面向大规模多目标特征选择的竞争性人工蜂群算法

王 晖[1]，钱佳瑶[1]，肖人彬[2]，王文君[3]，赵 嘉[1]，周新宇[4†]

(1. 江西水利电力大学 信息工程学院，南昌 330099；2. 华中科技大学 人工智能与自动化学院，武汉 430074；
3. 江西水利电力大学 工商管理学院，南昌 330099；4. 江西师范大学 人工智能学院，南昌 330022)

摘 要: 针对高维数据多目标特征选择存在的搜索空间大、优化效率低、易陷入局部最优等问题, 提出一种面向大规模多目标特征选择的竞争性人工蜂群算法. 该方法以特征子集规模为竞争依据将种群划分为败者与胜者, 分别设计基于特征约简的雇佣蜂搜索策略和基于错误率优化的跟随蜂搜索策略, 以实现差异化寻优. 采用外部存档引导的侦查蜂搜索策略以增强全局搜索能力, 避免算法早熟收敛. 在 12 个不同维度数据集上的实验结果表明, 所提出算法在超体积、最低分类错误率及运行效率等方面表现出较强竞争力, 能够快速获得特征规模小、分类精度高的 Pareto 最优解集.

关键词: 人工蜂群算法；多目标优化；大规模多目标优化；特征选择；特征约简；进化计算

中图分类号: **文献标志码:** A

DOI: 10.13195/j.kzyjc.2026.0514

引用格式: 王晖, 钱佳瑶, 肖人彬, 等. 面向大规模多目标特征选择的竞争性人工蜂群算法 [J]. 控制与决策, xxxx, x(x): xxxx-xxxx.

A competitive artificial bee colony algorithm for large-scale multi-objective feature selection

WANG Hui[1]，*QIAN Jia-yao*[1]，*XIAO Ren-bin*[2]，*WANG Wen-jun*[3]，*ZHAO Jia*[1]，*ZHOU Xin-yu*[4†]

(1. School of Information Engineering， Jiangxi University of Water Resources and Electric Power， Nanchang 330099， China； 2. School of Artificial Intelligence and Automation， Huazhong University of Science and Technology，Wuhan 430074，China；3. School of Business Administration，Jiangxi University of Water Resources and Electric Power， Nanchang 330099， China； 4. School of Artificial Intelligence， Jiangxi Normal University，Nanchang 330022，China)

Abstract: Multi-objective feature selection on high-dimensional data often faces problems such as large search space, low optimization efficiency, and being trapped in local optima. To address these issues, this paper proposes a competition-based artificial bee colony algorithm for large-scale multi-objective feature selection. The proposed method divides the population into losers and winners based on feature subset size. For each group, a feature reduction-oriented employed bee search strategy and an error rate optimization-oriented onlooker bee search strategy are designed separately to achieve differentiated optimization. In addition, an archive-guided scout bee search strategy is introduced to enhance global search ability and prevent premature convergence. Experiments on 12 datasets of varying dimensions show that the proposed approach is highly competitive in terms of hypervolume, minimum classification error rate, and computational efficiency. It can quickly find a Pareto optimal solution set with small feature subsets and high classification accuracy.

Keywords: artificial bee colony algorithm； multi-objective optimization； large-scale multi-objective optimization； feature selection； feature reduction； evolutionary computation

0 引 言

现实生活中存在大量需要同时优化多个目标的复杂问题, 这些目标往往相互冲突, 此类问题被称为多目标优化问题[1,2], 受到了学术界和工业界的广泛关注. 多目标优化问题旨在寻找一组 Pareto 最优解,

收稿日期: 2026-05-29；**录用日期:** 2026-10-12.
基金项目: 国家自然科学基金项目 (62166027).
责任编委: 陈家伟.
†通信作者. E-mail: xyzhou@jxnu.edu.cn.
本文附带电子附录文件, 可登录本刊官网该文“资源附件”区自行下载阅览.

这组解能够在多个冲突目标之间取得良好权衡, 从而为决策者提供灵活的选择方案. 随着大数据技术的飞速发展, 越来越多的多目标优化问题涉及成百上千个决策变量, 搜索空间随变量规模呈指数级增长, 在维持多目标之间良好平衡的同时还需应对高维决策空间的维度灾难问题, 给算法的收敛性与多样性带来了更为严峻的挑战. 这类同时具有多目标与大规模特性的优化问题, 统称为大规模多目标优化问题[3]. 多目标特征选择以最小化分类错误率和特征子集大小为优化目标, 旨在从高维数据集中筛选出最优特征子集, 是最具代表性和挑战性的一类大规模多目标优化问题[4,5]. 近年来, 众多研究将大规模多目标优化技术引入多目标特征选择, 相关方法主要分为三类: 基于分组的方法、基于降维的方法、基于增强搜索能力的方法[3].

基于分组的方法通过划分特征子组降低搜索复杂度. Cheng 等人提出 VGS-MOEA[6], 采用由粗到细的变粒度分组策略逐步细化搜索空间. Yu 等人提出 BDGMOEA[7], 基于 SU[8] 和 ReliefF[9] 进行特征分组, 并通过双向动态调整提升搜索效率和子集质量. Xia 等人提出 ISA-EFS[10], 根据 Pareto 个体共同选择的特征动态构建和调整分组, 以探索潜在的互补特征组合.

基于降维的方法通过预先过滤无关或冗余特征, 缩减后续进化搜索的规模. Ahadzadeh 等人提出 SFE-PSO[11], 先批量剔除不重要特征, 再在低维空间中进行精细搜索. Deng 等人提出 FTGGA[12], 利用 ReliefF 筛选高排名特征, 并以动态阈值引导交叉和变异. Yue 等人提出 DR-RPMODE[13], 通过冻结算子快速移除无关特征, 再结合冗余与偏好处理执行多目标差分进化.

基于增强搜索能力的方法旨在通过改进搜索策略, 提高高维特征空间中的搜索效率与解集质量. 人工蜂群 (ABC) 算法因结构简单、全局搜索能力强, 被广泛用于多目标特征选择. Hancer 等人提出 MOABC-FS[14], 结合人工蜂群算法与非支配排序, 同时优化分类错误率和特征数量. Wang 等人提出 FMABC-FS[15], 通过样本缩减和粒子更新策略降低评价开销并提高搜索效率. 除 ABC 算法外, 研究者还从其它进化搜索机制出发设计了多种增强搜索能力的方法. Zhang 等人提出 MOFS-BDE[16], 利用二进制变异与净化搜索增强搜索效率和解集多样性. Zhou 等人提出 PS-NSGA[17], 通过精度优先支配和多种搜索算子加速收敛, 并从 Pareto 解集中筛选最优子集. Cheng 等人提出 SM-MOEA[18], 借助引导矩阵、降维算子和修复算子搜索精简且高质量的特征子集. Jiao 等人提出 MFFS[19], 通过多任务协同进化、自适应搜索方向和重复解过滤兼顾搜索能力与种群多样性. Zhang 等人提出 IGEA[20], 以信息增益引导交叉和变异, 提高高维空间中的搜索效率. Liang 等人提出 JE-MMOEA[21], 通过离散—连续联合编码及相应搜索算子增强探索能力和特征子集多样性.

尽管上述方法在一定程度上缓解了多目标特征选择的优化难题, 但仍存在局限: 特征分组可能破坏特征间的协同关系, 预先降维可能误删有价值的特征组合, 而复杂搜索算子通常会增加计算开销. 实际上, 进化过程中种群内个体的特征数量天然存在差异. 引导特征较多的个体向特征较少的个体学习, 可以逐步剔除冗余特征、压缩搜索维度; 特征较少的个体受冗余特征干扰较小, 因而可侧重优化分类错误率, 使搜索方向更加明确. 由此, 以特征数量为竞争依据的差异化学习机制无需预先分组或降维, 也无需设计复杂算子, 即可推动种群向特征更少、错误率更低的方向演化. 基于竞争性群智能算法的思想[22], 本文设计了一种竞争性差异化搜索策略, 并结合 ABC 算法提出面向大规模多目标特征选择的竞争性人工蜂群算法 (CABC-MOFS). 本文的主要贡献如下:

(1) 提出以特征数为竞争依据的差异化搜索框架, 将种群个体划分为胜者和败者, 并设计特征状态学习策略使不同角色能够更有效地学习, 实现分工明确的协同优化.

(2) 在雇佣蜂阶段, 提出了基于特征约简的搜索策略, 以胜者为参考对象, 通过特征状态学习策略驱使败者主动删减冗余特征, 持续施加降维压力, 引导种群向特征数更少的区域收敛.

(3) 在跟随蜂阶段, 提出了基于错误率优化的搜索策略, 通过构造双层候选池选取分类性能更优的个体作为参考对象, 在特征数已较少、冗余干扰低的基础上进一步降低错误率, 引导种群向分类性能更优的区域收敛.

(4) 在侦查蜂阶段, 提出了基于存档引导的搜索策略, 维护包含非支配解和错误率最低解的外部存档, 当个体陷入停滞时, 以与其特征差异最大的存档解为引导进行重启, 帮助种群跳出局部最优, 增强算法的全局搜索能力.

1 大规模多目标特征选择问题

现实数据集通常包含数千甚至数万个特征. 特

征选择旨在从原始特征中筛选具有判别力的子集, 以降低计算复杂度、缓解过拟合并提升模型泛化能力. 由于特征子集的搜索空间随特征维度呈指数级增长, 且需要同时最小化相互冲突的分类错误率和特征数量. 因此, 特征选择问题本质上是一个大规模多目标优化问题.

大规模多目标特征选择问题的形式定义如下:

$$\begin{aligned}&\text{minimize } F(X)=(f_{\text{error}}(X),f_{\text{ratio}}(X)),\\&\text{subject to } X=(x_1,\ldots,x_D)\in\{0,1\}^D,\end{aligned}\quad(1)$$

其中, X表示一个候选特征子集的二进制决策向量, D为特征总数. 每个x_j对应第j个特征, $x_j=1$表示该特征被选入, $x_j=0$表示该特征未被选入.

第一个目标$f_{\text{error}}(X)$为特征子集X的分类错误率, 衡量特征子集的分类性能. 为避免类别不平衡对评估结果的影响, 本文采用平衡错误率, 定义为:

$$f_{\text{error}}(X)=\frac{1}{c}\sum_{i=1}^{c}\frac{FP_i}{|S_i|},\quad(2)$$

其中, c为类别总数, FP_i为第i类中被错误分类的样本数, $|S_i|$为第i类的样本总数.

第二个目标$f_{\text{ratio}}(X)$为特征选择率, 衡量特征子集的精简性, 定义为:

$$f_{\text{ratio}}(X)=\frac{1}{D}\sum_{j=1}^{D}x_j.\quad(3)$$

2　竞争性大规模多目标人工蜂群算法

2.1　基于竞争的角色划分

算法在每轮迭代中通过两两竞争划分种群个体角色, 其竞争方式借鉴了竞争性群智能算法[22]的思想. 具体而言, 种群个体被随机两两配对, 形成$N/2$组互不重复的竞争组合. 对任意一组配对个体X_i和X_k, 以特征数量作为评判依据: 特征较多的个体为败者X_{lose}, 特征较少的个体为胜者X_{win}. 两类个体根据竞争角色采用差异化搜索策略, 并向参考个体X_{ref}学习, 以提高算法的优化性能. 败者执行基于特征约简的雇佣蜂搜索, 向胜者学习以减少特征; 胜者执行基于错误率优化的跟随蜂搜索, 在冗余干扰较低的基础上提升分类性能. 通过对败者和胜者明确分工, 协同推进种群高效搜索到特征数少且分类性能优的特征子集.

2.2　特征状态学习策略

在差异化搜索过程中, 胜者与败者均通过学习参考个体X_{ref}完成自身特征子集的更新. 本文基于ABC 算法, 采用连续编码方式对种群个体进行编码, 每个个体对应一组候选特征子集, 其每一维表示数据集的一个特征, 取值满足$x_{i,j}\in[0,1]$. 为将连续优化结果转化为特征选择决策, 本文设定阈值$\theta=0.5$, 依据连续编码分量的数值大小判定特征取舍, 从而将连续解空间映射为离散的特征选取状态. 定义二进制状态变量$b_{i,j}$表示各特征的选取状态, 通过阈值规则完成映射, 依据判定结果即可得到实际选取的特征子集, 进而完成特征子集构建与分类性能评价. $b_{i,j}$的取值规则定义如下:

$$b_{i,j}=\begin{cases}1 \text{ if } x_{i,j}\geqslant\theta\\0 \text{ if } x_{i,j}<\theta\end{cases},\quad(4)$$

当$x_{i,j}\geqslant\theta$时, 表示个体X_i选中了第j个特征, 记为$b_{i,j}=1$; 反之, 表示未选中该特征, 记为$b_{i,j}=0$. 需要注意的是, $b_{i,j}$仅作为辅助符号用于特征子集的提取与评估, 算法的进化搜索、个体更新等核心操作仍在连续变量$x_{i,j}$上执行.

根据公式 (4), 特征选取状态由个体连续编码值与阈值θ的相对大小决定, 若直接使用标准 ABC 算法的搜索算子随机扰动个体, 生成的新个体无法有效借鉴参考个体的特征选择信息, 寻优过程随机性较强. 为此, 本文提出一种特征状态学习策略, 使种群个体在保持多样性的同时更有针对性地学习参考个体经验. 根据当前个体X_i和参考个体X_{ref}在第j维特征上的选择状态, 将学习过程分为两种情况:

(1) 当X_i和X_{ref}在第j维特征上选取状态一致 (即同时选中或同时舍弃该特征) 时, 二者在该特征上无选择差异, 此时利用标准 ABC 算法搜索算子的随机特性对该维度进行探索, 以维持种群多样性.

$$x_{i,j}(t+1)=x_{i,j}(t)+\phi\cdot(x_{i,j}(t)-x_{k,j}(t)),\quad(5)$$

其中, X_k为从种群中随机选取的个体, $\phi\sim U(-1,1)$为随机扰动因子, 控制学习步长. 由于更新操作可能导致连续编码变量超出取值范围$[0,1]$, 本文采用边界截断策略将越界变量修正至可行范围, 即通过$x_{i,j}=\max(0,\min(1,x_{i,j}))$将越界变量修正至可行范围.

(2) 当X_i和X_{ref}在第j维特征上的选取状态不一致时, 参考个体X_{ref}能够提供更具针对性的指导信息, 应以其为学习引导, 采用对应的策略完成维度更新. 具体地, 第j维的更新值由参考个体在该维的选取状态$b_{\text{ref},j}$及特征重要性FI_j共同决定, 如公式 (6) 所示.

$$x_{i,j}(t+1)=\begin{cases}U(\theta,1) \text{ if } b_{\text{ref},j}=1\wedge r_j<FI_j\\U(0,\theta) \text{ if } b_{\text{ref},j}=0\wedge r_j<(1-FI_j)\\x_{i,j}(t) \text{ otherwise}\end{cases},\quad(6)$$

其中, $r_j \sim U(0,1)$, $FI_j \in [0,1]$为第j维特征的特征重要性, 本文采用 SU[8] 指标计算, FI_j越大表示该特征与类标签的相关性越强, 即特征越重要.

若参考个体选中第j维特征, 即满足$b_{\text{ref},j}=1$, 表明该特征具有较高的贡献; 随后以特征重要度FI_j为采样概率, 对分量$x_{i,j}$在区间$(\theta,1]$内重新采样, 令其取值大于阈值θ, 完成该特征的选择操作. 若参考个体未选中第j维特征, 即$b_{\text{ref},j}=0$, 表明该特征的贡献偏低, 以概率$1-FI_j$对$x_{i,j}$在区间$[0,\theta)$内重采样, 使其取值小于阈值θ, 实现特征剔除. 其余情形下, 参考个体对此维度引导作用较弱, 故保留原数值不变, 避免种群个体过度向参考个体收敛.

2.3 基于特征约简的雇佣蜂搜索阶段

在标准 ABC 算法中, 雇佣蜂负责在全局范围内探索以改善当前个体. 败者个体所含特征数量多于胜者, 表明其特征子集存在较多冗余维度, 亟需对特征结构进行大幅度调整, 这一优化需求与雇佣蜂的全局探索特性高度契合. 因此, 本文将败者的学习过程对应雇佣蜂搜索阶段. 该阶段内, 败者个体X_{lose}以胜者个体X_{win}作为学习参考个体, 即令$X_{\text{ref}}=X_{\text{win}}$, 并执行特征状态学习策略, 具体实现流程如算法 1 所示. 借助该学习机制, 败者可有效借鉴胜者的特征选择经验. 由于胜者具备更精简的特征规模, 其舍弃的特征大多为无效冗余特征, 败者在学习过程中能够逐步剔除自身冗余特征, 持续推动特征维度精简, 引导种群向特征数更少的区域收敛.

算法 1 基于特征约简的雇佣蜂搜索阶段

输入: X_{lose}(败者), X_{win}(胜者);

输出: X'_{lose}(更新后的败者);

step 1: for $j=1$ **to** D;

step 2: if $b_{\text{lose},j}=b_{\text{win},j}$;

step 3: $x'_{\text{lose},j}\leftarrow$使用公式 (5) 更新;

step 4: else;

step 5: $x'_{\text{lose},j}\leftarrow$使用公式 (6) 更新;

step 6: end if;

step 7: end for;

step 8: **return** X'_{lose};

2.4 基于错误率优化的跟随蜂搜索阶段

完成雇佣蜂阶段搜索后, 败者已在降维导向下逐步剔除冗余特征. 胜者的特征子集结构精简, 冗余特征较少, 无需对特征结构进行大幅度调整, 其优化重心转为向分类性能更优的个体学习, 在保持特征数量精简的同时进一步降低分类错误率. 在 ABC 算法中, 跟随蜂依据个体的质量选择性地跟随优质个体进行局部精细化搜索, 与胜者的优化需求相契合. 因此, 本文将胜者的学习过程对应跟随蜂搜索阶段.

为引导胜者朝着更低分类错误率的方向迭代优化, 本文为胜者X_{win}构建双层候选池$\mathcal{C}$, 并从中筛选合适的参考个体X_{ref}, 引导胜者向错误率更低的方向探索. 第一层候选池$\mathcal{C}_1$包含种群中错误率与特征数量均优于X_{win}的个体, 该类个体在分类性能和特征精简程度上均更优, 是最优的学习对象. 如果第一层候选池$\mathcal{C}_1$为空, 则从第二层候选池$\mathcal{C}_2$中选取参考个体, $\mathcal{C}_2$仅包含种群中错误率低于X_{win}的个体, 不再约束特征数量, 以确保所选参考个体至少在分类性能上优于胜者. 如果两层候选池均为空, 则保留当前胜者个体不变. 确定参考个体后, 胜者以候选池选出的X_{ref}为参考执行特征状态学习策略, 通过引入优质特征以降低分类错误率, 具体流程如算法 2 所示.

2.5 基于存档引导的侦查蜂搜索阶段

经过雇佣蜂和跟随蜂阶段的差异化搜索, 种群可在精简特征数量的同时持续提升分类性能, 但部分个体可能在搜索过程中陷入局部最优, 使得优化进程陷入停滞. 在标准 ABC 算法中, 侦查蜂会舍弃停滞个体并随机生成新个体, 以帮助种群跳出局部最优. 然而, 这种纯随机的重启方式缺乏方向性, 尤其是在特征选择问题中, 随机生成的特征子集通常存在大量冗余维度, 难以有效提升分类效果. 针对上述问题, 本文对侦查蜂搜索机制进行改进, 提出基于存档引导的重启策略, 借助算法迭代过程中留存的优质历史个体, 为停滞个体提供有方向性的重启引导.

算法 2 基于错误率优化的跟随蜂搜索阶段

输入: X_{lose}(败者), X_{win}(胜者);

输出: X'_{win}(更新后的胜者);

step 1: $[\mathcal{C}_1,\mathcal{C}_2]\leftarrow$构造$X_{\text{win}}$的双层候选池;

step 2: if $\mathcal{C}_1\neq\emptyset$;

step 3: $X_{\text{ref}}\leftarrow$从$\mathcal{C}_1$中随机选取一个个体;

step 4: else if $\mathcal{C}_2\neq\emptyset$;

step 5: $X_{\text{ref}}\leftarrow$从$\mathcal{C}_2$中随机选取一个个体;

step 6: else;

step 7: $X_{\text{ref}}\leftarrow X_{\text{win}}$;

step 8: end if;

step 9: for $j=1$ **to** D;

step 10: if $b_{\text{win},j}=b_{\text{ref},j}$;

step 11: $x'_{\text{win},j}\leftarrow$使用公式 (5) 更新;

step 12: else;

step 13: $x'_{\text{win},j}\leftarrow$使用公式 (6) 更新;

step 14: end if;

step 15: end for;

step 16: return X'_{win};

为保存算法迭代过程中产生的优质历史个体, 本文引入一个外部存档$\mathcal{A}$, 其更新策略如算法 3 所示. 每次迭代结束后, 将当前种群P与存档$\mathcal{A}$合并得到集合$\mathcal{U}$, 并从$\mathcal{U}$中分别筛选出错误率最低的个体和非支配个体, 存入集合S_{min}和S_{nd}. 首先将S_{min}中全部个体加入存档$\mathcal{A}$, 若$\mathcal{A}$中个体的数量超过其容量上限N_A时, 则按特征子集规模由小到大排序, 仅保留前N_A个最精简的个体. 若$\mathcal{A}$仍未满, 则从S_{nd}中按拥挤度距离由大到小依次选择个体补充至$\mathcal{A}$, 直至$\mathcal{A}$存满或S_{nd}中全部个体加入, 以维持种群多样性.

算法 3　存档更新策略

输入: P(种群), $\mathcal{A}$(存档), N_A(存档大小);

输出: $\mathcal{A}$(更新后的存档);

step 1: $\mathcal{U} \leftarrow P \cup \mathcal{A}$;

step 2: $\mathcal{A} \leftarrow \emptyset$;

step 3: $S_{min} \leftarrow$从$\mathcal{U}$中筛选最低错误率的个体集合;

step 4: $S_{nd} \leftarrow$从$\mathcal{U}$中筛选非支配个体集合;

step 5: $\mathcal{A} \leftarrow S_{min}$;

step 6: if $|\mathcal{A}| > N_A$

step 7: 按特征子集规模升序对$\mathcal{A}$中的个体排序;

step 8: $\mathcal{A} \leftarrow$从$\mathcal{A}$中选择前N_A个个体;

step 9: end if

step 10: if $|\mathcal{A}| < N_A$

step 11: $k \leftarrow \min(N_A - |\mathcal{A}|,\ |S_{nd}|)$;

step 12: 按拥挤度距离降序对S_{nd}中的个体排序;

step 13: $S'_{nd} \leftarrow$从S_{nd}中选择前k个个体;

step 14: $\mathcal{A} \leftarrow \mathcal{A} \cup S'_{nd}$;

step 15: end if

step 16: return $\mathcal{A}$

随着搜索进程推进, 存档积累了各阶段的优质特征组合, 其中可能包含在竞争筛选过程中丢失的有价值特征, 能够为陷入停滞的个体提供可靠的优化参考. 当某个个体连续$limit$代未出现性能提升时, 即可判定该个体陷入停滞. 此时从外部存档$\mathcal{A}$中选取与当前停滞个体特征差异度最大的个体作为参考个体X_{ref}, 其定义为:

$$X_{\text{ref}} = \arg\max_{X_k \in \mathcal{A}} \sum_{j=1}^{D} |b_{i,j} - b_{k,j}|, \tag{7}$$

其中, D为特征总数, $b_{i,j}$和$b_{k,j}$分别为个体X_i和X_k在第j维特征上的选择状态.

选取存档$\mathcal{A}$中与停滞个体差异最大的优质个体作为参考个体, 可最大限度引导其探索全新特征组合, 充分挖掘迭代过程中被遗漏的有效特征. 随后停滞个体以X_{ref}为参考, 采用特征状态学习策略完成自身更新, 具体流程如算法 4 所示, 从而帮助种群跳出局部最优.

算法 4　基于存档引导的侦查蜂搜索阶段

输入: P(种群), $\mathcal{A}$(存档), $limit$(停滞阈值);

输出: R(重启后的个体集合);

step 1: $R \leftarrow \emptyset$;

step 2: for 每个$X_i \in P$;

step 3: if X_i连续$limit$代未能改善;

step 4: $X_{\text{ref}} \leftarrow$通过公式 (7) 选取与$X_i$特征差异最大的个体;

step 5: for $j = 1$ **to** D;

step 6: if $b_{i,j} = b_{\text{ref},j}$;

step 7: $x'_{i,j} \leftarrow$使用公式 (5) 更新;

step 8: else;

step 9: $x'_{i,j} \leftarrow$使用公式 (6) 更新;

step 10: end if;

step 11: end for;

step 12: $R \leftarrow R \cup \{X'_i\}$;

step 13: end if;

step 14: end for;

step 15: return R;

2.6　算法框架

算法首先初始化种群P和存档$\mathcal{A}$(第 1–2 行). 进入主循环后, 每次迭代将种群个体两两随机配对, 形成$N/2$组互不重复的配对组 (第 5 行), 并依据特征数量竞争, 从而将种群划分为胜者与败者两类角色. 随后, 对败者和胜者分别调用算法 1 和算法 2, 生成子代集合O(第 6–10 行). 对于种群中连续若干代未得到改进的停滞个体, 调用算法 4 对其进行重启, 得到子代集合R(第 11 行). $O \cup R$构成本代产生的全部子代. 将父代与子代合并为$P \cup O \cup R$后, 采用经典的非支配排序与拥挤度距离策略[23] 进行环境选择, 保留N个个体以更新种群 (第 12 行), 并调用算法 3 更新存档$\mathcal{A}$(第 13 行). 当算法达到终止条件后, 输出存档$\mathcal{A}$中的解集作为最终结果 (第 15 行).

算法 5　CABC-MOFS

输入: D(特征数), N(种群大小), $limit$(停滞阈值), N_A(存档大小);

输出: $\mathcal{A}$(存档);

step 1: $P \leftarrow$随机初始化种群;

step 2: $\mathcal{A} \leftarrow \emptyset$;
step 3: while 算法未终止;
step 4: $O \leftarrow \emptyset$;
step 5: 将种群P划分为$N/2$对竞争组合;
step 6: for 每对竞争组合$[X_{\text{lose}}, X_{\text{win}}]$;
step 7: $X'_{\text{lose}} \leftarrow$根据算法 1 执行雇佣蜂搜索;
step 8: $X'_{\text{win}} \leftarrow$根据算法 2 执行跟随蜂搜索;
step 9: $O \leftarrow O \cup \{X'_{\text{lose}}, X'_{\text{win}}\}$;
step 10: end for;
step 11: $R \leftarrow$根据算法 4 执行侦查蜂搜索;
step 12: $P \leftarrow$从$P \cup O \cup R$中挑选N个个体;
step 13: $\mathcal{A} \leftarrow$根据算法 3 对存档进行更新;
step 14: end while;
step 15: return $\mathcal{A}$;

3 实验研究

3.1 实验设置

3.1.1 数据集

本文选取 12 个高维数据集开展对比实验, 这些数据集均取自 UCI 机器学习库等开源数据库[24], 详细信息如表 1 所示. 所选数据集涵盖了中低维至高维的多种场景, 其中大部分数据集具有典型的高维特性, 对算法在高维特征空间中的搜索能力提出了较高要求. 对于每个数据集, 70% 的样本用于训练, 剩余 30% 用于测试.

表 1 12 个实验数据集的详细信息

数据集	特征数	实例数	类别数
ionosphere	34	351	2
Libras	90	360	15
MUSK1	166	476	2
gse72526	800	61	4
Yale	1,024	165	15
colon	2,000	62	2
SRBCT	2,308	83	4
warpAR10P	2,420	130	10
GLIOMA	4,434	50	4
Leukemia1	5,327	72	3
DLBCL	5,469	77	2
ALLAML	7,129	72	2

本文采用 KNN 分类器对候选特征子集和最终特征子集进行评价. KNN 是一种基于实例学习的非参数分类方法, 通过计算样本之间的距离, 并利用邻近样本的类别信息完成分类[25]. KNN 不需要复杂的模型训练过程, 能够直接反映不同特征子集的分类能力, 因此被广泛应用于多目标特征选择算法的性能评价中[18,19,21,26]. 在进化优化阶段, 由于算法需要频繁调用分类器评价大量候选特征子集, 本文采用 KNN($K = 1$) 结合留一交叉验证计算候选解的分类错误率, 并将其作为适应度函数值. 在最终测试阶段, 为了更加稳定地评价最终特征子集的泛化能力, 本文采用 KNN($K = 5$) 进行分类测试. 相比$K = 1$, $K = 5$能够综合多个邻近样本的信息, 降低单个样本噪声对分类结果的影响.

3.1.2 对比算法

本文选取八种具有代表性的多目标特征选择算法 (MOFS) 作为对比方法, 分别为 MOABC-FS[14]、FMABC-FS[15]、MOFS-BDE[16]、PS-NSGA[17]、MFFS[19]、SM-MOEA[18]、IGEA[20] 和 JE-MMOEA[21]. 其中, MOABC-FS 和 FMABC-FS 为基于 ABC 的 MOFS 算法, 在该类方法中具有较强代表性. 需要说明的是, MOABC-FS 包含二进制编码和连续型编码两个版本. 考虑到 FMABC-FS 和本文所提出的 CABC-MOFS 算法均采用连续型编码方式, 为保证编码方式的一致性, 本文选取 MOABC-FS 的连续型编码版本进行对比. MOFS-BDE、PS-NSGA、MFFS、IGEA 和 JE-MMOEA 均为通用型 MOFS 算法, 在处理高维特征选择问题上表现出良好的优化性能. SM-MOEA 则是专为高维特征选择设计的算法, 具有较强的降维能力, 可在高维特征空间中实现高效寻优. 综上, 本文从基于 ABC 的算法、通用型 MOFS 算法以及高维特征选择专用算法三个角度选取对比方法, 以此全面、客观地验证 CABC-MOFS 算法的综合性能.

3.1.3 参数设置

为保证实验对比的公平性, 本文根据各对比算法的原始文献以及相关研究中的常用设置确定其参数取值. 各算法的公共参数设置如下: 种群规模$N = 100$, 最大评估次数$maxFEs = 10000$. 对于 MOABC-FS、FMABC-FS 和 CABC-MOFS 这三种连续型编码算法, 本文均采用连续型编码特征选择算法中广泛使用的阈值$\theta = 0.5$[14,15,27], 将连续值转换为二进制取值.

3.1.4 评价指标

由于所有对比算法均为多目标优化算法, 本文选用超体积指标 (HV) 综合评价算法的收敛性与多样性. 为了从特征选择层面进一步分析算法性能, 本文引入最低分类错误率 (MCER) 作为辅助评价指标. 该指标选取训练集上分类错误率最低的特征子集, 并统计其在测试集上的分类错误率, 从而评估所选特征子集的泛化能力. 为降低统计误差, 所有算法均独立运行 30 次, 并采用显著性水平为 0.05 的 Wilcoxon 秩和检验对算法间的性能差异进行统计验证. 其中, 符号“+”、“≈”和“−”分别表示 CABC-MOFS 的

性能显著优于、与对比算法相当以及显著差于对比算法. 此外, 本文采用 Friedman 检验对所有算法在多个数据集上的综合性能进行统计分析, 并根据平均排名评价各算法的整体性能. 所有实验均在配备 Intel i5-10 500 CPU 和 16 GB 内存的 PC 上运行.

3.2　实验结果与分析

3.2.1　从多目标优化层面分析

表 2 统计了各算法在 12 组数据集上的 HV 平均值. 从实验结果可知, CABC-MOFS 在 8 个数据集上取得最高 HV 值, 仅在 Libras、Yale、GLIOMA 和 Leukemia1 数据集上未取得最优结果. Wilcoxon 秩和检验结果表明, CABC-MOFS 在大部分数据集上的性能显著优于对比算法, 且不存在性能显著劣于其他算法的情况. 此外, 从平均排名来看, CABC-MOFS 以 1.58 的平均排名值位居首位, 相较于第二名 JE-MMOEA 的 3.50 排名值优势显著, 表明本文所提算法在多目标性能上具有较强的竞争力.

表 2　CABC-MOFS 与 8 种对比算法在 12 个数据集上的 HV 值

数据集	MOABC-FS	FMABC-FS	MOFS-BDE	PS-NSGA	MFFS	SM-MOEA	IGEA	JE-MMOEA	CABC-MOFS
ionosphere	0.875 3(≈)	0.872 1(≈)	0.861 8(+)	0.654 6(+)	0.859 6(+)	0.862 1(≈)	0.810 1(+)	0.873 7(≈)	**0.876 2**
Libras	0.732 3(≈)	0.735 1(≈)	**0.742 6**(≈)	0.620 5(+)	0.736 0(≈)	0.731 9(≈)	0.629 3(+)	0.725 7(≈)	0.733 7
MUSK1	0.787 0(+)	0.843 0(≈)	0.822 0(+)	0.705 2(+)	0.849 0(≈)	0.840 6(+)	0.673 3(+)	0.854 2(≈)	**0.855 6**
gse72526	0.626 9(+)	0.710 0(+)	0.693 7(+)	0.874 0(≈)	0.867 0(+)	0.835 1(+)	0.859 2(+)	0.854 0(+)	**0.899 2**
Yale	0.444 9(+)	0.505 7(+)	0.489 1(+)	0.649 5(≈)	0.624 8(≈)	0.624 2(≈)	**0.665 5**(≈)	0.640 2(≈)	0.645 2
colon	0.493 5(+)	0.554 2(+)	0.530 6(+)	0.812 2(+)	0.839 8(≈)	0.793 2(+)	0.825 0(≈)	0.832 4(+)	**0.868 7**
SRBCT	0.566 8(+)	0.621 2(+)	0.616 9(+)	0.948 4(+)	0.930 9(+)	0.876 4(+)	0.961 3(+)	0.973 1(+)	**0.978 5**
warpAR10P	0.354 1(+)	0.399 6(+)	0.386 6(+)	0.670 1(+)	0.737 1(≈)	0.673 1(+)	0.740 8(≈)	0.572 5(+)	**0.741 3**
GLIOMA	0.479 6(+)	0.525 7(+)	0.525 4(+)	0.741 0(+)	**0.805 7**(≈)	0.736 2(+)	0.798 4(≈)	0.779 0(≈)	0.802 3
Leukemia1	0.471 1(+)	0.507 8(+)	0.522 2(+)	0.734 9(+)	0.893 3(+)	0.840 0(+)	0.920 1(≈)	**0.940 1**(≈)	0.936 3
DLBCL	0.536 5(+)	0.580 1(+)	0.569 6(+)	0.798 1(+)	0.893 6(+)	0.887 0(+)	0.915 8(+)	0.910 1(+)	**0.941 6**
ALLAML	0.479 9(+)	0.510 5(+)	0.515 1(+)	0.718 9(+)	0.909 9(+)	0.846 5(+)	0.947 2(≈)	0.953 1(≈)	**0.960 1**
+/≈/−	10/2/0	9/3/0	11/1/0	10/2/0	6/6/0	9/3/0	6/6/0	5/7/0	–
排名	7.92	6.33	6.92	5.58	3.58	5.42	4.17	3.50	**1.58**

CABC-MOFS 的性能优势主要得益于三个阶段的协同搜索策略. 首先, 雇佣蜂阶段采用基于特征约简的搜索策略, 减少冗余特征, 提高搜索效率并加快收敛. 其次, 跟随蜂阶段在较小特征子集的基础上, 以降低分类错误率为目标进一步优化解的分类性能. 考虑到前两个阶段可能误删部分判别特征, 使相关解陷入停滞, 侦查蜂阶段利用存档中的历史信息引导搜索, 以发现潜在的互补特征组合并增强算法跳出停滞状态的能力. 三个阶段相互配合, 实现了对分类错误率和特征子集规模的协同优化, 使 CABC-MOFS 在 HV 指标上表现出优势.

与已有基于人工蜂群的多目标特征选择算法相比, CABC-MOFS 针对高维特征空间设计了更具方向性的搜索机制. MOABC-FS 将人工蜂群算法与 Pareto 优化机制相结合, 实现了分类性能与特征子集规模的同时优化, 但仍主要依赖传统蜂群更新机制, 对特征冗余的主动处理能力有限. FMABC-FS 通过样本缩减策略降低分类器评价成本, 并改进蜂群搜索过程以提高搜索效率, 但在利用个体差异信息引导特征子集搜索方面仍有提升空间. 相比之下, CABC-MOFS 以特征子集规模为竞争依据, 通过胜者与败者之间的差异化学习主动推动特征约简, 并结合错误率优化策略提升分类性能, 同时利用外部存档中的历史优秀解增强搜索多样性. 因此, 该算法能够在缩小特征子集规模的同时保持较高的分类性能, 在高维多目标特征选择任务中展现出更强的优化能力.

为直观展示 CABC-MOFS 在分类错误率和特征子集规模上的表现, 图 1 给出了各算法在 6 个代表性数据集上获得的 Pareto 前沿分布, 其中上排为 ionosphere、gse72526 和 colon, 下排为 Leukemia1、DLBCL 和 ALLAML. 由图可见, 在 ionosphere 数据集上, CABC-MOFS 虽然能够获得较低的分类错误率, 但部分解需要选择相对较多的特征; 而在 gse72526 和 colon 数据集上, 其大部分解集中于较低的特征选择比例区域, 并保持了较低的分类错误率. 在下排的高维数据集上, CABC-MOFS 获得的非支配解主要分布在图中的左下区域, 表明其能够以较少的特征取得较低的分类错误率. 这说明本文提出的竞争机制能够有效引导种群搜索更具判别能力的特征组合, 在显著降低特征维度的同时保持良好的分类性能. 综合来看, PF 分布结果与 HV 指标的

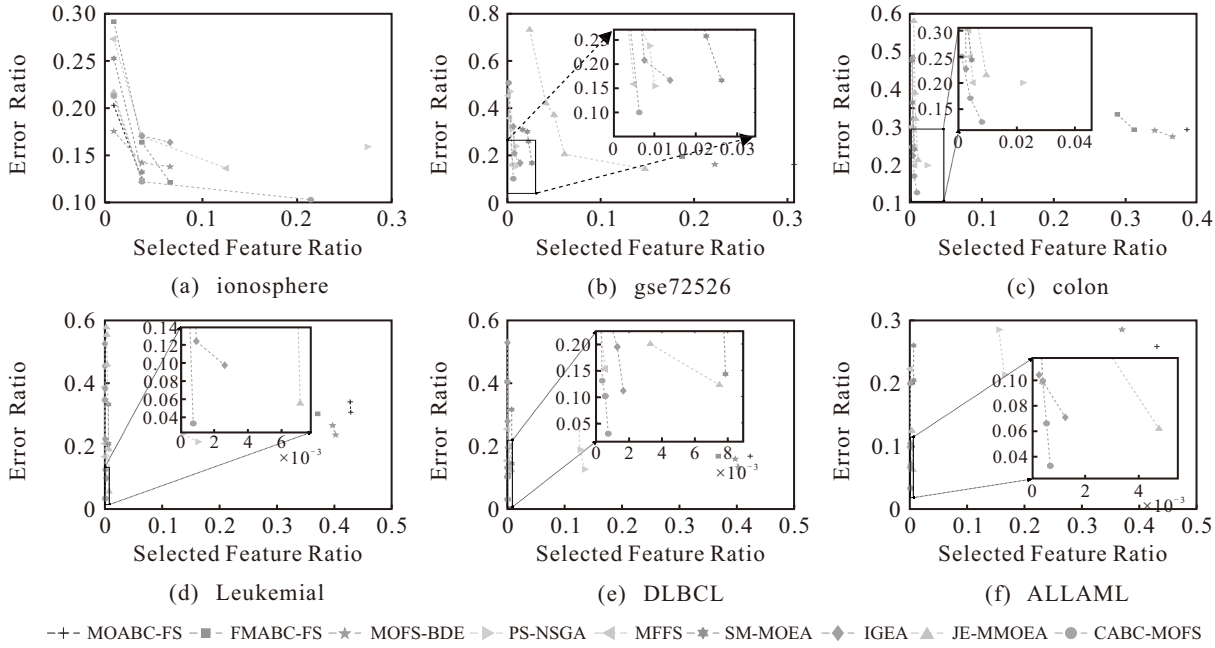

图 1 CABC-MOFS 与对比算法在 6 个代表性数据集上的 Pareto 前沿分布

分析结论基本一致, 进一步体现了 CABC-MOFS 在收敛性和多样性方面的竞争力.

3.2.2 从特征选择层面分析

前述多目标层面的分析分别从收敛性和多样性两个方面, 对两个优化目标进行综合评估. 而在实际应用场景中, 决策者更希望同时获取低分类错误率与精简的特征子集. 为此, 本节分别从分类错误率、特征子集规模两个维度, 对各算法性能开展单独分析与评估.

表 3 列出了各算法获得的最低分类错误率(MCER) 实验结果. 总体来看, CABC-MOFS 在 6 个数据集上取得最低分类错误率, 平均排名为 2.58, 综合表现位居首位. Wilcoxon 秩和检验结果进一步表明, CABC-MOFS 在大部分数据集上的分类性能显著优于其他对比算法. 在低维数据集中, 冗余特征通常较少, 特征的增加或删除可能对模型的泛化性能产生较大影响. 从表 3 前六个低维数据集的结果可以看出, 各算法的 MCER 差异整体较小, 说明不同算法在低维场景下均较容易获得分类性能较好的特征子集. 对于 CABC-MOFS 而言, 基于特征约简的搜索策略虽然能够减少冗余特征, 但低维数据集中可供删除的冗余特征较少, 较强的约简操作可能增加误删关键特征的风险, 因此其分类性能优势并不明显. 随着数据维度增加, 冗余特征逐渐增多, 算法的降维能力对分类性能的影响也更加突出. 由于通用型 MOFS 算法缺少针对高维特征空间的有效约简机制, 进化过程中容易保留或积累冗余特征, 从而削弱特征子集的泛化能力. 由表 3 可知, MOABC-FS、FMABC-FS 和 MOFS-BDE 等对比算法在高维数据集上的分类错误率通常高于 CABC-MOFS. 相比之

表 3 CABC-MOFS 与 8 种对比算法在 12 个数据集上的 MCER 值

数据集	MOABC-FS	FMABC-FS	MOFS-BDE	PS-NSGA	MFFS	SM-MOEA	IGEA	JE-MMOEA	CABC-MOFS
ionosphere	0.107 3(≈)	0.110 5(≈)	0.121 8(+)	0.175 0(+)	0.124 8(+)	0.124 5(+)	0.124 9(+)	0.111 9(≈)	**0.105 9**
Libras	0.268 2(≈)	0.265 7(≈)	**0.254 8(≈)**	0.285 4(≈)	0.266 5(≈)	0.271 2(≈)	0.262 3(≈)	0.274 9(≈)	0.269 0
MUSK1	0.134 7(≈)	0.138 0(≈)	**0.120 9(−)**	0.148 9(≈)	0.154 2(≈)	0.164 1(+)	0.121 4(−)	0.143 3(≈)	0.147 5
gse72526	0.139 8(≈)	0.148 4(≈)	0.134 0(+)	0.135 0(≈)	0.144 4(+)	0.170 9(+)	0.150 7(+)	0.132 3(≈)	**0.108 0**
Yale	0.385 8(≈)	0.370 4(≈)	**0.363 2(≈)**	0.373 4(≈)	0.410 9(≈)	0.408 7(≈)	0.363 6(≈)	0.388 7(≈)	0.388 5
colon	0.270 1(+)	0.271 2(+)	0.275 0(+)	0.189 5(+)	0.175 6(≈)	0.224 4(+)	0.191 0(≈)	0.180 8(+)	**0.143 6**
SRBCT	0.132 6(+)	0.140 5(+)	0.126 2(+)	0.024 5(≈)	0.074 3(+)	0.132 7(+)	0.040 9(+)	0.023 2(≈)	**0.022 6**
warpAR10P	0.490 4(+)	0.473 9(+)	0.476 7(+)	0.338 0(+)	0.288 1(≈)	0.354 5(+)	**0.283 3(≈)**	0.465 3(+)	0.284 0
GLIOMA	0.248 4(≈)	0.249 6(≈)	0.233 7(≈)	**0.198 2(≈)**	0.212 5(≈)	0.285 5(+)	0.221 0(≈)	0.237 3(≈)	0.217 0
Leukemia1	0.251 9(+)	0.251 3(+)	0.214 4(+)	0.180 1(+)	0.116 3(+)	0.168 7(+)	0.087 4(≈)	**0.062 1(≈)**	0.069 8
DLBCL	0.135 1(+)	0.139 6(+)	0.139 4(+)	0.103 1(+)	0.114 6(+)	0.114 8(+)	0.092 1(+)	0.095 3(+)	**0.063 9**
ALLAML	0.224 1(+)	0.237 8(+)	0.214 4(+)	0.187 4(+)	0.096 9(+)	0.162 7(+)	0.057 7(≈)	0.048 8(≈)	**0.043 7**
+/≈/−	6/6/0	6/6/0	8/3/1	6/6/0	6/6/0	10/2/0	4/7/1	3/9/0	−
排名	6.17	6.50	5.08	5.08	4.92	6.92	3.67	4.08	**2.58**

下, CABC-MOFS 通过特征约简、错误率优化和存档信息引导等策略协同搜索, 在减少冗余特征的同时尽可能保留具有判别能力的关键特征, 因此在高维数据集上表现出更好的分类性能.

为了进一步比较各算法的降维能力, 表 4 统计了各算法所得特征子集的大小. CABC-MOFS 在 7 个数据集上筛选出的特征数量最少, 平均排名值为 1.83, 位居首位. 这一结果与本文设计的基于特征约简的雇佣蜂搜索策略相契合. 该策略在搜索过程中持续施加降维压力, 引导败者向特征数量更少的优质个体学习, 从而获得更精简的特征子集. 尤其在 GLIOMA、Leukemia1、DLBCL 和 ALLAML 等高维数据集上, CABC-MOFS 选取的特征数量远少于大多数对比算法, 验证了该策略在高维场景下的有效性. 相比之下, MOABC-FS、FMABC-FS、MOFS-BDE 和 PS-NSGA 在高维数据集上选取的特征数量明显较多, 表明其降维能力相对较弱, 容易保留较多冗余特征. 这也在一定程度上解释了这些算法在表 3 的高维数据集上分类错误率相对较高的现象.

综合最低分类错误率与特征数量两项评价指标可知, CABC-MOFS 能够在分类性能与特征精简程度之间实现更好的平衡, 验证了本文所提竞争策略应用于多目标特征选择问题的有效性.

3.2.3　从计算效率层面分析

为进一步评估算法的运行效率, 表 5 统计了各算法在 12 个数据集上的运行时间. 可以看出, CABC-MOFS 在 6 个高维数据集上的运行时间最短, 并获得了最优的平均排名. 算法运行时间通常受到特征子集规模和搜索策略复杂度两个因素的共同影响: 一方面, 较小的特征子集能够降低分类器训练与测试过程中的评估开销; 另一方面, 算法自身的搜索、更新和辅助操作也会带来不同程度的计算成本. 因此, CABC-MOFS 的运行效率优势需要从降维效果和搜索策略设计两个方面进行解释.

表 4　CABC-MOFS 与 8 种对比算法在 12 个数据集上的特征子集大小

数据集	MOABC-FS	FMABC-FS	MOFS-BDE	PS-NSGA	MFFS	SM-MOEA	IGEA	JE-MMOEA	CABC-MOFS
ionosphere	3.166 7(≈)	2.833 3(≈)	3.133 3(≈)	8.400 0(+)	3.200 0(≈)	**2.800 0(≈)**	5.666 7(+)	3.066 7(≈)	3.433 3
Libras	10.900 0(≈)	11.833 3(+)	12.733 3(+)	17.900 0(+)	**8.766 7(≈)**	10.066 7(≈)	24.866 7(+)	11.200 0(≈)	10.000 0
MUSK1	34.300 0(+)	25.966 7(+)	29.566 7(+)	37.033 3(+)	16.400 0(≈)	**14.200 0(≈)**	58.933 3(+)	21.166 7(+)	15.033 3
gse72526	255.400 0(+)	165.400 0(+)	196.566 7(+)	9.533 3(≈)	**6.000 0(≈)**	15.233 3(+)	9.900 0(+)	62.400 0(+)	6.200 0
Yale	371.633 3(+)	308.266 7(+)	331.366 7(+)	42.866 7(+)	12.766 7(≈)	21.166 7(+)	27.533 3(+)	46.500 0(+)	**11.366 7**
colon	770.666 7(+)	592.633 3(+)	658.400 0(+)	49.933 3(+)	**5.600 0(−)**	11.166 7(+)	10.033 3(≈)	20.200 0(+)	6.900 0
SRBCT	913.133 3(+)	747.700 0(+)	787.033 3(+)	103.400 0(+)	9.433 3(≈)	12.966 7(+)	11.766 7(+)	36.566 7(+)	**6.000 0**
warpAR10P	970.500 0(+)	820.166 7(+)	882.666 7(+)	111.900 0(+)	9.700 0(≈)	23.533 3(+)	18.166 7(+)	58.533 3(+)	**7.233 3**
GLIOMA	1858.266 7(+)	1574.900 0(+)	1634.866 7(+)	542.833 3(+)	11.900 0(≈)	29.200 0(+)	12.166 7(+)	66.566 7(+)	**6.300 0**
Leukemia1	2292.700 0(+)	2021.133 3(+)	2079.366 7(+)	827.100 0(+)	11.866 7(≈)	46.900 0(+)	10.333 3(+)	49.033 3(+)	**5.133 3**
DLBCL	2337.900 0(+)	2030.333 3(+)	2114.800 0(+)	816.000 0(+)	18.633 3(≈)	59.866 7(+)	9.166 7(+)	52.866 7(+)	**5.166 7**
ALLAML	3124.300 0(+)	2740.433 3(+)	2841.533 3(+)	1230.266 7(+)	21.833 3(≈)	54.833 3(+)	8.900 0(+)	53.200 0(+)	**5.700 0**
+/≈/−	10/2/0	11/1/0	11/1/0	11/1/0	0/11/1	9/3/0	11/1/0	10/2/0	−
排名	8.08	6.33	7.42	6.25	2.42	3.58	4.33	4.75	**1.83**

表 5　CABC-MOFS 与 8 种对比算法在 12 个数据集上的运行时间 (秒)

数据集	MOABC-FS	FMABC-FS	MOFS-BDE	PS-NSGA	MFFS	SM-MOEA	IGEA	JE-MMOEA	CABC-MOFS
ionosphere	3.957 8(−)	4.479 1(−)	14.251 4(+)	30.225 8(+)	7.190 0(+)	**3.320 0(−)**	7.972 2(+)	29.718 9(+)	6.075 6
Libras	4.240 0(−)	7.452 5(+)	15.802 8(+)	26.442 0(+)	6.554 8(+)	**4.167 2(−)**	8.572 9(+)	28.232 1(+)	5.505 8
MUSK1	8.233 6(−)	11.820 1(+)	24.623 0(+)	44.564 1(+)	10.020 5(+)	**7.587 7(−)**	15.593 0(+)	37.078 9(+)	9.391 3
gse72526	**1.630 4(−)**	5.905 8(+)	26.719 5(+)	7.869 1(+)	4.698 0(+)	4.443 3(+)	5.532 8(+)	24.437 6(+)	2.390 0
Yale	5.942 3(+)	13.714 2(+)	39.097 0(+)	17.553 4(+)	5.687 7(+)	6.934 7(+)	8.376 3(+)	28.256 1(+)	**3.616 2**
colon	**3.905 2(−)**	11.346 1(+)	52.752 5(+)	14.215 0(+)	8.612 0(+)	14.802 9(+)	8.175 2(+)	33.525 8(+)	4.435 7
SRBCT	4.783 6(+)	14.544 9(+)	60.860 0(+)	17.468 6(+)	8.523 8(+)	16.956 4(+)	7.663 3(+)	33.592 0(+)	**3.792 8**
warpAR10P	8.427 3(+)	31.402 0(+)	71.449 0(+)	25.784 6(+)	9.550 5(+)	19.514 0(+)	10.236 2(+)	35.816 4(+)	**4.232 9**
GLIOMA	**4.786 4(−)**	16.732 2(+)	114.853 7(+)	24.303 4(+)	13.904 4(+)	53.800 5(+)	11.882 3(+)	44.303 1(+)	5.215 9
Leukemia1	9.848 8(+)	29.127 7(+)	138.776 0(+)	36.416 1(+)	17.640 8(+)	71.305 3(+)	17.217 5(+)	53.291 3(+)	**7.511 7**
DLBCL	14.834 6(+)	26.224 5(+)	150.759 0(+)	38.694 5(+)	17.779 6(+)	83.979 6(+)	16.237 5(+)	53.698 8(+)	**7.514 8**
ALLAML	20.002 3(+)	31.058 9(+)	186.970 1(+)	43.322 3(+)	21.352 7(+)	125.271 4(+)	18.603 0(+)	58.671 0(+)	**8.137 7**
+/≈/−	6/0/6	11/0/1	12/0/0	12/0/0	12/0/0	9/0/3	12/0/0	12/0/0	−
排名	1.92	5.17	8.50	6.92	3.83	5.00	4.08	7.75	1.83

首先, 从特征子集规模来看, MFFS、SM-MOEA 和 IGEA 所选特征数相对较少, 其运行时间也相对较短, 说明特征子集规模是影响计算效率的重要因素. 随着原始特征维度显著增加, 尤其是在 GLIOMA、Leukemia1、DLBCL 和 ALLAML 等高维数据集上, 降维效果能够更明显地转化为运行时间优势.

其次, 从搜索策略复杂度来看, MOABC-FS、FMABC-FS 和 CABC-MOFS 均基于 ABC 框架, 整体搜索流程相对简洁, 因此策略层面的额外计算开销较低. 这也是三种基于 ABC 的 MOFS 算法在多数数据集上均能保持较短运行时间的重要原因. CABC-MOFS 采用基于竞争的差异化搜索策略, 竞争划分过程仅需对个体进行简单比较即可完成胜者与败者的划分, 运算成本较低; 雇佣蜂和跟随蜂分别向对应的参考个体学习并执行差异化搜索, 因此所提出策略并未引入过多额外计算开销. 同时, 得益于前述特征约简作用, CABC-MOFS 能够获得更小的特征子集, 进一步降低适应度评估开销. 因此, 相比于 MOABC-FS 和 FMABC-FS, CABC-MOFS 在保持 ABC 框架简洁性的同时, 通过更有效的降维策略进一步提升了运行效率.

综合来看, CABC-MOFS 的效率优势既来源于高效的算法策略, 也得益于特征约简机制带来的评估开销降低, 因此能够在高维特征选择任务中表现出更高的计算效率.

3.3 策略有效性分析

CABC-MOFS 算法包含三个搜索阶段: 基于特征约简的雇佣蜂搜索、基于错误率优化的跟随蜂搜索和基于存档引导的侦查蜂搜索. 其中前两个阶段融入了基于竞争的差异化搜索策略, 记为 S1, 第三阶段的基于存档引导的侦查蜂搜索策略记为 S2. 为了验证 CABC-MOFS 中各改进策略的有效性, 本文构建了三类变体对比算法:

1. **MOABC**: 基础多目标人工蜂群算法, 不包含任何改进策略.

2. **MOABC+S1**: 仅在基础算法 MOABC 中加入策略 S1, 侦查蜂阶段沿用标准人工蜂群算法的随机重启方式.

3. **MOABC+S2**: 仅在基础算法 MOABC 中加入策略 S2, 雇佣蜂与跟随蜂阶段采用标准人工蜂群算法的搜索方式.

4. **MOABC+S1+S2**: 融合 S1 和 S2 两项改进策略, 即为本文所提 CABC-MOFS 算法

表 6 列出了各变体算法在 12 个数据集上的 HV 值. 通过比较 MOABC 与 MOABC+S1, 可以验证策略 S1 的有效性. 引入策略 S1 后, 算法平均排名值由 3.42 提升至 2.00, MOABC+S1 在大部分数据集上的 HV 值优于 MOABC, 且在 colon、SRBCT、warpAR10P、Leukemia1 等高维数据集上 HV 值有大幅提升, 表明基于竞争的差异化搜索策略 S1 可有效提升算法性能, 且在高维特征空间中的改进效果更为明显.

通过比较 MOABC 与 MOABC+S2, 可以验证策略 S2 的有效性. 引入策略 S2 后, 算法平均排名值由 3.42 提升至 3.08, 且 MOABC+S2 在 12 个数据集中的 8 个数据集上 HV 值优于 MOABC, 表明基于存档引导的侦查蜂搜索策略 S2 能够有效维持种群多样性, 避免算法陷入局部最优.

对比各变体算法与 CABC-MOFS 的表现, 可进一步验证两种策略的协同增效作用. 同时引入 S1 与 S2 后, CABC-MOFS 平均排名值提升至 1.50,

表 6 CABC-MOFS 与变体算法在 12 个数据集上的 HV 值

数据集	MOABC	MOABC+S1	MOABC+S2	CABC-MOFS
ionosphere	0.8687±0.0264	0.8756±0.0278	0.8709±0.0246	**0.8762±0.0250**
Libras	0.7252±0.0378	**0.7373±0.0315**	0.7230±0.0338	0.7337±0.0337
MUSK1	0.8527±0.0218	0.8489±0.0253	0.8484±0.0275	**0.8556±0.0172**
gse72526	0.8625±0.0560	0.8568±0.0482	0.8583±0.0471	**0.8992±0.0479**
Yale	0.6521±0.0438	0.6556±0.0537	**0.6574±0.0669**	0.6452±0.0578
colon	0.7498±0.0663	0.8366±0.0784	0.7874±0.0709	**0.8687±0.0548**
SRBCT	0.8929±0.0740	0.9756±0.0302	0.9239±0.0420	**0.9785±0.0248**
warpAR10P	0.5890±0.0648	**0.7460±0.0547**	0.6753±0.0636	0.7413±0.0626
GLIOMA	0.7926±0.0733	0.7949±0.0955	0.7903±0.0807	**0.8023±0.0801**
Leukemia1	0.8047±0.0921	0.9328±0.0807	0.8832±0.0867	**0.9363±0.0861**
DLBCL	0.8806±0.0753	**0.9428±0.0623**	0.9170±0.0840	0.9416±0.0735
ALLAML	0.8615±0.0631	0.9411±0.0567	0.9288±0.0642	**0.9601±0.0487**
排名	3.42	2.00	3.08	**1.50**

整体性能最优. 相较于仅引入单一策略的 MOABC+S1 和 MOABC+S2, CABC-MOFS 在 colon、SRBCT、Leukemia1、ALLAML 等高维数据集上的优势更为突出, 表明两种策略具有良好的互补性, 协同应用能够实现比单一策略更显著的性能提升.

综上所述, 两种策略均能独立地改善算法性能, 基于竞争的差异化搜索策略是 CABC-MOFS 取得优势的关键, 而基于存档引导的侦查蜂搜索策略则进一步增强了算法的稳定性, 二者的结合发挥了协同效应, 验证了 CABC-MOFS 中各改进策略的有效性.

4 结 论

针对高维数据多目标特征选择中搜索空间大、优化效率低、易陷入局部最优等问题, 本文提出了一种面向大规模多目标特征选择的竞争性人工蜂群算法 (CABC-MOFS). 该算法以特征子集规模为竞争依据, 将种群划分为胜者与败者, 分别设计基于特征约简的雇佣蜂搜索与基于错误率优化的跟随蜂搜索, 实现差异化协同进化. 此外, 引入基于存档引导的侦查蜂搜索, 借助外部存档为停滞个体提供有方向性的重启引导, 有效维持种群多样性并避免早熟收敛.

在 12 个不同维度数据集上的对比实验结果表明, CABC-MOFS 在超体积 (HV)、最低分类错误率 (MCER)、特征子集规模及运行时间等指标上均优于 8 种主流对比算法. 策略有效性分析结果表明, 基于竞争的差异化搜索策略与基于存档引导的侦查蜂搜索策略均可独立提升算法性能, 二者协同作用可实现更优的性能. 尽管 CABC-MOFS 在高维特征选择任务中表现出良好的降维能力和优化性能, 但其特征约简策略具有较强的特征压缩倾向, 在冗余特征较少的数据集上可能存在误删有价值特征的风险. 未来将进一步设计自适应特征约简机制, 以更好地平衡特征精简性与分类性能, 并探索所提竞争机制在多任务特征选择及其他大规模多目标优化场景中的扩展应用.

参考文献 (References)

[1] 张晓冬, 燕洁晨, 孙家正. 动态扰动下云制造服务组合的区间多目标优化方法[J]. 控制与决策, 2026, 41(1): 19-30.
(Zhang X D, Yan J C, Sun J Z. Interval multi-objective optimization for cloud manufacturing service composition under dynamic disturbances[J]. Control and Decision, 2026, 41(1): 19-30.)

[2] 李二超, 闫世韬. 多源知识自适应融合预测的动态多目标进化优化方法[J]. 控制与决策, 2026, 41(9): 2370-2388.
(Li E C, Yan S T. Dynamic multi-objective evolutionary optimization method based on adaptive fusion prediction of multi-source knowledge[J]. Control and Decision, 2026, 41(9): 2370-2388.)

[3] Tian Y, Si L C, Zhang X Y, et al. Evolutionary large-scale multi-objective optimization: A survey[J]. ACM Computing Surveys, 2022, 54(8): 1-34.

[4] Li J D, Cheng K W, Wang S H, et al. Feature selection: A data perspective[J]. ACM Computing Surveys, 2018, 50(6): 1-45.

[5] 王霞, 张珊, 王勇, 等. 基于评分机制的类贪心森林优化特征选择算法[J]. 控制与决策, 2025, 40(2): 517-527.
(Wang X, Zhang S, Wang Y, et al. Feature selection using greedy-like forest optimization algorithm based on scoring mechanism[J]. Control and Decision, 2025, 40(2): 517-527.)

[6] Cheng F, Cui J J, Wang Q J, et al. A variable granularity search-based multiobjective feature selection algorithm for high-dimensional data classification[J]. IEEE Transactions on Evolutionary Computation, 2023, 27(2): 266-280.

[7] Yu K J, Sun S R, Liang J, et al. A bidirectional dynamic grouping multi-objective evolutionary algorithm for feature selection on high-dimensional classification[J]. Information Sciences, 2023, 648: 119619.

[8] Hall M A. Correlation-based feature selection for machine learning[D]. Hamilton: The University of Waikato, 1999.

[9] Kononenko I, Šimec E, Robnik-Šikonja M. Overcoming the myopia of inductive learning algorithms with RELIEFF[J]. Applied Intelligence, 1997, 7(1): 39-55.

[10] Xia M M, Li K X, Wan J C, et al. An instance selection assisted evolutionary method for high-dimensional feature selection[J]. IEEE Transactions on Evolutionary Computation, 2026, 30(3): 1225-1239.

[11] Ahadzadeh B, Abdar M, Safara F, et al. SFE: A simple, fast, and efficient feature selection algorithm for high-dimensional data[J]. IEEE Transactions on Evolutionary Computation, 2023, 27(6): 1896-1911.

[12] Deng S B, Li Y L, Wang J K, et al. A feature-thresholds guided genetic algorithm based on a multi-objective feature scoring method for high-dimensional feature selection[J]. Applied Soft Computing, 2023, 148: 110765.

[13] Yue X Z, Liao Y H, Peng H, et al. A high-dimensional feature selection algorithm via fast dimensionality reduction and multi-objective differential evolution[J]. Swarm and Evolutionary Computation, 2025, 94: 101899.

[14] Hancer E, Xue B, Zhang M J, et al. Pareto front feature selection based on artificial bee colony optimization[J]. Information Sciences, 2018, 422: 462-479.

[15] Wang X H, Zhang Y, Sun X Y, et al. Multi-objective feature selection based on artificial bee colony: An acceleration approach with variable sample size[J]. Applied Soft Computing, 2020, 88: 106041.

[16] Zhang Y, Gong D W, Gao X Z, et al. Binary differential

evolution with self-learning for multi-objective feature selection[J]. Information Sciences, 2020, 507: 67-85.

[17] Zhou Y, Zhang W J, Kang J H, et al. A problem-specific non-dominated sorting genetic algorithm for supervised feature selection[J]. Information Sciences, 2021, 547: 841-859.

[18] Cheng F, Chu F X, Xu Y, et al. A steering-matrix-based multiobjective evolutionary algorithm for high-dimensional feature selection[J]. IEEE Transactions on Cybernetics, 2022, 52(9): 9695-9708.

[19] Jiao R W, Xue B, Zhang M J. Benefiting from single-objective feature selection to multiobjective feature selection: A multiform approach[J]. IEEE Transactions on Cybernetics, 2023, 53(12): 7773-7786.

[20] Zhang B H, Wang Z Q, Li H T, et al. Information gain-based multi-objective evolutionary algorithm for feature selection[J]. Information Sciences, 2024, 677: 120901.

[21] Liang J, Yang J T, Yue C T, et al. A joint-encoding evolutionary algorithm for multimodal multiobjective feature selection in classification[J]. IEEE Transactions on Evolutionary Computation, 2025, 29(6): 2834-2848.

[22] Tian Y, Zheng X T, Zhang X Y, et al. Efficient large-scale multiobjective optimization based on a competitive swarm optimizer[J]. IEEE Transactions on Cybernetics, 2020, 50(8): 3696-3708.

[23] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.

[24] Kelly M, Longjohn R, Nottingham K. The UCI machine learning repository[EB/OL]. (2026-03-20), https://archive.ics.uci.edu.

[25] Cover T, Hart P. Nearest neighbor pattern classification[J]. IEEE Transactions on Information Theory, 1967, 13(1): 21-27.

[26] Saadatmand H, Akbarzadeh-T M R. Many-objective jaccard-based evolutionary feature selection for high-dimensional imbalanced data classification[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46(12): 8820-8835.

[27] Li L J, Xuan M L, Lin Q Z, et al. An evolutionary multitasking algorithm with multiple filtering for high-dimensional feature selection[J]. IEEE Transactions on Evolutionary Computation, 2023, 27(4): 802-816.

作者简介

王晖 (1982−), 男, 教授, 博士, 主要研究方向为进化计算、多目标优化等, E-mail: huiwang@juwp.edu.cn;

钱佳瑶 (2001−), 女, 硕士生, 主要研究方向为群智能、多目标优化, E-mail: jyqian@juwp.edu.cn;

肖人彬 (1965−), 男, 教授, 博士, 主要研究方向为群体智能、大规模个性化定制, E-mail: rbxiao@hust.edu.cn;

王文君 (1982−), 女, 副教授, 硕士, 主要研究方向为管理决策与优化, E-mail: wangwenjun@juwp.edu.cn;

赵嘉 (1981−), 男, 教授, 博士, 主要研究方向为数据挖掘、智能计算, E-mail: zhaojia@juwp.edu.cn;

周新宇 (1987−), 男, 副教授, 博士, 主要研究方向为进化计算及应用, E-mail: xyzhou@jxnu.edu.cn.