DOI:10.19651/j.cnki.emt.2314287

面向特征选择任务的改进蜣螂优化算法*

李 珺 徐 秦
(东北林业大学计算机与控制工程学院 哈尔滨 150000)

摘 要:蜣螂优化算法是一种基于蜣螂不同行为模式的新型启发式算法,与其他算法相比的收敛速度更快,逃脱局部最优的能力更强。针对蜣螂优化算法不能进行特征选择的问题,在蜣螂优化算法的基础上提出了蜣螂灰狼融合算法。该算法基于 3 种改进策略:精英初始化种群策略、灰狼蜣螂融合策略、运行加速策略,进一步提高蜣螂优化算法在特征选择任务上的性能,并给出了算法整体的伪代码。实验结果表明,比较其他改进型启发式算法,蜣螂灰狼融合优化算法在 12 个分类数据集中能够得到更高精度、更低维度的特征子集,同时兼备收敛速度、运行速度更快的优点。

关键词:特征选择;蜣螂优化算法;分类
中图分类号:TP301.6 **文献标识码**:A **国家标准学科分类代码**:510.40

Improved dung beetle optimization for feature selection tasks

Li Jun Xu Qin
(College of Computer and Control Engineering, Northeast Forestry University, Harbin 150000, China)

Abstract: The dung beetle optimization (DBO) algorithm is a novel heuristic algorithm inspired by the behaviors of dung beetles. It exhibits faster convergence speed and stronger ability to escape local optima compared to other algorithms. However, the DBO algorithm lacks the capability of performing feature selection. In this paper, propose algorithm of dung beetle and grey wolf fusion (DBOG) as an improvement to the DBO algorithm specifically designed for feature selection tasks. The DBOG incorporates three enhancement strategies: elite initialization population strategy, grey wolf-dung beetle fusion strategy, and runtime acceleration strategy. These strategies aim to further enhance the performance of the DBO algorithm in feature selection tasks. Additionally, we provide pseudocode for the overall algorithm. Experimental results demonstrate that, compared to other improved heuristic algorithms, the DBOG achieves higher accuracy and lower-dimensional feature subsets across 12 classification datasets. Moreover, it offers advantages such as faster convergence speed and computational efficiency.

Keywords: feature selection; dung beetle optimizer; classification

0 引 言

特征选择是机器学习中的一个重要步骤,高维冗余的特征子集会降低机器学习模型的拟合能力和可解释性[1]。特征子集中的特征很多是与目标属性无关的,或者这部分特征的作用可以被其他特征替代,也就是冗余的,而特征之间的不同组合对机器学习模型性能的影响也是不同的[2]。在这种情况下,如何挖掘出具有最低特征维度同时发挥出最高分类精度的特征子集就尤为重要。

启发式搜索算法的策略相较于过滤式、嵌入式、穷举搜索等其他特征选择策略的优点是平衡了算法的时间复杂度、精度和搜索的深度[3]。一些传统的启发式搜索算法已经在特征选择领域有广泛应用,例如遗传算法(genetic algorithm, GA)[4]、粒子群算法(particle swarm optimization, PSO)[5]等,近年来一些新型启发式搜索算法的改进被证明在特征选择领域也有良好的性能,例如哈里斯鹰优化算法(Harris hawks optimization, HHO)[6]、灰狼优化算法(grey wolf optimizer, GWO)[7]、樽海鞘优化算法(salp swarm algorithm, SSA)[8]、黑寡妇蜘蛛生殖优化算法(black widow optimization, BWO)[9]等。因此,利用启发式搜索算法进行特征选择是一种合理且高效的方法。

但根据无免费午餐定理,这些算法在面向不同的任务时需要根据实际情况进行改进,以达到更好的效果。现有的算法改进策略可以分为 3 类:优化种群初始化方式[10-12]、改进算法学习策略[13-15]、算法混合策略[16-18]。具体来讲,初

收稿日期:2023-08-03
* 基金项目:国家自然科学基金面上项目(52071074)资助

始化方式的改进可以增强算法初始的多样性，让算法以更优秀的起点出发，但复杂的初始化方式会增加算法的计算成本；改进算法学习策略，主要通过引进诸如莱维飞行系数、柯西变异系数、relief 系数、对立学习等方法，对产生的个体解进行扰动产生新解，进而跳出局部最优。这种方式可以一定程度上改善最优个体和种群的表现，但引入的系数需要通过经验和大量实验进行调整，且效果的随机性较大；算法的融合策略，利用各算法的优势互补，有助于克服单一算法的局限性，但算法之间的交互带来了更多的计算复杂性。

蜣螂优化算法（dung beetle optimization，DBO）是由 Shen 等在 2022 年 11 月提出的一种新型启发式搜索算法，通过模拟蜣螂的行为来实现全局搜索和求解最优值，该算法在一众基准函数和测试函数中表现优于许多其他优化算法[19]。但当蜣螂优化算法面向特征选择任务时，会出现两个问题：一是蜣螂优化算法不能直接运行特征选择任务，还未有直接将蜣螂优化算法应用于特征选择算法的文献；二则与蜣螂优化算法的运行机制相关：蜣螂优化算法本质属于亚群类型的算法，即划分种群为不同的个体，分别按照不同的公式更新个体解。这极大扩展了算法的搜索空间，然而在面对特征选择任务时，蜣螂优化算法的部分公式效果不尽人意，浪费了本就有限的计算资源。

针对蜣螂优化算法存在的以上问题，使用精英初始化种群策略改进种群的初始化分布，通过与灰狼优化算法融合改善蜣螂优化算法的部分公式在面对特征选择任务表现不佳的现象。同时，为了改善算法融合和初始化策略带来的计算复杂度增长问题，又提出了运行加速策略以改善算法的运行表现，使得算法在保证效果的同时拥有更快的运行速度。

1　DBOG 算法

1.1　二进制策略

使用启发式算法进行特征选择属于包装式方法，可以分为两个步骤：第 1 步，算法按照其设定准则生成种群，种群中的个体即为一个个特征子集；第 2 步，使用机器学习算法参与的评分函数评估种群中个体的优劣，循环这两个步骤一直到某个停止标准。原始的蜣螂优化算法是在连续的搜索空间中寻找最优解，而特征选择任务则是在离散的二进制空间中寻找最优解，所以需要二进制策略将蜣螂优化算法产生的解编码为特征子集。

二进制策略就是把算法产生的每一个解离散化为二值 0～1 向量，0 和 1 分别对应该位置的特征是否被选中。采用阈值法将种群的个体解映射为 0～1 向量，即：

$$U(d)=\begin{cases}1, & X(d)\geqslant 0.5\\ 0, & X(d)<0.5\end{cases} \tag{1}$$

$U(d)$ 即为向量的第 d 维，即第 d 个特征。$X(d)$ 则对应算法更新的个体解。

如图 1 所示，该特征向量的含义为：数据集的特征维度为 10，而算法产生的某个解，是第 2、5 和 8 位特征被选中。

特征子集	0	0	1	0	0	1	0	0	1	0
特征序号	0	1	2	3	4	5	6	7	8	9

图 1　特征选择任务中的二进制向量说明

1.2　精英初始化种群策略

算法开始运行时，需要给予种群中的每个个体一个初始解。为了使算法快速收敛到最优位置，对算法的初始化进行改进，提出一种精英初始化种群策略。

Logistic 混沌映射生成的随机数随机性强、分布均匀等特点[20]，故使用 Logistic 混沌映射代替随机值进行初始种群的随机化。Logistic 混沌映射公式如下：

$$X(k+1)=\mu\cdot x(k)\cdot(1-x(k)) \tag{2}$$

其中，μ 为分支参数，范围为(3.569 945 6,4]，$x(k)$ 为当前产生的随机值，初始值是范围在(0,1)内的随机数。

为了使种群能够在初始阶段就尽可能多的探索解空间，对生成的解进行反向，生成更多的解。种群中第 K 个个体的反向解 $\boldsymbol{X}_f$ 的求取公式如下：

$$X_f(k)=\mathrm{lb}+\mathrm{ub}-X(k) \tag{3}$$

lb 和 ub 分别为种群的上界和下界。

但通过随机数生成的初始解具有随机性，不能保证一定处于较优位置。因此，本文先使用两种抽样方法：自助法、3 折交叉采样对所有样本进行抽样，以改善数据集中样本分布不均和噪声等问题，提高模型的鲁棒性。再对每一份样本计算其 RliefF 特征重要性系数，得到每个特征的 6 个特征重要性系数，求其平均值。根据该平均值，得到每个特征的重要性排序，进而计算种群个体解的每个编码值的缩放倍率 Z，求取公式如下：

$$Z=l+((D-R)\times(u-l))/D \tag{4}$$

$$\boldsymbol{X}'=\boldsymbol{X}\cdot Z \tag{5}$$

$\boldsymbol{X}'$ 即为更新后的精英个体解。其中，设定 l 为 0.5，u 为 1.5，D 为维度数，R 为当前特征在鲁棒特征排序中的位置。整个算法的初始化策略流程如图 2 所示。

1.3　灰狼蜣螂融合机制

原始的蜣螂优化算法通过种群中不同的蜣螂行为模式实现算法的寻优，但一些行为在特征选择任务上的表现并不优秀，因此引入灰狼优化算法[21]替代换蜣螂的滚球（含跳舞）、觅食这两种行为，提出基于蜣螂优化算法与灰狼优化算法融合的 DBOG（dung beetle optimization gray-wolf）算法。算法的模型如下：

首先在种群中选择最优秀的 3 只蜣螂 a、b、c，种群中原本的滚球和觅食蜣螂围绕这 3 只蜣螂进行位置更新：

$$\begin{aligned}\boldsymbol{X}1&=\boldsymbol{Xa}-\boldsymbol{A}1\cdot\boldsymbol{Da}\\ \boldsymbol{X}2&=\boldsymbol{Xb}-\boldsymbol{A}2\cdot\boldsymbol{Db}\\ \boldsymbol{X}3&=\boldsymbol{Xc}-\boldsymbol{A}3\cdot\boldsymbol{Dc}\end{aligned} \tag{6}$$

式中：$\boldsymbol{X}1$、$\boldsymbol{X}2$、$\boldsymbol{X}3$ 是当前蜣螂分别向 a、b、c 移动的步长。

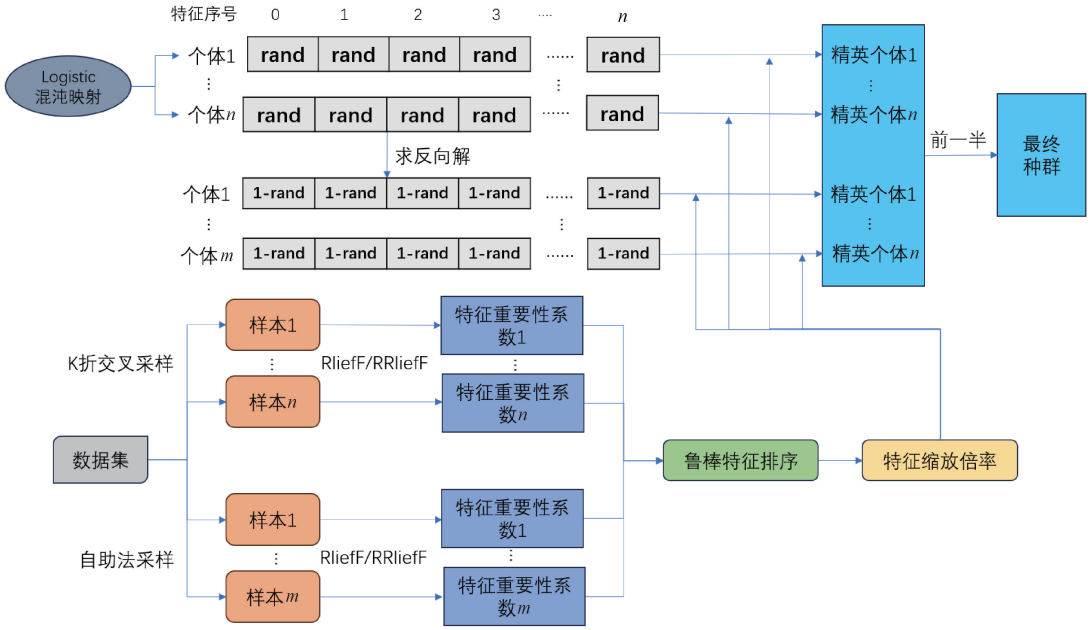


图 2　精英初始化策略

$\boldsymbol{Xa}$、$\boldsymbol{Xb}$、$\boldsymbol{Xc}$ 分别为当前 a、b、c 的位置。式中的 $\boldsymbol{Da}$、$\boldsymbol{Db}$、$\boldsymbol{Dc}$ 分别表示当前蜣螂与 a、b、c 的距离，$\boldsymbol{A}1$、$\boldsymbol{A}2$、$\boldsymbol{A}3$ 是系数向量。$\boldsymbol{Da}$、$\boldsymbol{Db}$、$\boldsymbol{Dc}$ 的计算公式如下：

$$\begin{aligned}\boldsymbol{Da} &= \boldsymbol{C}1 \cdot \boldsymbol{Xa} - \boldsymbol{X}(t)\\ \boldsymbol{Db} &= \boldsymbol{C}2 \cdot \boldsymbol{Xb} - \boldsymbol{X}(t)\\ \boldsymbol{Dc} &= \boldsymbol{C}3 \cdot \boldsymbol{Xc} - \boldsymbol{X}(t)\end{aligned} \tag{7}$$

式中：$\boldsymbol{C}1$、$\boldsymbol{C}2$、$\boldsymbol{C}3$ 是随机向量。计算公式为：$\boldsymbol{C} = 2 \cdot \boldsymbol{r}2$

$\boldsymbol{A}1$、$\boldsymbol{A}2$、$\boldsymbol{A}3$ 的计算公式为：$\boldsymbol{A} = 2d \cdot \boldsymbol{r}1 - d$

其中，$\boldsymbol{r}1$ 和 $\boldsymbol{r}2$ 为两个随机向量，每个维度的值都在 0～1 之间。d 为收敛因子，从 2 到 0 线性递减。计算公式如下：$d = 2 - t \cdot (\frac{2}{t_{\max}})$，$t$ 为当前迭代次数。

替换后的滚球蜣螂和觅食蜣螂的位置更新公式即为：

$$\boldsymbol{X}(t+1) = (\boldsymbol{X}1 + \boldsymbol{X}2 + \boldsymbol{X}3)/3 \tag{8}$$

$\boldsymbol{X}(t+1)$ 是当前蜣螂在本轮迭代中的下一个位置。

而繁殖和小偷蜣螂则按照原本的位置更新方式进行更新，繁殖蜣螂会先确定一个繁殖的区域：

$$\begin{aligned}\boldsymbol{Lb}^* &= \max(\boldsymbol{X}^* \cdot (1-R), \boldsymbol{Lb})\\ \boldsymbol{Ub}^* &= \min(\boldsymbol{X}^* \cdot (1+R), \boldsymbol{Ub})\end{aligned} \tag{9}$$

式中：$\boldsymbol{Lb}^*$ 和 $\boldsymbol{Ub}^*$ 分别是产卵区域的下界和上界，其中 $R = 1 - t/\mathrm{T}_{\max}$，$\boldsymbol{X}^*$ 为当前局部最优位置。

繁殖蜣螂产下的卵球位置更新公式如下：

$$\boldsymbol{X}(t+1) = \boldsymbol{X}^* + \boldsymbol{b}1 \cdot (\boldsymbol{X}(t) - \boldsymbol{Lb}^*) + \boldsymbol{b}2 \cdot (\boldsymbol{X}(t) - \boldsymbol{Ub}^*) \tag{10}$$

式中：$\boldsymbol{b}1$ 和 $\boldsymbol{b}2$ 为两个大小为 $1 \times D$ 的随机向量，D 为数据集的维度。

小偷蜣螂的位置更新公式：

$$\boldsymbol{X}(t+1) = \boldsymbol{X}^b + \mathrm{S} \cdot \boldsymbol{g} \cdot (|\boldsymbol{X}(t) - \boldsymbol{X}^*| + |\boldsymbol{X}(t) - \boldsymbol{X}^b|) \tag{11}$$

式中：$\boldsymbol{g}$ 为随机向量，服从正态分布，大小为 $1 \times D$，S 为常量，$\boldsymbol{X}^b$ 为全局最优位置。

算法的种群组成比例为：以灰狼模式迭代的蜣螂为 0.3，繁殖蜣螂为 0.3，小偷蜣螂为 0.4。

1.4　运行加速策略

由于这些启发式搜索算法的“逐优”机制，即群体中其他个体都会向最优秀的一些个体靠拢，在算法运行后期，许多个体会产生相似的解，这些相似的解在特征选择任务上被映射为二值向量后，便有可能会产生相同的向量。重复的计算相同向量所对应的适应度值会浪费大量的时间，因此本文设计一种运行加速策略如图 3 所示。

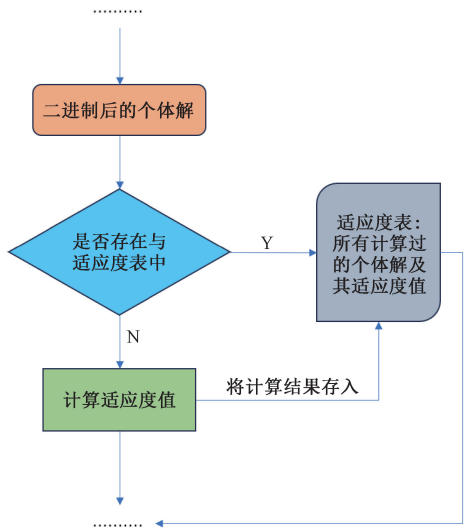


图 3　算法加速策略示意图

受禁忌搜索算法的启发，仿照禁忌表的概念，建立适应度表，存储种群中每个个体产生的二值化向量及其适应度值，在计算适应度值之前，检索是否有相同的向量已经计算过适应度值，如果计算过，则直接取用表中的值。如果没有计算过，则计算后存储入表内。

1.5　适应度函数设计

适应度函数值计算公式如式(11)：

$$fitness = \alpha \cdot error + \beta \cdot (1 - \frac{n}{N}) \tag{12}$$

$$error = 1 - acc \tag{13}$$

由于面向分类任务，使用分类准确率 accuracy 作为 acc。其中 α 设置为 0.99，β 设置为 0.01，n 为解中包含的特征数量，N 为特征总数。

1.6　算法的整体描述

为了更好的说明算法的整体流程，给出算法的伪代码表如表 1。

表 1　DBOG 算法的伪代码表

The framework of the DBOG algorithm
1. Initialize the population $\boldsymbol{X}$ using the initialization strategies (2,3,4,5) for i=1 to n.
2. Calculate the fitness values of the population and select the top three best grey wolves, and their corresponding solutions Xa, Xb, Xc.
3. **while** (t <=T)
4.　**for** i=1 to N
5.　**if** i==grey wolf then
6.　Initialize parameters $\boldsymbol{d}$, A and C
7.　update grey wolf position by using (8)
8.　**end if**
9.　**if** i==brood ball dung then
10.　update brood ball beetle position by using (10)
11.　**end if**
12.　**if** i ==thief dung beetle then
13.　update dung beetle position by using (11)
14.　**end if**
15. **end for**
16.　**if** the newly generated position is better than before then
17.　Update it
18.　**end if**
19.　Update the first three grey wolves Xa, Xb, and Xc
20.　t=t+1
21. **end while**
22. **return** the best grey wolf Xa

DBOG 算法的时间复杂度分析：假设数据集有 D 个特征，算法的种群数量为 N，迭代次数为 T，种群适应度排序的时间复杂度为 $O(Y)$，每种采样方法的 Relief 重要性系数时间复杂度为 $O(R)$。循环开始时，按照精英初始化策略进行种群初始化的时间复杂度为 $O(Y\times2\times D\times N+2\times R)$，灰狼行为、繁殖蜣螂行为、小偷蜣螂行为以及最后更新种群排序的时间复杂度为 $O(Y\times T\times N\times D+Y)$，则 *DBOG* 算法的总时间复杂度为 $O(2\times R+Y+Y\times N\times D\times(T+2))$。

2　实验与讨论

2.1　数据集信息及对比算法

选取 UCI 数据库中的 12 个数据集在相同实验条件下对 DBOG 与对比算法进行实验，选取的数据集涵盖了从 19 个特征的低维数据集到 1 203 个特征的高维数据集，样本数从 87 个小样本数据集到 6 598 个样本充足的数据集均有涵盖。表 2 是其详细信息。

表 2　数据集的信息描述

数据集	特征数	样本数	类别数
absentism	19	740	2
ChurnData	27	200	2
Darwin	450	174	2
Toxicity 2	1203	171	2
Musk 1	163	476	2
Musk 2	163	6 598	2
LSVT	313	126	2
Flowmeters A	35	87	2
Flowmeters B	50	92	3
Flowmeters C	42	181	4
Flowmeters D	42	180	4
ParkinsonDatabase	46	240	2

为了证明改进后的蜣螂优化算法在特征选择任务上的有效性，选取近 5 年来较新颖、效果较好、覆盖不同改进策略的 5 种改进算法进行对比。首先是两阶段融合变异灰狼特征选择算法（two-phase mutation grey wolf optimizer，TMGWO），采用了遗传算法的交叉变异优化灰狼优化算法迭代机制，相比较原始的灰狼优化算法在特征选择任务上表现更为优秀[22]。由于 DBOG 算法同样采用了灰狼优化算法机制，故采用 TMGWO 算法作为对照。随后是基于对立学习和新型局部搜索算法的樽海鞘特征选择算法（improved salp swarm algorithm，ISSA）[23]与基于对立学习的鲸鱼优化算法（opposition-based whale optimization algorithm）[24]，该两种算法使用对立学习策略的方式优化个体解和初始解，提高算法的性能。改进的全局花卉授粉算法（modified global flower pollination algorithm，MGFPA）[25]采用 HBSS 有界搜索空间机制调

配两个父代个体进而产生更优秀的个体解。

为了进一步证明改进的先进性,选取改进正弦算法引导的蜣螂优化算法(modified sine algorithm dung beetle optimization,MSADBO)作为对比,该算法首先使用Bernoulli映射改进种群初始化方式,采用改进的正弦算法替代滚球蜣螂的跳舞行为公式,使用自适应高斯-柯西变异扰动最优个体值,在测试函数上取得了较好的效果[26]。对比算法详细信息及参数设置如表3所示。

表3　对比算法的详细信息及实验参数设置

算法名称	描述	参数设置
TMGWO	两阶段融合变异灰狼特征选择算法	MU=0.5
ISSA	基于对立学习和新型局部搜索算法的樽海鞘特征选择算法	LSA=10
OBWOA	基于对立学习的鲸鱼优化算法	$B=1$
MGFPA	改进的全局花卉授粉算法	Gamma=0.01,beta=1.5,P=0.8
DBO	蜣螂优化算法	$S=0.5$,$k=0.1$,$b=0.3$
MSADBO	改进正弦算法引导的蜣螂优化算法	$\omega_{max}=0.9$,$\omega_{min}=0.782$,$k=0.1$,$b=0.3$,$R=1$

2.2　实验环境及参数设置

算法的软件环境为python3.10,硬件环境CPU为i3-10100,RAM为16 GB,硬盘1 TB。

DBOG算法与对比算法的结果均采用独立运行20次重复实验后计算的指标平均值来作为检验标准,采用10折交叉验证,数据集中的80%为训练集,20%为测试集。分类器则采用KNN分类器,其中$K=5$。所有算法的种群规模为30,迭代次数为50。评价指标选取3种:运行时间T、特征维度FD、分类准确率ACC。

2.3　实验结果与讨论

DBOG算法与对比算法在12个数据集上的平均分类准确率对比结果如表4所示。

表4　DBOG与5种对比算法在平均分类准确率上的对比

数据集	DBOG	DBO	ISSA	TMGWO	OBWOA	MGFPA	MSADBO
Absentism	**0.995 9**	0.992 4	0.994 5	0.995 5	0.995 5	0.994 9	0.987 4
ChurnData	**0.807 9**	0.787 9	0.805 5	**0.807 9**	0.789 5	0.809 4	0.796 9
Darwin	**0.882 6**	0.846 5	0.811 8	0.842 2	0.845 2	0.836 2	0.857 3
Toxicity 2	**0.713 7**	0.683 5	0.617 4	0.683 2	0.683 3	0.632 9	0.704 3
Musk 1	**0.872 1**	0.844 8	0.846 2	0.845 8	0.827 5	0.861 2	0.835 5
Musk 2	**0.899 5**	0.886 5	0.864 0	0.886 5	0.868 6	0.884 2	0.894 5
LSVT	0.816 6	0.797 9	0.667 1	0.796 1	0.800 8	0.664 2	**0.817 1**
Flowmeters A	**0.754 7**	0.690 8	0.705 5	0.733 8	0.713 7	0.715 0	0.715 5
Flowmeters B	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Flowmeters C	**0.915 9**	0.907 1	0.910 6	0.910 0	0.901 3	0.908 5	0.907 2
Flowmeters D	**0.849 9**	0.840 5	0.847 7	0.849 4	0.845 5	0.849 9	0.847 7
ParkinsonDatabase	0.828 7	0.820 4	**0.835 0**	0.821 6	0.822 9	0.832 0	0.825 4

由于适应度函数被设计为综合考虑特征维度和分类准确率,当算法间的收敛的最终效果也就是最低适应度值没有足够大的差距时,就有可能造成某些DBOG算法在特征维度和分类准确率这两种指标中的某一种无法优越于其他对比算法。可以看到,DBOG算法在12个数据集中的11个数据集中能够取得平均分类准确率的最高值,在ParkinsonDatabase数据集中虽然没能取得最高精度,但相较对比算法中的其中3个算法仍取得了较好的成绩。

DBOG算法与其他5种对比算法在运行时间和特征维度上的比较结果如表5和6所示。

在运行时间和特征维度指标上,DBOG相比较对比算法在运行时间指标上在11个数据集上取得领先,对比未改进之前的DBO算法,说明运行加速策略是有效的。而在特征维度指标上,DBOG在9个数据集上维持优势,说明DBOG算法能够在保持最高分类精度的情况下,以更快的运行速度选择最小的特征维度,更好的完成特征选择任务。

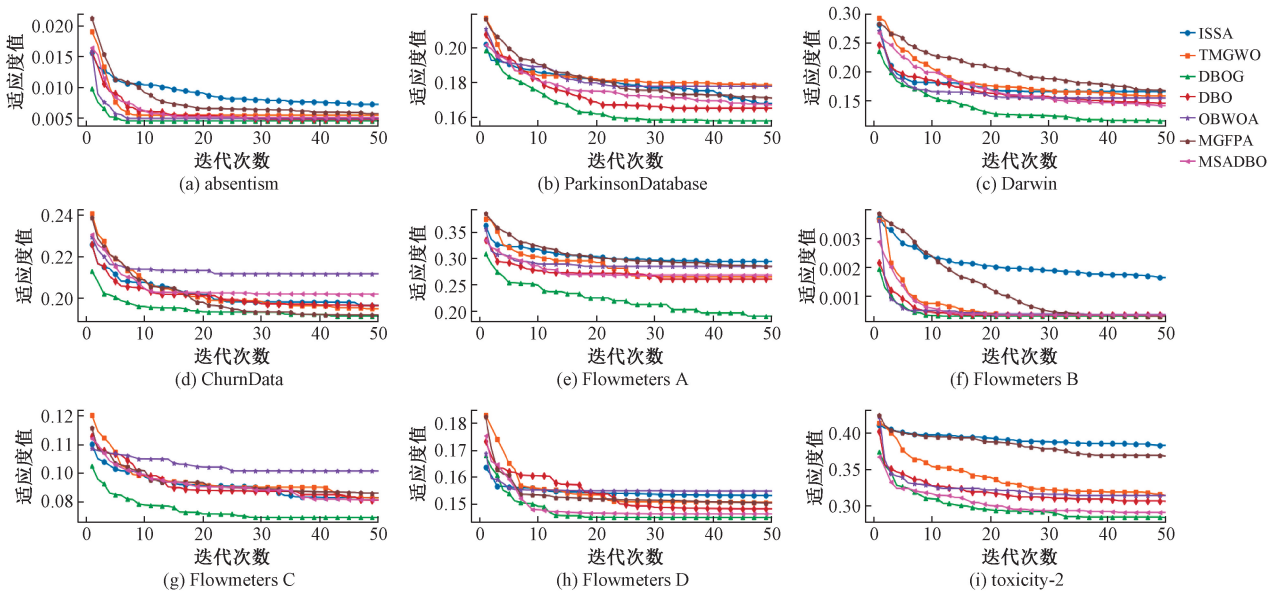
为了证明DBOG算法寻找最优值时的收敛速度优势以及精英化种群初始策略的有效性,给出DBOG算法与其他算法的收敛对比如图4所示。

表 5　DBOG 与 5 种对比算法在运行时间上的对比

数据集	DBOG	DBO	ISSA	TMGWO	OBWOA	MGFPA	MSADBO
absentism	**18**	69.1	121.3	106.3	204.9	71.4	76.4
ChurnData	**37.3**	64.2	85.9	98.5	138.7	64.2	71.4
Darwin	**121.2**	167.0	236.2	1 224.5	599.6	143.9	133.4
Toxicity 2	**123.5**	203.0	374.8	1 025.2	727.4	270.0	207.2
Musk 1	111.2	**84.3**	111.7	304.4	168.2	84.9	102.1
Musk 2	**1 046.0**	1 333.7	2 116.5	5 849.7	6 523.6	1 477.3	5 698.0
LSVT	**65.1**	149.1	273.0	1 110.8	334.5	196.4	113.8
Flowmeters A	**20.0**	59.2	79.3	107.1	146.2	62.0	67.0
Flowmeters B	**25.6**	65.3	94.3	138.2	221.5	68.2	74.0
Flowmeters C	**65.8**	71.9	105.1	139.4	177.4	77.2	79.0
Flowmeters D	23.6	69.1	93.3	123.6	157.5	77.1	78.4
ParkinsonDatabase	**40.8**	70.0	111.7	143.6	149.9	83.2	69.8

表 6　DBOG 与 5 种对比算法在特征维度上的对比

数据集	DBOG	DBO	ISSA	TMGWO	OBWOA	MGFPA	MSADBO
absentism	1.2	**1.0**	3.7	**1.0**	**1.0**	1.4	1.5
ChurnData	8.5	7.4	10.2	12.5	8.7	8.7	**6.2**
Darwin	**27.0**	62.0	190.4	67.4	37.8	213.1	58
Toxicity 2	**22.6**	67.1	520.9	33.1	27.3	586.1	44.5
Musk 1	**36.8**	**52.0**	80.4	57.0	51.6	79.8	47.7
Musk 2	**16.7**	22.2	75.1	20.8	18.2	67.8	18.2
LSVT	**5.9**	8.3	101.4	16.4	6.8	111.9	12.5
Flowmeters A	**2.7**	3.6	2.9	2.9	3.3	8.0	4
Flowmeters B	**2.0**	2.4	8.6	**2.0**	2.3	**2.0**	2.5
Flowmeters C	**5.7**	8.6	12.5	9.5	12.7	10.4	11.8
Flowmeters D	**3.5**	6.8	9.6	5.9	7.4	7.4	5.1
ParkinsonDatabase	10.0	10.7	20.2	**8.5**	10.3	21.0	13.1



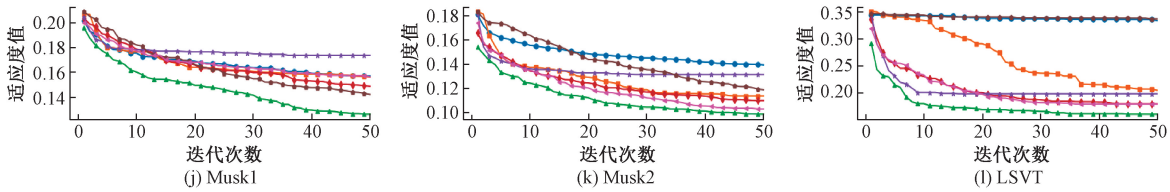


图 4　DBOG 算法与其他算法的收敛图对比

收敛图可以反映算法的初始化效果、收敛速度、最终收敛效果。精英初始化策略在 12 个数据集中的 11 个中能够让算法以更高的起点开始运行,算法在收敛初期就找到更好的位置,为后续寻找最优解奠定基础。而在收敛速度以及最终收敛效果指标上,DBOG 算法在 11 个数据集上表现优于其他算法,仅在 Flowmeters B 数据集中与除 ISSA 以外的其他算法表现相同。除此之外,我们还发现 DBOG 算法在高维数据集(Darwin 450 个特征、Toxicity 2 1 203 个特征、Musk 1 163 个特征、Musk 2 163 个特征、LVST 313 个特征)中表现优于在低维数据集中的表现,且在高维数据集中与其他算法的差距要比低维数据集中的大,证明 DBOG 算法在特征选择任务尤其是高维数据集上具有优越性。

MSADBO、TMGWO、OBWOA、MGFPA 算法的改进虽然采用了优化种群初始化、改进算法学习策略等方法,但依靠初始化改进、变异、反向学习等方式去更新个体解存在一定程度的随机性,尤其是面对特征选择任务时,一个很差或很好个体解的反向解、扰动解并不一定会优于当前种群中的最优解,从而陷于局部最优导致算法无法收敛到更优解。其中 MSADBO 同样是采用引入其他算法进行融合的策略,并且对种群初始化方式进行了改进。但 DBOG 算法优于 MSADBO 和其他改进型算法之处在于,面向特征选择任务的针对性改进。即针对蜣螂的滚球和跳舞行为公式在特征选择任务上表现不佳的问题进行有效算法的融合替换,更充分的利用种群中每个个体的计算资源,从而达到优于其他算法的效果。

3　结　　论

本文使用二进制策略使蜣螂优化算法能够运行特征选择任务,针对原始蜣螂优化算法存在的问题,使用三种优化策略:精英初始化种群策略、蜣螂优化算法与灰狼优化算法融合策略、运行加速策略。首先使用精英初始化策略优化算法的初始种群。其次针对原始蜣螂优化算法中部分行为在特征选择任务中表现不佳的问题,引入灰狼优化算法行为进行替代,有效改善了种群的寻优性能。最后,对于算法种群中存在相同解的问题提出运行加速策略。实验数据显示,相比原本的蜣螂优化算法以及同类特征选择算法,蜣螂灰狼融合优化算法 DBOG 能够在更短的时间内收敛到更优秀的适应度值,以更少的特征维度获得更高的特征子集,证明了 DBOG 算法的有效性。

但 DBOG 算法还存在由精英初始化策略带来的算法时间复杂度稍高的问题,未来可以考虑采用轻量化初始化策略进一步优化。算法中一些固定参数也可以尝试使用自适应策略进行调整,或将进一步提高 DBOG 算法的性能。在未来的研究中,可以考虑将 DBOG 算法应用于更多的公共数据集,或者将其应用于优化支持向量机或神经网络的参数等研究。

参考文献

[1] DOKEROGLU T, DENIZ A, KIZILOZ H E. A comprehensive survey on recent metaheuristics for feature selection[J]. Neurocomputing, 2022, 494: 269-296.

[2] 常梦容,王海瑞,肖杨. mRMR 特征筛选和随机森林的故障诊断方法研究[J]. 电子测量与仪器学报,2022,36(3):175-183.

[3] DEHKORDI A A, SADIQ A S, MIRJALILI S, et al. Nonlinear-based chaotic harris hawks optimizer: algorithm and internet of vehicles application[J]. Applied Soft Computing, 2021,109: 107574.

[4] 方志,余粟. 基于 IGA-Optuna-LightGBM 的民航潜在旅客预测[J]. 国外电子测量技术,2022,41(10):142-147.

[5] HUDA R K, BANKA H. New efficient initialization and updating mechanisms in PSO for feature selection and classification [J]. Neural Computing and Applications, 2020, 32: 3283-3294.

[6] ABDEL-BASSET M, DING W, EL-SHAHAT D. A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection[J]. Artificial Intelligence Review, 2021, 54: 593-637.

[7] EMARY E, ZAWBAA H M, HASSANIEN A E. Binary grey wolf optimization approaches for feature selection[J]. Neurocomputing, 2016, 172: 371-381.

[8] FARIS H, HEIDARI A A, ALA M A Z, et al. Time-varying hierarchical chains of salps with random weight networks for feature selection [J]. Expert Systems with Applications, 2020, 140: 112898.

[9] 李郅琴,杜建强,聂斌,等. 基于黑寡妇算法的特征选择方法研究[J]. 计算机工程与应用,2022,58(16):147-156.

[10] 李鹏,陈守静,杨山山,等. 基于 Logistic 映射的果蝇算

法优化 Otsu 图像分割方法[J]. 国外电子测量技术，2022，41(7)：9-17.

[11] 董奕含，喻志超，胡天跃，等. 基于改进蜣螂优化算法的瑞雷波频散曲线反演方法[J]. 油气地质与采收率，2023，30(4)：86-97.

[12] JIAN X，WENG Z. A logistic chaotic JAYA algorithm for parameters identification of photovoltaic cell and module models [J]. Optik，2020，203：164041.

[13] AL-QANESS M A A，EWEES A A，FAN H，et al. Modified aquila optimizer for forecasting oil production[J]. Geo-Spatial Information Science，2022，25(4)：519-535.

[14] EWEES A A，ABD ELAZIZ M，HOUSSEIN E H. Improved grasshopper optimization algorithm using opposition-based learning[J]. Expert Systems with Applications，2018，112：156-172.

[15] LUO J，CHEN H，XU Y，et al. An improved grasshopper optimization algorithm with application to financial stress prediction[J]. Applied Mathematical Modelling，2018，64：654-668.

[16] 张仪，冯伟，王卫军，等. 融合 LSTM 和 PPO 算法的移动机器人视觉导航[J]. 电子测量与仪器学报，2022，36(8)：132-140.

[17] MAHAJAN S，ABUALIGAH L，PANDIT A K，et al. hybrid aquila optimizer with arithmetic optimization algorithm for global optimization tasks [J]. Soft Computing，2022，26(10)：4863-4881.

[18] MAFARJA M M，MIRJALILI S. Hybrid whale optimization algorithm with simulated annealing for feature selection[J]. Neurocomputing，2017，260：302-312.

[19] XUE J，SHEN B. Dung beetle optimizer：A new meta-heuristic algorithm for global optimization[J]. The Journal of Supercomputing，2023，79(7)：7305-7336.

[20] LI R，LIU Q，LIU L. Novel image encryption algorithm based on improved logistic map[J]. IET Image Processing，2019，13(1)：125-134.

[21] MIRJALILI S，MIRJALILI S M，LEWIS A. Grey wolf optimizer[J]. Advances in engineering software，2014，69：46-61.

[22] ABDEL-BASSET M，EL-SHAHAT D，EL-HENAWY I，et al. A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection [J]. Expert Systems with Applications，2020，139：112824.

[23] TUBISHAT M，IDRIS N，SHUIB L，et al. Improved salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection[J]. Expert Systems with Applications，2020，145：113122.

[24] ABD ELAZIZ M，OLIVA D. Parameter estimation of solar cells diode models by an improved opposition-based whale optimization algorithm [J]. Energy conversion and management，2018，171：1843-1859.

[25] SHAMBOUR M K Y，ABUSNAINA A A，ALSALIBI A I. Modified global flower pollination algorithm and its application for optimization problems [J]. Interdisciplinary Sciences：Computational Life Sciences，2019，11：496-507.

[26] 潘劲成，李少波，周鹏，等. 改进正弦算法引导的蜣螂优化算法[J]. 计算机工程与应用，2023，59(22)：92-110.

作者简介

李珺，博士，教授，博士生导师，主要研究方向为数据挖掘、机器学习等。

E-mail：lijun2010@nefu.edu.can

徐秦，硕士研究生，主要研究方向为群智能算法、机器学习等。

E-mail：1041439369@qq.com