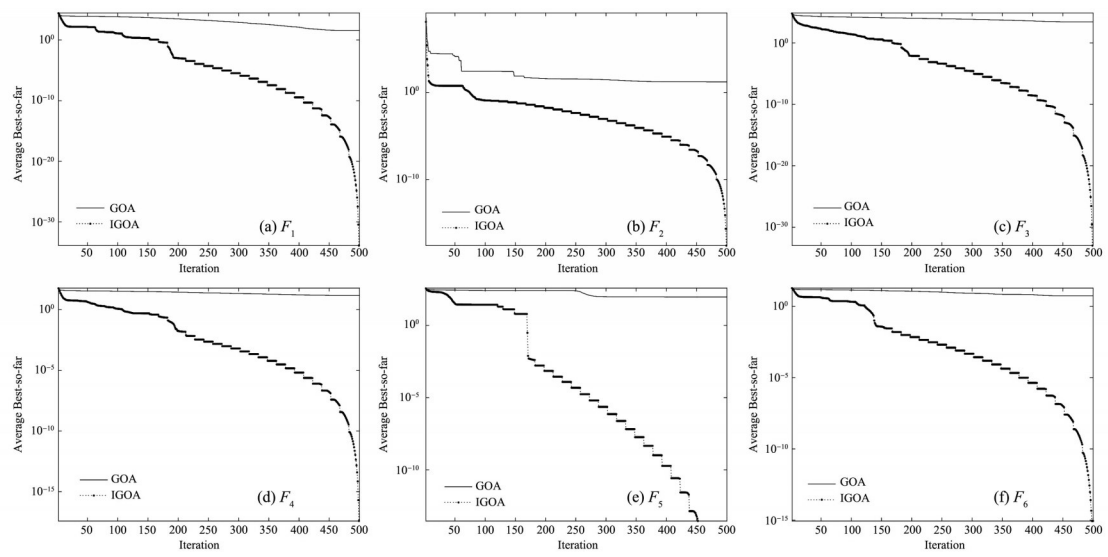

图4　IGOA和GOA算法的收敛曲线(*Dim*=30)

Fig.4　Convergence curve of IGOA and GOA algorithm (*Dim*=30)

传统蝗虫优化算法采用线性递减系数,所以无法有效平衡算法在迭代过程中的全局探索和局部开发能力.而本文的IGOA算法不仅采用非线性递减系数,同时还引入非线性权重系数和个体扰动策略,不仅能有效地平衡算法的全局探索和局部开发能力,提高算法的收敛速度,还可以避免算法陷入局部最优,提高算法寻优精度.因此,IGOA算法无论是寻优精度还是收敛速度都明显优于传统蝗虫优化算法,而且比李洋州和顾磊[12]提出的最新的改进算法也有明显优势.

4.2　基于IGOA的特征选择方法　为证明本文提出的基于IGOA的特征选择方法的有效性,在如表3所示的七个数据集上对算法进行测试.首先比较基于IGOA的特征选择方法(IGOA-FS)、基于传统蝗虫优化算法的特征选择方法(GOA-FS)以及采用全特征进行训练的KNN算法的性

能. 设置种群规模为30,算法最大迭代次数为100次,所有算法独立运行10次,取分类准确率均值及所选择的特征数来评价算法性能,测试结果如表4所示(表中黑体字为所对比算法中的最优值).

表3 实验数据集

Table 3 Experimental datasets

Datasets		特征个数	实例数
D1	BreastCancerEW	30	569
D2	Zoo	16	101
D3	Heart	12	270
D4	Parkinson	22	197
D5	Congress	16	435
D6	Wine	13	178
D7	Colon	2000	62

表4 算法在七个数据集上的特征选择性能的比较

Table 4 Feature selection performance of algorithms on seven datasets

数据集		FULL	GOA-FS	IGOA-FS
D1	Accuracy	0.951	0.959	**0.976**
	Features	30	**11.2**	13.5
D2	Accuracy	0.961	0.931	**0.963**
	Features	16	**6.6**	7.1
D3	Accuracy	0.763	0.768	**0.801**
	Features	12	6.6	**6.4**
D4	Accuracy	0.908	**0.949**	**0.949**
	Features	22	8.9	**8.4**
D5	Accuracy	0.940	0.945	**0.970**
	Features	16	5.5	**3.3**
D6	Accuracy	0.944	0.951	**0.960**
	Features	13	6.2	**5.8**
D7	Accuracy	0.677	0.745	**0.833**
	Features	2000	**675.2**	691.9

由表4可知,本文提出的IGOA-FS方法所选特征子集的长度仅在D1,D2及D7数据集上略多于GOA-FS方法,但其分类准确率明显优于GOA-FS以及未进行特征选择的KNN算法. 并且,在其余四个数据集上,IGOA-FS无论是分类准确率还是所选特征子集的长度都在三种方法中均为最优. 尤其和未进行特征选择的KNN算法相比,IGOA-FS不仅提高了算法的分类准确率,还能大幅减少算法训练所需的特征数. IGOA-FS算法将特征选择问题转化为函数最优解的求解问题,由于其函数优化效果已被证明明显优于GOA算法,因此和GOA-FS算法及采用全特征的KNN算法相比,IGOA-FS能找到最佳特征子集,提高分类精度,证明IGOA-FS方法能有效地进行特征选择,减少冗余特征对分类器性能的影响.

为了比较本文提出的IGOA-FS方法与其他基于群智能优化的特征选择方法的性能优劣,将其与Mafarja and Mirjalili[14]提出的基于鲸鱼优化的特征选择方法、Emary et al[15]提出的基于蚁狮优化的特征选择方法以及Sayed et al[16]提出的基于混沌乌鸦搜索的特征选择方法相对比,分类准确率对比结果如表5所示(表中黑体字为对比算法得到的最优值,"—"表示参考文献未给出相应数据),算法的平均特征选择率如图5所示.

表5 IGOA-FS与其他算法的性能对比

Table 5 Performance of IGOA - FS and other algorithms on seven datasets

Data set	ALO[15]	CCSA[16]	WOA-CM[14]	IGOA-FS
D1	0.930	0.903	0.971	**0.976**
D2	0.909	0.937	**0.980**	0.963
D3	**0.826**	0.788	0.807	0.801
D4	—	0.908	—	**0.949**
D5	0.929	—	0.956	**0.970**
D6	0.911	—	0.959	**0.960**
D7	—	—	**0.909**	0.833

由表5可知,在特征数较少的数据集上,IGOA-FS的分类准确率仅在D2,D3,D7数据集上略劣于Mafarja and Mirjalili[14]和Emary et al[15]提出的方法,而在其余四个数据集上的分类准确率均明显高于其他对比算法. 根据图5可知,IGOA-FS的特征选取率仅在D2及D3数据集上略高于WOA-CM以及CCSA,而在其余数据集上的特征选择率均低于其他对比算法,也就是说,IGOA-FS在其余数据集上不仅能获得更高的分

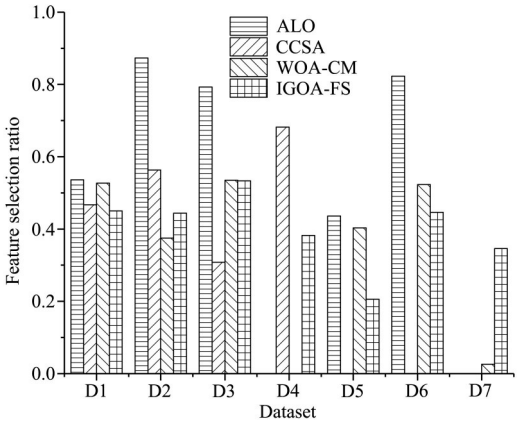

图5　IGOA-FS和其他算法在七个数据集上的平均特征选取率对比

Fig. 5　Average feature selection ratio of IGOA-FS and other algorithms on seven datasets

类准确率,而且所选择的特征子集的长度比其他对比算法更低,特征选择性能更好.

而在特征数较多的数据集D7上,IGOA-FS的分类精度及特征选择率虽然略劣于WOA-CM,但与GOA-FS及采用全特征进行训练的KNN算法相比仍具有明显优势,证明IGOA-FS算法能在特征数较多的数据集上进行有效的特征选择,而且在特征维度较高的情况下[17],算法的性能仍具有一定的提升空间.

综上所述,本文提出的IGOA-FS算法能够有效地进行特征选择,降低数据维度,提高算法分类性能,和其他特征选择算法相比有明显的优势.

5　结　论

首先针对传统蝗虫优化算法寻优精度低、收敛速度慢的问题,采用三种策略进行改进,并通过基准测试函数证明所提出的改进算法IGOA在寻优精度和收敛速度方面均有明显提升.将改进算法应用于特征选择问题,提出了一种基于改进蝗虫优化算法的特征选择方法IGOA-FS,并在七个数据集上对算法进行了测试,证明该方法能够有效地进行特征选择,提高分类器性能.最后,通过与其他特征选择算法进行对比,证明本文提出的方法确实具有一定优势.如何对算法进行改进,使其能够在更高的特征维度下仍具备优异性能将是下一步的主要研究内容.

参考文献

[1] 李炜,巢秀琴.改进的粒子群算法优化的特征选择方法.计算机科学与探索,2019,13(6):990－1004.(Li W, Chao X Q. Improved particle swarm optimization method for feature selection. Journal of Frontiers of Computer Science and Technology, 2019,13(6):990－1004.)

[2] 张震,魏鹏,李玉峰等.改进粒子群联合禁忌搜索的特征选择算法.通信学报,2018,39(12):60－68.(Zhang Z, Wei P, Li Y F, et al. Feature selection algorithm based on improved particle swarm joint taboo search. Journal on Communications, 2018, 39(12):60－68.)

[3] Gao W F, Hu L, Zhang P, et al. Feature selection by integrating two groups of feature evaluation criteria. Expert Systems with Applications, 2018, 110: 11－19.

[4] Mafarja M M, Mirjalili S. Hybrid whale optimization algorithm with simulated annealing for feature selection. Neurocomputing, 2017, 260:302－312.

[5] Kennedy J, Eberhart R. Particle swarm optimization//Proceedings of ICNN′95-International Conference on Neural Networks. Perth, Australia: IEEE, 1995:1942－1948.

[6] Mirjalili S. The ant lion optimizer. Advances in Engineering Software, 2015, 83:80－98.

[7] Mirjalili S, Lewis A. The whale optimization algorithm. Advances in Engineering Software, 2016, 95:51－67.

[8] Saremi S, Mirjalili S, Lewis A. Grasshopper optimisation algorithm: theory and application. Advances in Engineering Software, 2017, 105: 30－47.

[9] Ewees A A, Elaziz M A, Houssein E H. Improved grasshopper optimization algorithm using opposition-based learning. Expert Systems with Applications, 2018, 112:156－172.

[10] Luo J, Chen H L, Zhang Q, et al. An improved grasshopper optimization algorithm with application to financial stress prediction. Applied Mathematical Modelling, 2018, 64:654－668.

[11] Arora S, Anand P. Chaotic grasshopper optimization algorithm for global optimization. Neural Computing

and Applications, 2019, doi: 10.1007/s00521-018-3343-2.

[12] 李洋州,顾磊. 一种基于曲线自适应和模拟退火的蝗虫优化算法. 计算机应用研究, 2019, doi: 10.19734/j.issn.1001-3695.2018.07.0580. (Li Y Z, Gu L. Grasshopper optimization algorithm based on curve adaptive and simulated annealing. Application Research of Computers, 2019, doi: 10.19734/j.issn.1001-3695.2018.07.0580.)

[13] 杨菊蜻,张达敏,何锐亮等. 基于Powell搜索的混沌鸡群优化算法. 微电子学与计算机, 2018, 35(7): 78—82. (Yang J Q, Zhang D M, He R L, et al. A chaotic chicken optimization algorithm based on powell search. Microelectronics & Computer, 2018, 35(7): 78—82.)

[14] Mafarja M, Mirjalili S. Whale optimization approaches for wrapper feature selection. Applied Soft Computing, 2018, 62: 441—453.

[15] Emary E, Zawbaa H M, Parv B. Feature selection based on antlion optimization algorithm// 2015 3rd World Conference on Complex Systems (WCCS). Marrakech, Morocco: IEEE, 2015: 1—7.

[16] Sayed G I, Hassanien A E, Azar A T. Feature selection via a novel chaotic crow search algorithm. Neural Computing and Applications, 2019, 31(1): 171—188.

[17] Yu L, Liu H. Feature selection for high-dimensional data: a fast correlation-based filter solution// Proceedings of the 20th International Conference on Machine Learning. Washington DC, USA: AAAI Press, 2003: 856—863.

(责任编辑 杨可盛)