

融合差分进化与分散觅食策略的黏菌算法并应用在高维特征选择上

袁李佩¹, 匡芳君², 郑媛萍¹

¹中国温州大学计算机与人工智能学院, 浙江 温州

²中国温州商学院信息工程学院, 浙江 温州

收稿日期: 2026年3月20日; 录用日期: 2026年5月21日; 发布日期: 2026年5月29日

摘要

在基因数据这类高维小样本数据的分类任务中, 特征选择已成为提升模型泛化能力与可解释性的关键预处理环节。黏菌算法作为一类新型群智能优化方法, 具备结构简洁、易于实现的优点。但在处理复杂优化问题时面临探索与开发能力失衡、易陷入局部最优等局限。为此, 本文提出一种融合差分进化机制与分散觅食策略的增强型黏菌算法(DBSMA)。该算法通过引入差分进化的变异与交叉操作增强种群多样性, 并设计自适应分散觅食策略以扩大全局搜索范围, 从而实现算法在全局探索与局部开发之间的有效平衡。在CEC2017标准测试集上的对比实验表明, DBSMA在求解精度、收敛速度与鲁棒性方面均优于12种主流算法与16种改进算法。本文还构建其二进制版本bDBSMA并应用于9个高维医学数据集的特征选择任务, 实验结果显示, bDBSMA在显著降低所选特征数量的同时, 还能保持较高的分类精度, 验证了bDBSMA在高维特征选择任务中的有效性与实用价值。

关键词

黏菌算法, 差分进化, 分散觅食, 特征选择

Slime Mould Algorithm Integrating Differential Evolution and Dispersive Foraging Strategy for High-Dimensional Feature Selection

Lipei Yuan¹, Fangjun Kuang², Yuanping Zheng¹

¹College of Computer and Artificial Intelligence, Wenzhou University, Wenzhou Zhejiang

²School of Information Engineering, Wenzhou Business College, Wenzhou Zhejiang

Abstract

In classification tasks involving high-dimensional, small-sample data such as genomic data, feature selection has become a critical preprocessing step for improving model generalization and interpretability. As a novel swarm intelligence method, the slime mould algorithm is characterized by its simple structure and ease of implementation. However, when addressing complex optimization problems, it faces limitations such as imbalance between exploration and exploitation, as well as a tendency to become trapped in local optima. To address these issues, this paper proposes an enhanced slime mould algorithm, named DBSMA, which integrates a differential evolution mechanism with a dispersed foraging strategy. The algorithm enhances population diversity by incorporating the mutation and crossover operations of differential evolution, and designs an adaptive dispersed foraging strategy to expand the global search range, thereby achieving an effective balance between global exploration and local exploitation. Comparative experiments on the CEC2017 benchmark test suite demonstrate that DBSMA outperforms 12 mainstream algorithms and 16 improved algorithms in terms of solution accuracy, convergence speed, and robustness. Furthermore, a binary version, bDBSMA, is constructed and applied to feature selection tasks on nine high-dimensional medical datasets. Experimental results show that bDBSMA significantly reduces the number of selected features while maintaining high classification accuracy, verifying its effectiveness and practical value in high-dimensional feature selection tasks.

Keywords

Slime Mould Algorithm, Differential Evolution, Dispersed Foraging, Feature Selection

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

近年来,基因组学、医学成像等领域的的数据呈现指数级增长,高维小样本问题成为机器学习的主要挑战之一[1]。这类数据包含大量不相关、冗余特征,不仅增加计算复杂度,还易导致过拟合,降低模型泛化能力[2]。特征选择作为关键的预处理手段,旨在筛选出信息丰富且精简的特征子集,在提升分类性能、增强模型可解释性方面具有重要作用[3]。

特征选择方法可分为过滤型、封装型、嵌入型和融合型[4]。其中,封装型方法以分类性能为评价指标,虽精度较高但计算成本大。由于特征选择本质上是 NP 难问题,元启发式算法因其不依赖梯度信息、能有效跳出局部最优而受到广泛关注[5]。元启发式典型算法包括遗传算法[6]、粒子群优化[7]、灰狼优化器[8]、鲸鱼优化算法[9]等。

黏菌算法(Slime Mould Algorithm, SMA)是 2020 年提出的一种新型群智能优化方法,通过模拟黏菌觅食行为中的正负反馈机制实现优化搜索[10]。SMA 算法结构简单,已经在不少领域得到应用[11][12]。不过在处理复杂优化问题时,该算法仍存在收敛速度慢、易陷入局部最优等不足[13]。现有的改进策略包括混合其他算法、引入变异策略、优化初始化等[14][15],但仍难以在全局探索与局部开发之间取得理想平衡。

为此, 本文提出一种融合差分进化机制与分散觅食策略的增强型黏菌算法 DBSMA。其中差分进化 (Differential Evolution, DE) [16] 的变异操作可以利用随机个体差异生成新解, 能在不依赖最优个体的前提下保持种群多样性。当 SMA 进入停滞或早熟收敛时, DE 的差分向量可有效扰动种群, 跳出局部极值。两者的结合本质上是差分进化机制探索与黏菌算法本身开发的异质互补。分散觅食机制[17]隐含了基于适应度的选择性扰动: 较好的解被局部精化, 较差的解被主动放弃或替换。这种策略嵌入 SMA 后, 可以动态平衡全局探索与局部搜索的比例——避免 DE 的过度随机破坏已找到的优良区域, 同时防止 SMA 过度聚集于某个子空间。另外, 本文还构建了二进制版本 bDBSMA 用于高维特征选择, 并在高维复杂医学数据集中取得了不错的结果。总结本文主要工作如下:

- (1) 引入差分进化机制增强种群多样性, 帮助算法跳出局部最优;
- (2) 增加分散觅食策略, 分配部分个体探索未知区域, 提升全局搜索能力;
- (3) 在 CEC 2017 测试集上与 28 种算法对比, 验证 DBSMA 的优越性;
- (4) 将 bDBSMA 应用于 9 个高维医学数据集, 验证其在特征选择中的有效性。

2. 黏菌算法

黏菌算法 SMA 是由李世民等人在 2020 年提出的一种元启发式优化方法[10]。他们通过对黏菌觅食时正负反馈交互作用的模拟, 凝练出三种具有不同收缩特征的觅食行为模式: 靠近食物、包裹食物以及自适应振荡寻找食物。

(1) 靠近食物

黏菌根据食物浓度调整位置, 更新公式为:

$$\bar{X}(t+1) = \begin{cases} \bar{X}_b(t) + \bar{vb} * (\bar{W} * \bar{X}_A(t) - \bar{X}_B(t)), & r < p \\ \bar{vc} * \bar{X}(t), & r \geq p \end{cases} \quad (\text{公式 2.1})$$

其中 \bar{X}_b 为当前最优位置, \bar{vb} 和 \bar{vc} 为振荡参数, \bar{W} 为权重系数。

参数 p 控制更新策略的选择, 计算方程如下:

$$p = \tanh |S(i) - DF| \quad (\text{公式 2.2})$$

(2) 包裹食物

通过权重 \bar{W} 模拟黏菌静脉粗细变化, \bar{W} 表达式如下所示:

$$\overline{W(SmellIndex(l))} = \begin{cases} 1 + r * \log \left(\frac{bF - S(i)}{bF - wF} + 1 \right), & \text{condition} \\ 1 - r * \log \left(\frac{bF - S(i)}{bF - wF} + 1 \right), & \text{other} \end{cases} \quad (\text{公式 2.3})$$

bF 和 wF 分别为当前迭代中最优与最差适应度值。condition 表示取排名位于前一半的个体。

(3) 自适应振荡

参数 \bar{vb} 和 \bar{vc} 随迭代次数自适应调整, 前期增强探索, 后期促进收敛。

参数 \bar{vb} 的方程如下所示:

$$\bar{vb} = [-a, a] \quad (\text{公式 2.4})$$

其中 a 的值计算如下:

$$a = \operatorname{arctanh}\left(-\left(\frac{t}{\operatorname{Max}_t}\right)+1\right) \quad (\text{公式 2.5})$$

公式中 t 表示算法当前迭代次数, Max_t 表示算法最大的迭代次数。

参数 \overline{vc} 的计算公式如下:

$$\overline{vc} = 1 - \left(\frac{t}{\operatorname{Max}_t}\right) \quad (\text{公式 2.6})$$

基于上述原理, 黏菌种群个体的位置更新数学公式如下:

$$\bar{X}(t+1) = \begin{cases} \operatorname{rand} * (UB - LB) + LB, & \operatorname{rand} < z \\ \bar{X}_b(t) + \overline{vb} * (\bar{W} * \bar{X}_A(t) - \bar{X}_B(t)), & \operatorname{rand} \geq z, r < p \\ \overline{vc} * \bar{X}(t), & \operatorname{rand} \geq z, r \geq p \end{cases} \quad (\text{公式 2.7})$$

其中 UB 和 LB 表示搜索空间的上下界, rand 和 r 表示 0 到 1 之间的随机值。

3. 提出的增强型黏菌算法 DBSMA

虽然原始 SMA 的性能优于大多数群智能算法, 但原始 SMA 算法更新最优值附近的位置时, 容易陷入局部最优值。本文在原始 SMA 的基础上引入差分进化机制和分散觅食策略, 以平衡 SMA 的探索和利用能力, 帮助 SMA 跳出局部最优。

3.1. 差分进化机制

为增强种群多样性, 引入差分进化(DE)的变异与交叉操作[16]。对于每个个体, 随机选择三个不同个体 X_{r_1} 、 X_{r_2} 和 X_{r_3} 生成变异向量:

$$U_i = X_{r_1} + F * (X_{r_2} - X_{r_3}) \quad (\text{公式 3.1})$$

式中, F 代表 DE 的变异算子, 在本研究中, 突变算子设定为 $F = 0.4$ 。

随后借助交叉操作来构建试验向量 V_i :

$$V_{ij} = \begin{cases} U_{ij}, & \operatorname{rand} < P_{CR} \text{ or } j = j_{\operatorname{rand}} \\ X_{ij}, & \text{Otherwise} \end{cases} \quad (\text{公式 3.2})$$

其中, P_{CR} 是 DE 的交叉算子。本文中取 $P_{CR} = 0.1$ 。

最后通过选择操作保留优良候选解:

$$X_i = \begin{cases} V_i, & f(V_i) < f(X_i) \\ X_i, & \text{Otherwise} \end{cases} \quad (\text{公式 3.3})$$

3.2. 分散觅食策略

受人工蜂群算法(ABC)的启发, 设定部分个体为“觅食黏菌”, 负责探索未知区域[17]。分散率 DR 随迭代自适应递减:

$$DR = DR_{\max} - (DR_{\max} - DR_{\min}) * \frac{FES}{\operatorname{Max}_FES} \quad (\text{公式 3.4})$$

式中 DR_{\max} 和 DR_{\min} 都是介于 $[0,1]$ 之间的值, 本文设定 $DR_{\max} = 0.4$, $DR_{\min} = 0$ 。

当随机数 $\operatorname{rand} > DR$ 时, 个体执行分散觅食更新:

$$\bar{X}(t+1) = \begin{cases} \bar{X}(t) + \alpha * (\bar{X}_m(t) - \bar{X}_n(t)), & \text{if } rand > DR \\ \bar{X}(t), & \text{otherwise} \end{cases} \quad (\text{公式 3.5})$$

其中 α 是一个比例控制因子, 服从 $\alpha \sim N(0.5, 0.1^2)$ 分布。 $\bar{X}_m(t)$ 和 $\bar{X}_n(t)$ 是随机选择的个体。

3.3. 增强型黏菌算法 DBSMA

原始 SMA 的权重参数 W 能够使算法在保证快速收敛的同时保持一定的扰动率。振荡参数 vb 通过特定的方式缩小黏菌的个体位置, 保证前期勘探的有效性和后续开发的准确性。位置更新策略参数 p 和 3 种不同的位置更新保证了 SMA 在不同搜索阶段具有更好的自适应性。但是在 SMA 的迭代过程中, 随着迭代次数的增加, 黏菌菌群会收敛到当前最优位置, 这就导致 SMA 在求解具有多个局部最优的函数时极易陷入局部最优。为此, 本文中加入了差分进化机制, 利用机制中的突变过程, 增加黏菌群具有跳出局部最优陷阱的可能性。此外, 参数 z 作为控制随机初始化黏菌个体数量的常数, 其取值通常极小。这一设定使得随机分布的个体在种群总数中所占比重微乎其微, 从而导致算法一旦陷入局部最优, 便难以借助随机个体实现有效逃逸, 缺乏相应的跳出机制。因此, 本文中引入分散觅食策略, 目的是分出一部分黏菌进行分散觅食, 扩大了种群搜索范围, 增加算法的多样性。同时本文利用贪婪算法的思想, 不断更新保留最优个体的信息, 使得种群不断向更佳的食物源靠近, 提升了算法的收敛性。最终本文提出融合了差分进化机制和分散觅食策略所形成的 DBSMA, 算法 1 描述了该算法的伪代码, 而图 1 给出了算法的流程图。

算法 1 DBSMA 的伪代码

初始化参数 popsize, Max_FEs;

初始化黏菌种群 X_i ($i = 1, 2, 3, \dots, n$);

初始化控制参数 z, a ;

计算黏菌的适应度值;

按适应度值升序排序;

更新最佳适应度值 bestFitness 和最佳位置 X_b ;

While ($t \leq \text{Max_FEs}$)

For $i = 1$ to n

随机选择个体 X_{r1}, X_{r2}, X_{r3} ;

根据公式 3.1 更新 U_i ;

根据公式 3.2 更新 V_i ;

根据公式 3.3 更新 X_i ;

更新最佳适应度值 bestFitness 和最佳位置 X_b ;

End

根据公式 2.6 计算权重 W ;

For $i = 1$ to popsize

根据公式 2.2 更新参数 p ;

根据公式 2.4 和公式 2.5 更新 vb ;

根据公式 2.6 更新 vc ;

续表

根据公式 2.7 更新黏菌种群位置;
更新最佳适应度值 $bestFitness$ 和最佳位置 X_b ;

End

根据公式 3.4 和公式 3.5 更新觅食黏菌;
计算适应度值并采用贪婪策略更新 X ;

更新最佳适应度值 $bestFitness$ 和最佳位置 X_b ;

$t = t + 1$;

End While

Return 最佳适应度值 $bestFitness$ 和最佳位置 X_b ;

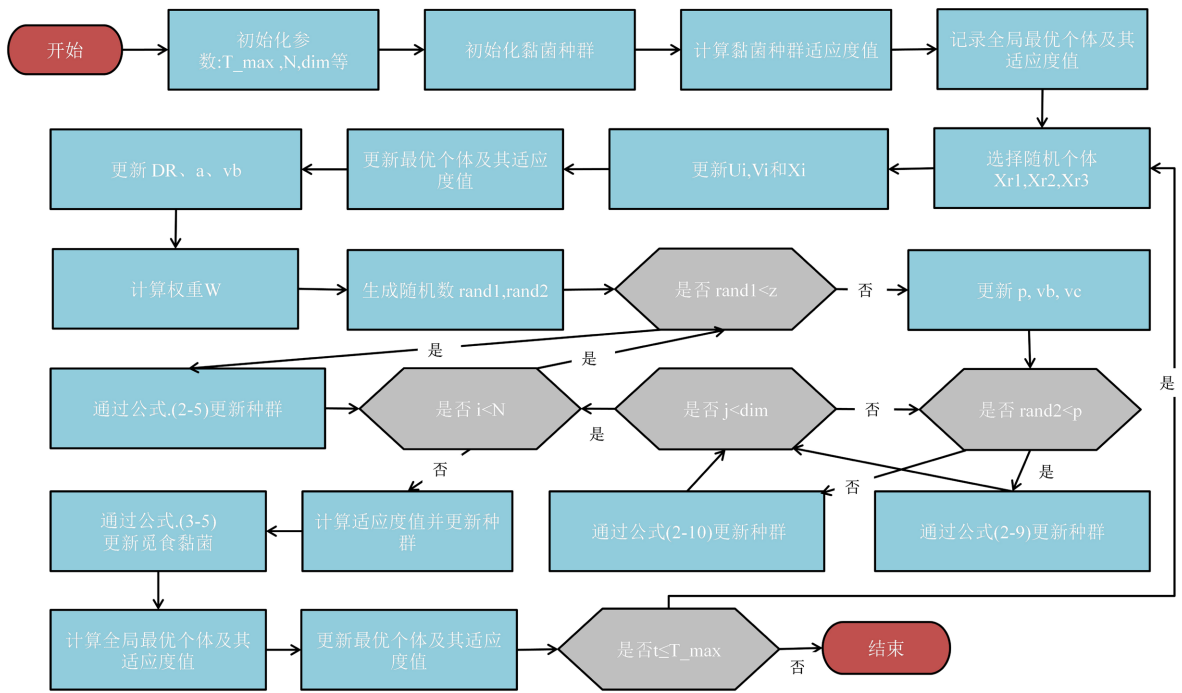


Figure 1. Flowchart of DBSMA
图 1. DBSMA 流程图

4. 实验结果与分析

为评估 DBSMA 的优化性能, 本研究选取 CEC2017 基准测试集进行实验, 包含单峰、多峰、混合与组合四类函数。实验环境为 Windows 10, 12thGenInter(R)Core(TM)i5-125003.00GHz, 16 GB RAM, MATLAB R2021b。所有算法种群规模设为 30, 最大评价次数为 300,000, 独立运行 30 次。

4.1. 参数敏感性分析

为评估关键参数 F 、 P_{CR} 、 DR_{max} 对算法性能的影响, 本文在标准测试函数上开展单参数敏感性实验。各参数独立变化, 其余参数固定。

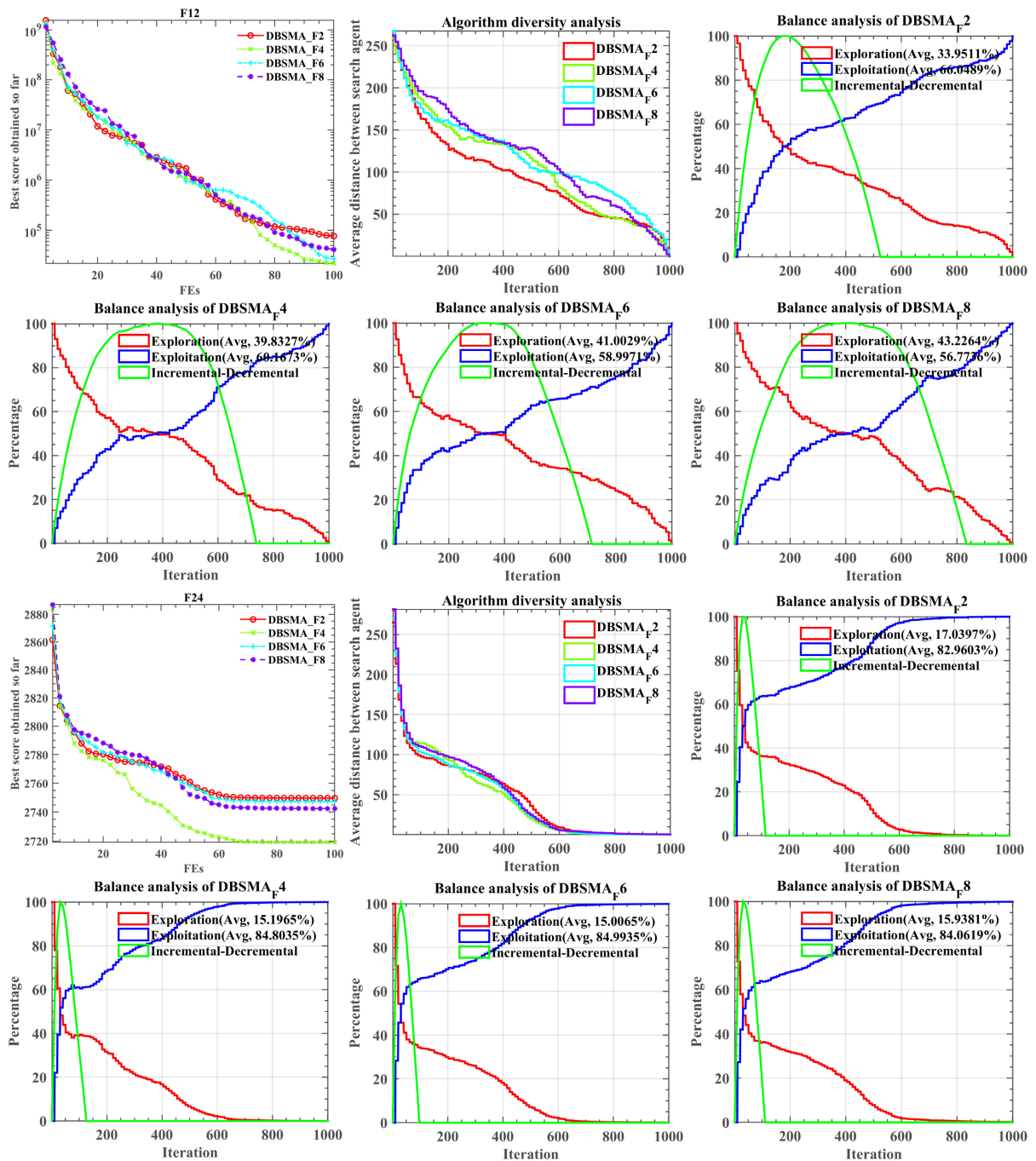


Figure 2. Sensitivity analysis results of parameter F

图 2. 参数 F 敏感性实验结果

本文对参数 F 取值 0.2、0.4、0.6、0.8。由于篇幅问题，本文选取部分函数结果进行展示，具体结果由图 2 可知。一般情况下，随着参数 F 的取值增大，算法的多样性增强，这是因为参数 F 是差分进化机制的突变因子， F 越大，种群个体突变的概率就会越高，从而增加算法的多样性。同时对于不同的函数，同一算法探索和开发的平衡点并不同，这正说明对于具体的问题需要选取不同的参数。由于医学数据诊断往往需要高精度，因此根据迭代曲线，本文选取 $F = 0.4$ 。

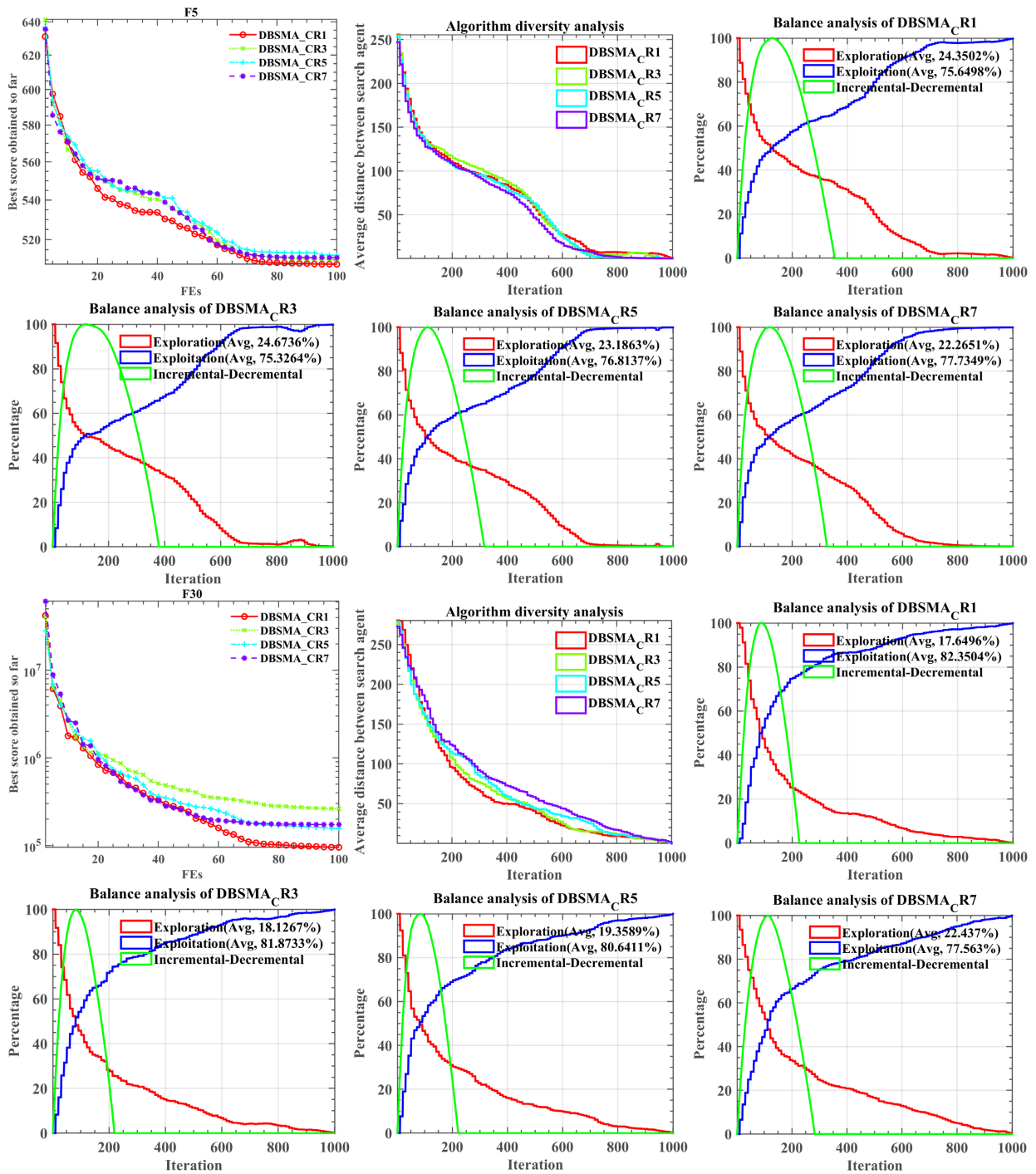


Figure 3. Sensitivity analysis results of parameter P_{CR}

图 3. 参数 P_{CR} 敏感性实验结果

本文对参数 P_{CR} 取值 0.1、0.3、0.5、0.7。由图 3 可知，在函数 F5 中，参数 P_{CR} 对算法多样性的影响不大，且随着 P_{CR} 值越大，算法会越来越注重开发阶段。但在复杂的混合函数 F30 中，参数 P_{CR} 显著影响算法的多样性，并影响算法探索阶段。由于高维医学特征选择是复杂且要求高精度的问题，因此本文选取 $P_{CR} = 0.1$ 。

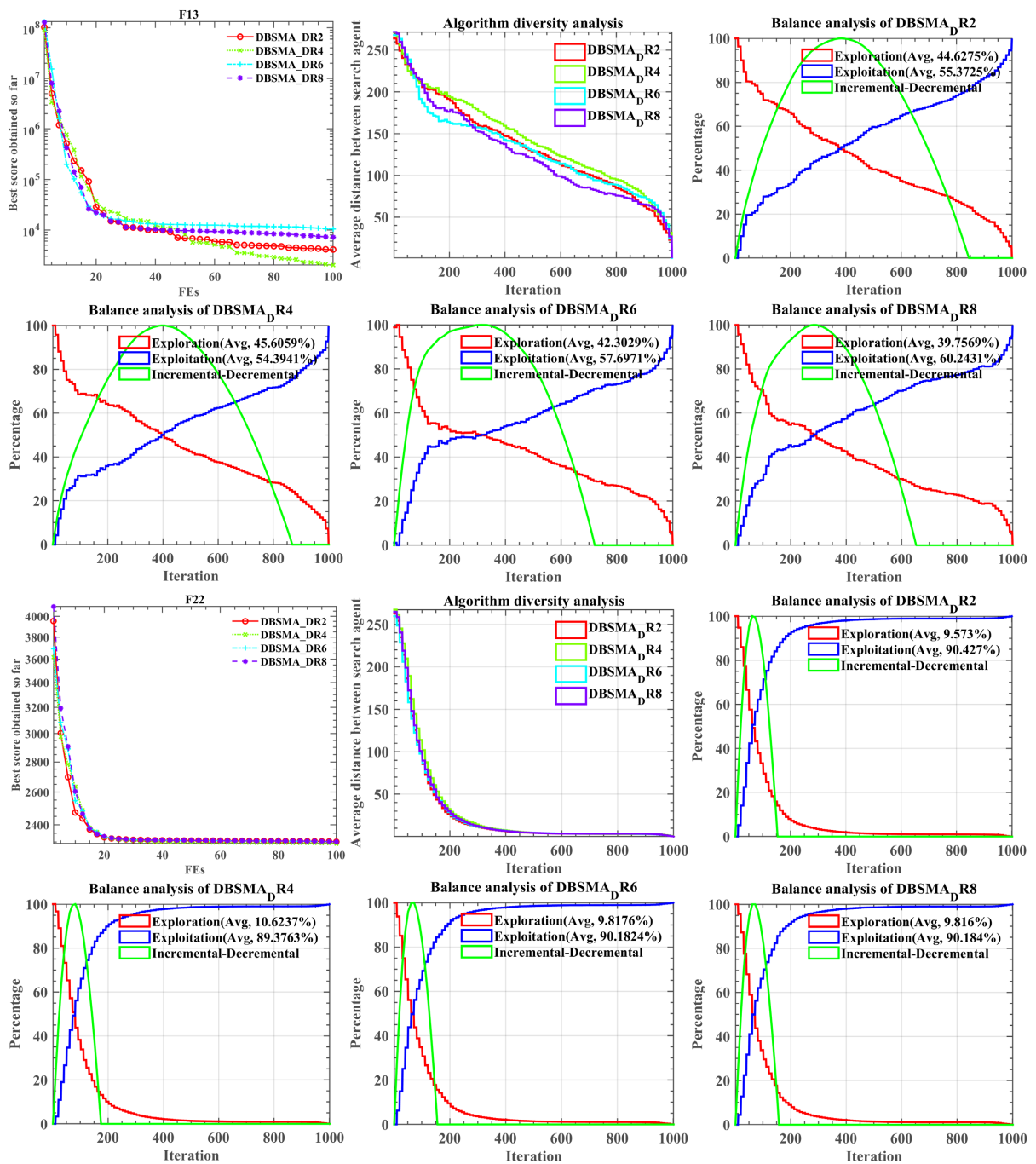


Figure 4. Sensitivity analysis results of parameter DR_{max}

图 4. 参数 DR_{max} 敏感性实验结果

本文对参数 DR_{max} 取值 0.2、0.4、0.6、0.8。从图 4 结果可以看出，在组合函数 F13 中，参数 DR_{max} 对多样性和探索开发平衡性影响显著，过高和过低的 DR 都会导致多样性变低，算法偏向开发，可以从图 4 中得知，参数 $DR_{max} = 0.4$ 时在函数 F13 上表现最好。而在混合函数 F22，参数 DR 的影响较小，但当 $DR_{max} = 0.4$ ，算法的多样性有较小的提升且有较大的概率进行探索。这为算法找到更优解提供了更大的概率，因此本文取 $DR_{max} = 0.4$ 。

需要指出的是,上述单参数最优值是在固定其他参数下获得的,实际中参数之间存在耦合。本文主要选取了在复杂函数中取得精度更高的参数,这对高维医学特征选择任务尤为主要。本文给出的参数配置在后续完整实验中持续取得优于对比算法的结果,证明了其有效性。对于未来使用本算法的用户,建议在计算资源允许时进行任务特定的参数调优。若追求通用配置,可采用本文推荐的组合作为起点。

4.2. 与先进原始算法的比较

将 DBSMA 与 SMA、DE、ABC、PSO 等 12 种主流算法对比,结果如图 5 所示。DBSMA 在总体排名中位列第一。表 1 Wilcoxon 符号秩检验结果表明, DBSMA 在 22 个函数上显著优于 SMA,在 28 个函数上优于 DE,验证了其优越性。

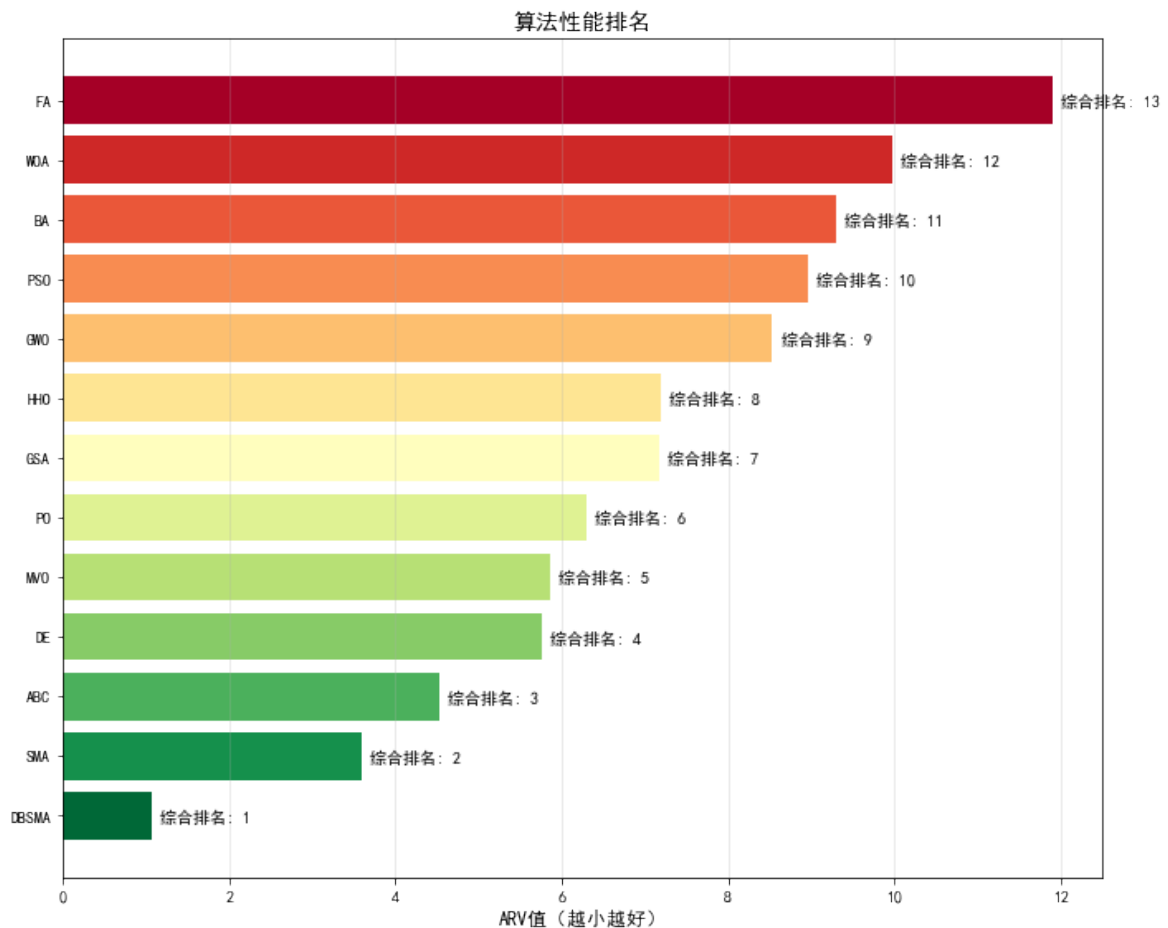


Figure 5. Friedman test rank comparison of DBSMA with 12 basic optimization algorithms

图 5. DBSMA 与 12 种基本优化算法的 Friedman 检验排名图

为了进一步展示 DBSMA 的性能,在图 6 中描述了算法在 CEC2017 上的迭代图。图 6 中 DBSMA 在单峰函数 F1 和 F2 中均可以找到更好的最优解,这是由于差分进化机制和分散觅食策略的充分探索和利用。分散觅食策略有助于加强对全局最优解的探索,同时将差分进化机制和原始 SMA 的开发能力进一步强化。F5、F7、F8 和 F10 是多模态函数的代表,用于检测函数跳出局部最优值的能力。由图 6 可以明显看出,与其他基本函数相比, DBSMA 可以更好地跳出局部最优并找到全局最优解。这是由于差分进化机制中的突变和分散觅食策略的分散搜索,使得大部分黏菌在向局部最优解迭代靠近的同时,有一部分黏

菌可以跳出局部最优陷阱，引导种群趋近于全局最优解。F12、F16、F18 和 F20 表示混合函数，F22 和 F27 表示组合函数。可以看出，即使面对更加复杂的情况，DBSMA 也可以在一定程度上跳出局部点，找到更好的位置。这表明，通过两种机制的结合，DBSMA 很好的平衡了勘探和开发两个过程。

Table 1. Wilcoxon signed-rank test results of DBSMA and 12 basic optimization algorithms

表 1. DBSMA 与 12 种基本优化算法的 Wilcoxon 符号秩检验结果

DBSMA 与 12 种基本优化算法的 Wilcoxon 检验结果							
+/-/-	DBSMA	SMA	DE	ABC	PSO	FA	MVO
	~	22/0/8	28/0/2	28/1/1	30/0/0	30/0/0	30/0/0
+/-/-	BA	GSA	HHO	WOA	GWO	PO	
	29/0/1	30/0/0	22/0/8	27/0/3	30/0/0	30/0/0	

注：“+”表示 DBSMA 明显优于对比算法，“=”表示两者无显著差异，“-”表示 DBSMA 明显劣于对比算法。

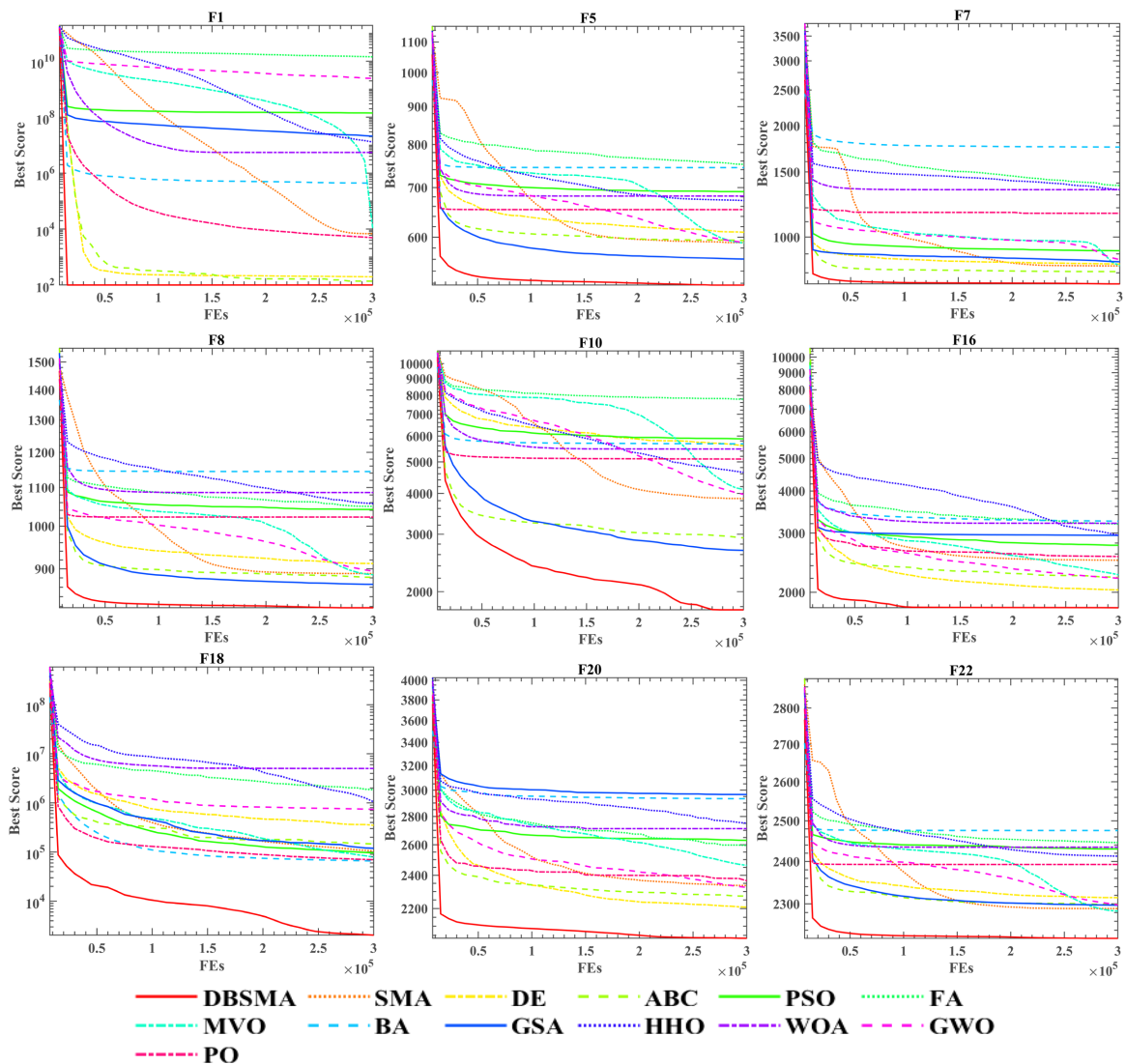


Figure 6. Convergence graph of DBSMA and 12 basic optimization algorithms

图 6. DBSMA 与 12 种基本优化算法的收敛图

4.3. 与优秀改进算法的比较

为了进一步证明 DBSMA 在全局搜索中的卓越性能, 本节将 DBSMA 与 16 种改进算法(如 CEC 冠军算法 EBOwithCMA、优秀增强体 ALCPSO 等)对比, 结果如图 7 所示, DBSMA 总体排名第一。表 2 Wilcoxon 符号秩检验结果表明, DBSMA 在 27 个函数上显著优于 CEC 冠军算法 EBOwithCMA, 验证了其优越性。

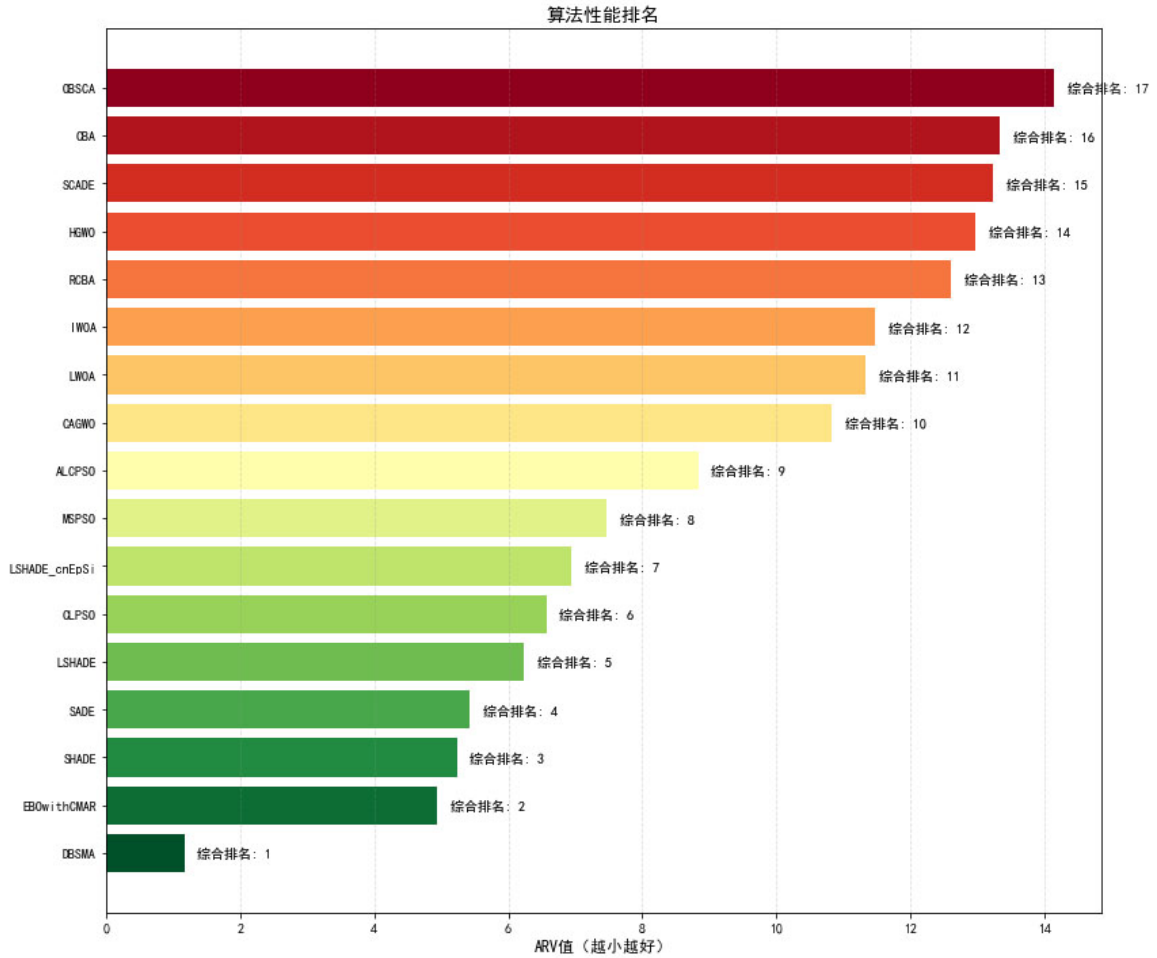


Figure 7. Friedman test ranking diagram of DBSMA and 16 improved optimization algorithms
图 7. DBSMA 与 16 种改进优化算法的 Freidman 检验排名图

Table 2. Wilcoxon signed-rank test results of DBSMA and 16 improved optimization algorithms
表 2. DBSMA 与 16 种改进优化算法的 Wilcoxon 符号秩检验结果

DBSMA 与 16 种改进优化算法的 Wilcoxon 检验结果						
+ / = / -	DBSMA	EBOwithCMAR	LSHADE_cnEpSi	ALCPSO	CLPSO	LSHADE
	~	27/1/2	29/0/1	30/0/0	29/0/1	29/0/1
+ / = / -	SADE	SHADE	SCADE	RCBA	CBA	LWOA
	27/0/3	28/0/2	25/0/5	30/0/0	30/0/0	30/0/0
+ / = / -	IWOA	HGWO	CAGWO	OBSCA	MSPSO	
	28/0/2	28/0/2	29/0/1	30/0/0	28/2/0	

注：“+”表示 DBSMA 明显优于对比算法，“=”表示两者无显著差异，“-”表示 DBSMA 明显劣于对比算法。

为了进一步进行比较和分析,图8展示了从30个基准函数中选择的一些测试函数的收敛曲线的示例,将DBSMA和其他高级MA的收敛速度和收敛结果进行了可视化。从图8中可以看出,当迭代进行到一定次数后,代表DBSMA的红线始终位于函数收敛图的底部,这表明了DBSMA总能更快找到更优的解,说明DBSMA在寻找最优解上具有优越性。而且,可以从图8中看出,大多数时候,DBSMA比其它算法曲线更早下降到全局最优解附近,在迭代早期就达到了收敛状态。说明DBSMA在保持解的优越性的同时,也具有好的收敛性。因此,根据比较实验可以得出结论,本文所提出的DBSMA算法表现出显著的优化能力和出色的稳定性。

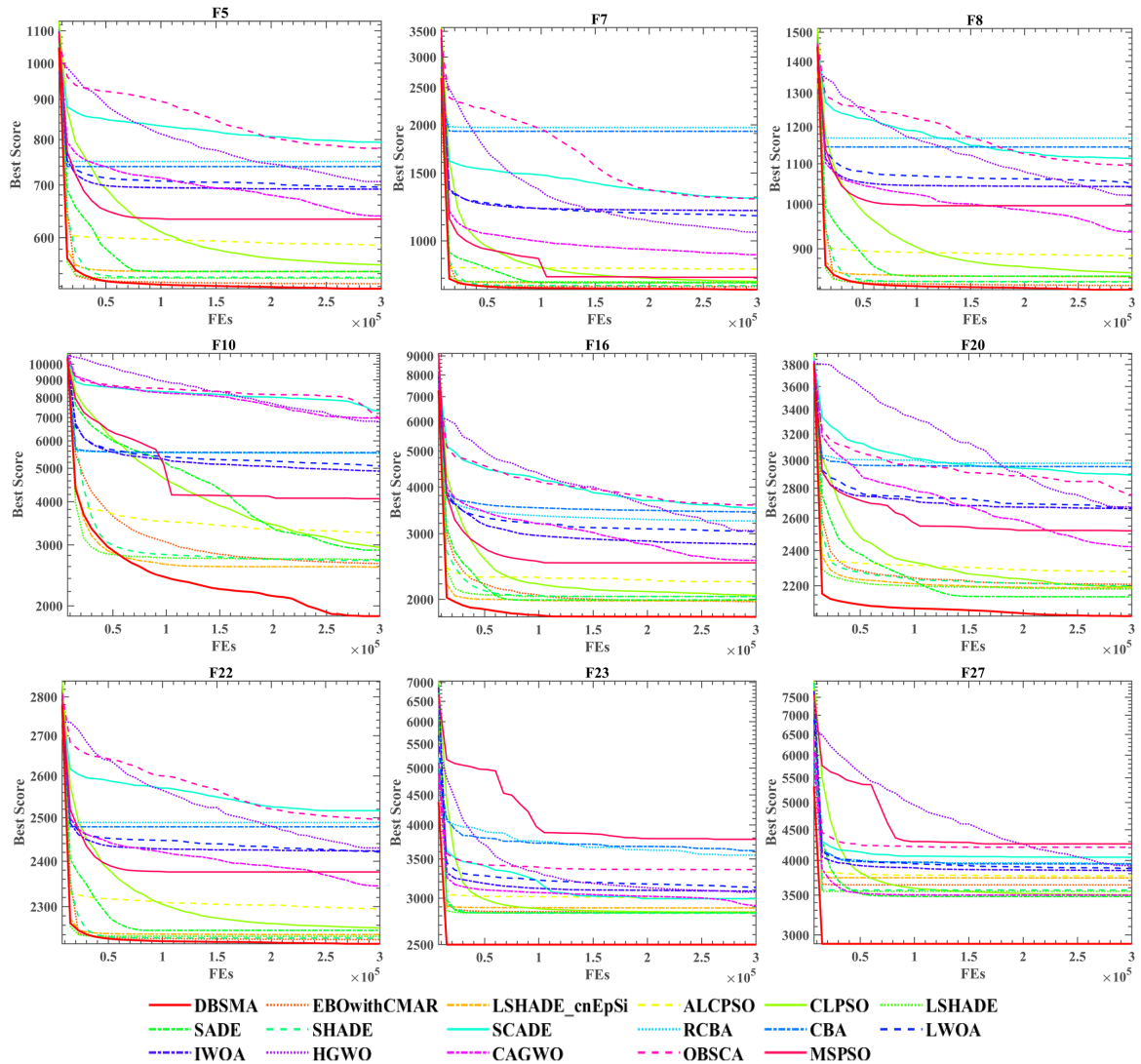


Figure 8. Convergence graph of DBSMA and 16 improved algorithms

图8. DBSMA与16种改进算法的收敛图

5. 特征选择应用

5.1. 二进制 DBSMA

DBSMA本质上是一种适用于连续空间的群体智能算法,而特征选择是一个典型的离散优化任务,

解空间是二进制的。因此，必须将 DBSMA 连续位置信息离散化为“0”和“1”两种状态，才能适用于特征选择这种离散优化问题。

为此，本文采用 V 形传递函数将 DBSMA 映射至二进制离散空间里，得到 bDBSMA：

$$X^{bin} = \begin{cases} 1, & \text{if sigmoid}(X^{count}) \geq rand \\ 0, & \text{otherwise} \end{cases} \quad (\text{公式 5.1})$$

其中， X^{count} 表示代理在连续空间中的位置， X^{bin} 则为转换后的二进制位置。

5.2. 数据集与实验设置

选取 9 个高维医学数据集，数据来源于公开可得的数据库 UCI: <https://archive.ics.uci.edu/datasets>，数据集具体信息如表 3 所示。数据特征维度从 2000 至 15009 不等，样本量介于 60 至 308 之间。

Table 3. Detailed information on high-dimensional medical datasets

表 3. 高维医学数据集详细信息

Dataset	Classes	Samples	Features
Colon	2	62	2000
Leukemia1	2	72	5327
Leukemia2	3	72	11,225
SRBCT	4	83	2308
Lung_Cancer	2	181	12,553
Prostate_Tumor	2	102	10,509
Tumors_9	9	60	5726
Tumors_11	2	174	12,353
Tumors_14	4	308	15,009

本文采用十折交叉验证，KNN 作为分类器，对比算法包括 bSMA、BPSO、bWOA、bMESMA 等 11 种。此外，还使用了四个指标来评估其分类性能，即平均适应度，分类的平均错误率、所选特征的平均数量和平均运行时间。

5.3. 结果和分析

在比较过程中，每种分类方法识别的特征子集被用作 KNN 分类器的输入，分类准确性作为输出指标。表 4 展示了四个评价指标的 Freidman 检验结果。从表 4 的结果可以看出，所提出的 bDBSMA 在平均适应度、平均错误率和平均特征数量三个核心性能指标上均取得最优排名，ARV 分别为 1.22、1.44 和 1.44，显著优于其他对比算法。这一结果充分验证了差分进化机制与分散觅食策略在提升黏菌算法特征选择性能方面的有效性。

在平均适应度指标上，bDBSMA 以 1.22 排名第一，超出第二 2.11，断层居于首位。这个结果表明对比其它二进制算法，bDBSMA 在特征选择综合性能上表现更好，能够同时平衡分类精度和特征降维。

在平均错误率指标上，bDBSMA 以 1.44 位居第一。这说明，在进行高维特征选择中，bDBSMA 能够在众多候选特征子集中，挑选出正确率度更高的子集。

平均特征数量指标可以表现算法的特征降维能力。从表 4 的检验结果可以看出，在这个指标上，bDBSMA 同样位居榜首，且显著优于其它二元优化算法。结合平均错误率指标结果，证明了 bDBSMA 在

保持高分类精度的同时，能够选择出特征数量最少的特征子集。在高维医学数据分析中，这项指标尤为重要。因为较少的特征不仅降低了后续计算开销，还增强了模型的可解释性，有助于识别真正具有诊断价值的医学指标。

从运行时间指标上看，bDBSMA 的 Friedman 检验结果排名第八，表现最优的算法是 bWOA。bWOA 算法简单，步骤少，因此运行时间短。而本文提出的 bDBSMA 引入了差分进化和分散觅食两个策略，增加了额外的计算开销。但是，在实际应用中，特征选择的精度往往比计算效率更重要，尤其是在医疗诊断这类对算法模型准确性要求极高的领域里。因此，相比于 bDBSMA 在特征选择核心指标上的显著优势，计算成本的增加是在可接受范围内的。

图 9 展示了 bDBSMA 在 9 个高维医学数据集上的收敛曲线。可以看出，在 SRBCT、Lung_Cancer、Prostate_Tumor、Tumors_9、Tumors_11、Tumors_14 这 6 个高维数据集上，代表 bDBSMA 的红线总是能够早早地收敛到底部。这说明 bDBSMA 在处理高维数据中，也能够迭代前期就获得更优的适应度值，并在迭代后期也保持稳定。进一步证实了 bDBSMA 在高维数据特征选择上，也具有好的收敛性和鲁棒性。

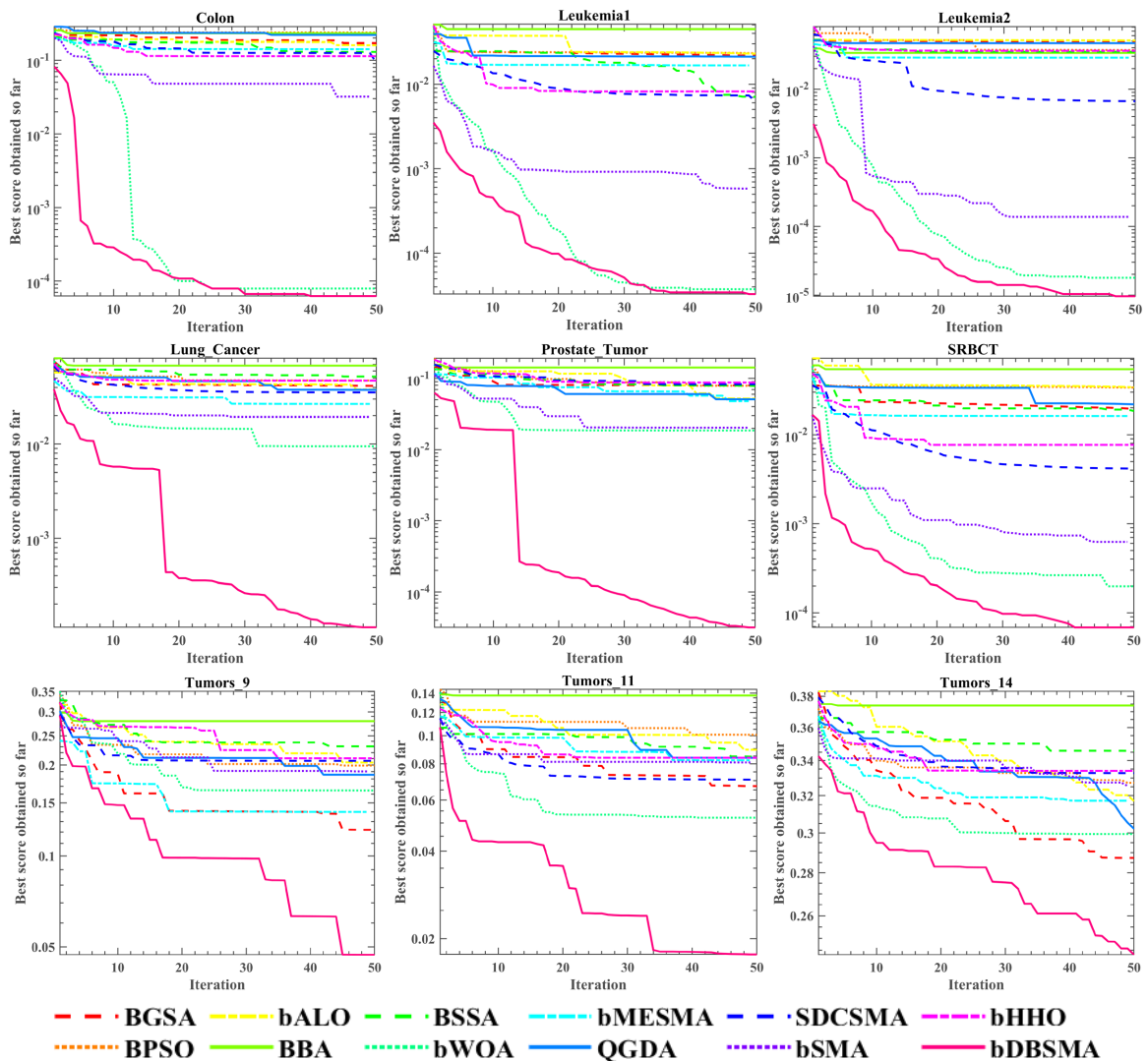


Figure 9. Convergence Curves of bDBSMA on nine high-dimensional medical datasets

图 9. bDBSMA 在 9 个高维医学数据集上的收敛曲线

通过以上实验结果和分析, 可以得出 **bDBSMA** 在高维特征选择上具有优异的算法性能。结合 **bDBSMA** 算法本身, 可以进一步分析得出, **bDBSMA** 优异的性能主要得益于两个策略的引入: (1) 差分进化机制的引入增强了种群的多样性, 通过变异和交叉操作产生更具潜力的候选解, 帮助算法跳出局部最优。(2) 分散觅食策略通过分配部分个体专门负责探索未知区域, 扩大了搜索范围, 有效避免了算法过早收敛。(3) 两种机制的协同作用使得 **bDBSMA** 在探索与开发之间实现了更好的平衡, 从而在保证高分类精度的同时, 能够筛选出最具代表性的特征子集。

综上所述, 所提出的融合差分进化与分散觅食策略显著地提高了 **SMA** 对高维数据特征选择任务的分类性能, 且 **bDBSMA** 尤其适用于对分类准确性和模型可解释性要求较高的医疗诊断场景。

Table 4. Friedman test of algorithms on four evaluation metrics

表 4. 各算法在四个评估指标上的 Friedman 检验

	平均适应度		平均错误率		平均特征数量		平均运行时间	
	ARV	排名	ARV	排名	ARV	排名	ARV	排名
bDBSMA	1.22	1	1.44	1	1.44	1	6.33	8
BGSA	7	7	7.67	9	7	8	5.78	7
BPSO	8.33	10	8.11	10	8.89	12	3.78	4
bALO	7.22	8	7.44	8	7.33	10	3.33	3
BBA	8.56	11	9.33	12	6.78	7	4.67	6
BSSA	6.22	5	5.33	5	6.33	5	4.11	5
bWOA	3.33	2	2.89	2	5.22	4	2.69	1
bMESMA	6.89	6	6	6	6.44	6	9.33	11
QGDA	8.22	9	6.89	7	8.33	11	7.56	9
SDCSMA	5.78	4	4.67	4	5.22	3	12	12
bSMA	4.44	3	3.89	3	3.67	2	2.78	2
bHHO	8.78	12	9	11	7.22	9	8.22	10

6. 结论和未来方向

黏菌算法 **SMA** 在全局优化过程中存在的探索与开发能力失衡、易陷入局部最优, 针对这个问题, 本文提出了一种融合差分进化机制与分散觅食策略的增强型黏菌算法 **DBSMA**。该算法通过引入差分进化的变异交叉操作增强种群多样性, 帮助个体跳出局部最优; 同时借助分散觅食策略分配部分黏菌个体专门负责探索未知区域, 扩大全局搜索范围。两种机制的协同作用使 **DBSMA** 在探索与开发之间实现了更为有效的平衡。

本文将 **DBSMA** 与 12 种主流算法及 16 种改进算法在 IEEE CEC 2017 标准测试集上进行对比实验, 并采用 Friedman 检验和 Wilcoxon 符号秩检验对实验结果进行统计分析。实验结果表明, **DBSMA** 在求解精度、收敛速度和鲁棒性方面均展现出明显优势。在总体排名中, **DBSMA** 位列第一, 显著优于其他对比算法。收敛曲线分析进一步表明, **DBSMA** 在各类测试函数上均能更快收敛至更优解, 体现了其强大的全局搜索与局部开发能力。

在此基础上, 本研究通过引入 V 形传递函数构建了适用于特征选择问题的二进制版本 **bDBSMA**。本文选取了 9 个高维医学数据集进行测试, 将 **bDBSMA** 与 11 种二进制优化算法进行对比, 采用十折交叉

验证和 KNN 分类器评估各算法性能。最终 5.3 节中 Friedman 检验结果表明, 四个评价指标, bDBSMA 在平均适应度、分类错误率和特征数量三个指标上均排名第一, 显著优于其它 11 个对比算法。其中 bDBSMA 分类错误率的 ARV 只有 1.44, 选择的特征数量 ARV 为 1.44, 这说明相比于其它算法, 本文提出的 bDBSMA 能够在保持高分类精度的同时, 有效降低特征维度, 验证了 bDBSMA 算法在特征筛选工作中的可行性与实用价值。

尽管本文已经在优化性能和特征选择实验上证实了 DBSMA 算法的优越性, 但该算法仍然存在一些需要进一步研究的问题。例如, 虽然 DBSMA 在处理大规模优化问题时, 能够在搜索空间内找到更好的解, 但是 DBSMA 需要较高的计算时间成本; 后续研究可以考虑将 DBSMA 算法改为并行模式, 以此提高运算效率, 减少时间成本。另外, 本文只将算法运用在高维基因数据特征选择中, 在其它复杂的优化领域并没有进行实验; 因此, 在未来研究中, 可以考虑将 DBSMA 算法应用在更多实际优化问题中。

参考文献

- [1] Guyon, I. and Elisseeff, A. (2003) An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, **3**, 1157-1182.
- [2] Donoho, D.L. (2000) High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality. Stanford University, 1-32.
- [3] Liu, H. and Yu, L. (2005) Toward Integrating Feature Selection Algorithms for Classification and Clustering. *IEEE Transactions on Knowledge and Data Engineering*, **17**, 491-502. <https://doi.org/10.1109/tkde.2005.66>
- [4] Chandrashekar, G. and Sahin, F. (2014) A Survey on Feature Selection Methods. *Computers & Electrical Engineering*, **40**, 16-28. <https://doi.org/10.1016/j.compeleceng.2013.11.024>
- [5] Blum, C. and Roli, A. (2003) Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys*, **35**, 268-308. <https://doi.org/10.1145/937503.937505>
- [6] Fortini, P. and Barakat, R. (1981) An Algorithm for Gene Frequency Changes for Linked Autosomal Loci Based on Genetic Algebras. *Journal of Mathematical Analysis and Applications*, **83**, 135-143. [https://doi.org/10.1016/0022-247x\(81\)90252-3](https://doi.org/10.1016/0022-247x(81)90252-3)
- [7] Kennedy, J. and Eberhart, R. (1995) Particle Swarm Optimization. *Proceedings of ICNN'95—International Conference on Neural Networks*, Perth, 27 November-1 December 1995, 1942-1948. <https://doi.org/10.1109/icnn.1995.488968>
- [8] Mirjalili, S., Mirjalili, S.M. and Lewis, A. (2014) Grey Wolf Optimizer. *Advances in Engineering Software*, **69**, 46-61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- [9] Mirjalili, S. and Lewis, A. (2016) The Whale Optimization Algorithm. *Advances in Engineering Software*, **95**, 51-67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- [10] Li, S., Chen, H., Wang, M., Heidari, A.A. and Mirjalili, S. (2020) Slime Mould Algorithm: A New Method for Stochastic Optimization. *Future Generation Computer Systems*, **111**, 300-323. <https://doi.org/10.1016/j.future.2020.03.055>
- [11] Mostafa, M., Rezk, H., Aly, M. and Ahmed, E.M. (2020) A New Strategy Based on Slime Mould Algorithm to Extract the Optimal Model Parameters of Solar PV Panel. *Sustainable Energy Technologies and Assessments*, **42**, Article ID: 100849. <https://doi.org/10.1016/j.seta.2020.100849>
- [12] Zhao, S., Wang, P., Heidari, A.A., Chen, H., Turabieh, H., Mafarja, M., et al. (2021) Multilevel Threshold Image Segmentation with Diffusion Association Slime Mould Algorithm and Renyi's Entropy for Chronic Obstructive Pulmonary Disease. *Computers in Biology and Medicine*, **134**, Article ID: 104427. <https://doi.org/10.1016/j.compbiomed.2021.104427>
- [13] Premkumar, M., Jangir, P., Sowmya, R., Alhelou, H.H., Heidari, A.A. and Chen, H. (2021) MOSMA: Multi-Objective Slime Mould Algorithm Based on Elitist Non-Dominated Sorting. *IEEE Access*, **9**, 3229-3248. <https://doi.org/10.1109/access.2020.3047936>
- [14] Abdel-Basset, M., Chang, V. and Mohamed, R. (2020) HSMA_WOA: A Hybrid Novel Slime Mould Algorithm with Whale Optimization Algorithm for Tackling the Image Segmentation Problem of Chest X-Ray Images. *Applied Soft Computing*, **95**, Article ID: 106642. <https://doi.org/10.1016/j.asoc.2020.106642>
- [15] Altay, O. (2022) Chaotic Slime Mould Optimization Algorithm for Global Optimization. *Artificial Intelligence Review*, **55**, 3979-4040. <https://doi.org/10.1007/s10462-021-10100-5>
- [16] Storm, R. and Price, K. (1997) Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over

- Continuous Spaces. *Journal of Global Optimization*, **11**, 341-359. <https://doi.org/10.1023/a:1008202821328>
- [17] Hu, J., Gui, W., Heidari, A.A., Cai, Z., Liang, G., Chen, H., *et al.* (2022) Dispersed Foraging Slime Mould Algorithm: Continuous and Binary Variants for Global Optimization and Wrapper-Based Feature Selection. *Knowledge-Based Systems*, **237**, Article ID: 107761. <https://doi.org/10.1016/j.knosys.2021.107761>