

DOI:10.13232/j.cnki.jnju.2023.05.007

## 基于自适应环境因子熵权决策的多目标特征选择

李 涛<sup>1,3\*</sup>, 李佳霖<sup>2</sup>, 阮 宁<sup>2</sup>, 徐久成<sup>1,3</sup>

(1. 河南师范大学计算机与信息工程学院, 新乡, 453007; 2. 河南师范大学软件学院, 新乡, 453007;  
3. “智慧商务与物联网技术”河南省工程实验室, 新乡, 453007)

**摘 要:** 多目标进化算法在特征选择方面有显著的优势, 但其求解高维数据最优特征子集的性能依然较差, 且从获得的 Pareto 解集中选择合理最优解仍是一个挑战性的问题. 为了解决该问题, 提出一种基于自适应环境因子熵权决策的多目标特征选择算法. 首先, 通过设计环境因子来自适应识别关键特征, 优化候选特征子空间; 其次, 将环境因子嵌入改进的交叉算子和变异算子, 实现全局最优特征子集的自适应搜索; 最后, 利用关联环境因子的熵权决策策略, 从获得的 Pareto 解集中选出最优解. 实验表明, 与现有的五种多目标特征选择算法相比, 提出的算法具有更高的分类精度, 并能准确地获取全局最优解, 验证了该算法的有效性.

**关键词:** 多目标优化, 特征选择, 熵权决策, 自适应算法

中图分类号: TP181

文献标志码: A

## Adaptive environmental factor entropy weight decision-making-based multi-objective feature selection

Li Tao<sup>1,3\*</sup>, Li Jialin<sup>2</sup>, Ruan Ning<sup>2</sup>, Xu Jiucheng<sup>1,3</sup>

(1. College of Computer and Information Engineering, Henan Normal University, Xinxiang, 453007, China;

2. College of Software, Henan Normal University, Xinxiang, 453007, China;

3. Engineering Lab of Henan Province for Intelligence Business & Internet of Things, Xinxiang, 453007, China)

**Abstract:** Multi-objective evolutionary algorithms have significant advantages in feature selection, but their performance in solving the optimal feature subset of high-dimensional data is still poor, and selecting a reasonable optimal solution from the obtained Pareto solution set remains a challenging issue. To solve this problem, this paper proposes a multi-objective feature selection algorithm based on adaptive environmental factor entropy weight decision-making. The advantages of the proposed algorithm are as follows. Firstly, key features are adaptively identified by designing environmental factors to optimize candidate feature subspaces. Secondly, environment factors are embedded into improved crossover and mutation operators to achieve adaptive search for the globally optimal feature subset. Finally, using an entropy weight decision-making strategy that correlates environmental factors, the optimal solution is selected from the obtained Pareto solution set. Experiments show that the proposed algorithm has higher classification accuracy and accurately obtains the global optimal solution compared to the existing five multi-objective feature selection algorithms, verifying the effectiveness of the proposed algorithm.

**Key words:** multi-objective optimization, feature selection, entropy weight decision-making, adaptive algorithm

随着大数据技术的迅速发展, 日益增加的海量高维数据<sup>[1-3]</sup>对传统数据分析与处理方法提出

了新的挑战. 特征选择<sup>[4-6]</sup>是一种重要的数据降维技术, 通过剔除数据集中的无关或冗余特征, 形

基金项目: 国家自然科学基金(61976082), 河南省高等学校重点科研项目(22B520013)

收稿日期: 2023-06-19

\* 通讯联系人, E-mail: litao0116@163.com

成一个结构上更紧凑的特征子集,从而减少学习模型的训练时间并提高泛化能力.特征选择方法在数据挖掘、机器学习和生物信息学等领域得到广泛应用,取得了较好的成果.

常见的特征选择方法可分为过滤式方法<sup>[7]</sup>、封装式方法<sup>[8-10]</sup>和嵌入式方法<sup>[11]</sup>三大类.过滤式方法在未训练学习器的情况下评估特征子集,虽然其扩展性强且计算成本低,但由于缺少学习器的训练学习导致分类性能较差.封装式方法结合具体学习器的误差来衡量候选特征子集,取得了较好的分类准确性,但其需要计算特征间的交互关系,计算开销较大.嵌入式的方法将特征选择与学习器训练相结合,在学习算法的执行过程中完成最优特征子集的选择.虽然上述三类特征选择方法已有大量探索,但仍存在特征子集陷入局部最优、关键特征选择不准确等问题.假设一个数据集的特征数为 $n$ ,则存在 $2^n-1$ (除去空集)种候选特征子集,采用穷举搜索高维数据特征显然不可行.针对高维数据特征选择亟须解决的问题,一是由于维度高导致搜索空间巨大,需要减少评估候选特征子集的计算量;二是特征间存在复杂的互转化关系,需要准确识别与选择关键特征来构造全局最优特征子集.

近年来,进化计算方法(Evolutionary Computing Algorithms, EAs)因其强大的全局寻优能力引起了研究者的广泛关注,基于进化计算的特征选择方法可以提高全局最优解的搜索效率,根据优化目标的不同可以分为基于单目标的特征选择优化方法和基于多目标的特征选择优化方法.虽然基于多目标优化的特征选择方法一次运行就能获得多个解,但从获得的 Pareto 解集中正确选择合理的唯一特征子集仍是难点.比较经典的多目标优化算法是 NSGA-II<sup>[12-14]</sup>,主要采用非支配排序与拥挤距离的思想来评估两个优化目标上的个体,最后构成 Pareto 解集.近期,针对特征选择问题,学者们也提出众多有效的基于进化计算的多目标特征选择方法. Xu et al<sup>[15]</sup>提出 NSGA-II 与 MOPSO 混合方法进行多目标特征选择,虽然探索性强,收敛性强,还能避免陷入局部最优的困境,但筛选的特征子集仍包含大量的冗余特征,且时间复杂度较高. Liu et al<sup>[16]</sup>提出基于高斯混合模

型和改进 NSGA-II 算法的多目标优化方法,利用平均距离将整个种群划分为若干子种群,再对各个子种群执行选择、交叉和变异操作.该方法能有效保持 Pareto 最优解集的多样性,进一步提高算法的收敛性,但忽略了特征之间的复杂相关性,增加了剔除关键特征的可能性,折中解较差. Zhang et al<sup>[17]</sup>提出一种改进的 NSGA-II 算法,引入余弦相似性,使下一代的种群个体向偏好方向聚集,然后根据预期的特征首选项来调整解决方案的设置方向,以此评估优化算法的收敛性、解多样性和偏好一致性,但其没有针对获得的 Pareto 解集进行分析,无法获取最优个体.

上述基于 NSGA-II 的改进与优化方法一定程度上改善了特征子集的分类性能,但 NSGA-II 在处理高维特征选择问题时仍存在不足.一是在设计进化算子时没有考虑自适应性对特征子集选取的影响,多目标进化搜索的 Pareto 解集质量差.通常个体交叉和变异过程主要依赖个体适应度,会错误保留冗余特征或删除相关特征,导致特征子集的分类性能不佳.二是 Pareto 解集中缺少合理最优解选择策略.尽管多目标优化特征选择方法提供了优秀解决方案,但解决方案的选择往往依赖决策者的主观需求,缺少合理最优解的选择依据,所以选择的特征子集的可解释性较差.

为了解决上述问题,本文提出一种基于自适应环境因子熵权决策的多目标特征选择算法(Adaptive Environmental Factor Entropy Weight Decision-Making-Based Multi-Objective Feature Selection Algorithm, AMFS),主要框架如图 1 所示.首先,根据特征关系设计环境因子,依据特征的重要性对候选特征进行自适应赋值;随后,基于环境因子分别提出改进的自适应交叉算子和自适应变异算子,并生成新的种群个体作为候选特征子集;最后,给出基于环境因子的熵权决策来选取最优解.

## 1 相关工作

鉴于过滤式和封装式特征选择方法的优点,相关学者研究两者混合式特征选择<sup>[18]</sup>,并提出众多基于多目标优化的特征选择方法,在处理高维特征选择时表现良好.这主要是由于过滤式阶段

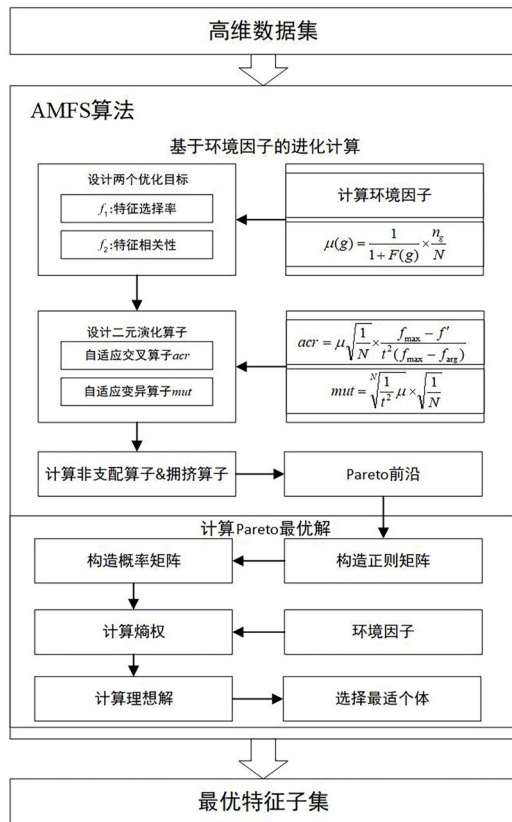


图 1 本文提出的 AMFS 算法的框架

Fig. 1 The framework of the proposed AMFS algorithm

能有效剔除大量无关特征及冗余特征,减少特征维度,缩小求解空间;封装式阶段结合具体优化算法,依据目标函数的优化结果来判定特征子集的选择,能获取高精度的特征子集。

Zhang et al<sup>[19]</sup> 提出自学习二差分进化算法,用基于概率差的二元变异算子引导个体快速定位潜在的最优区域,还提出一种新型净化搜索算子来提高位于最优区域精英个体的自学习能力。具有拥挤距离的高效非支配排序算子可以降低差分进化中算子选择的计算复杂度,大幅度缩短搜索时间,提升分类表现,但在高维小样本数据集的效果不理想,不具有普适性。Zhang et al<sup>[20]</sup> 提出基于成本的特征选择多目标粒子群优化算法,采用概率的编码技术和有效的混合算子,将拥挤距离、外部档案和 Pareto 支配关系应用于粒子群算法,明显缩减了特征子集的搜索空间,有效解决了约简变量过多导致的局部最优问题,但没有考虑最优全局解集的情况。孙哲人等<sup>[21]</sup> 提出一个基于多

样性分类和距离回归的进化算法,把种群中的所有解作为训练样本,并根据是否为最小相关解,把训练样本分类为正负样本,使模型学习到训练样本中含有的分类信息。此外,它采用 Kriging 作为局部回归代理模型,其选择种群中距离当前候选解最近的解作为训练样本,拟合训练样本与理想点的距离,然后通过 K-means 方法把候选解划分为  $\mu$  个簇,从每个簇中选择一个用于真实评估的候选解。该方法在低维度特征空间取得了不错的效果,但随着目标数量的增加,目标空间呈指数扩大,算法效果不理想。Zhou et al<sup>[22]</sup> 提出一种基于两级粒子协作的多目标优化特征选择策略来进行高维数据的特征选择,考虑了特征数量、分类错误率和距离测量三个目标,将二进制粒子群优化与两级粒子协作策略相结合,有效地减少了特征数量,还组合了随机生成的普通粒子和 ReliefF 过滤粒子来实现快速收敛。然而,该方法的平均分类精度效果不佳。Rashno et al<sup>[23]</sup> 提出一种新型的多目标粒子群优化特征选择方法,将特征向量解码为粒子并在二维优化空间中进行排名,对粒子进行排序,将优化空间分成主导粒子带与非主导粒子带,通过数学与实验详细解析了优化空间中均匀与非均匀分布对粒子模型的影响,将粒子秩与特征秩运用到迭代过程来更新粒子的速度与位置,找到多目标优化空间中最接近原点的最佳粒子。然而,该方法没有解决优势粒子与非优势粒子之间粒子距离的优化组合,所以得到的最佳粒子不一定是 Pareto 解集中的最优解。Wang et al<sup>[24]</sup> 提出基于样本还原策略和进化算法的多目标特征选择框架,该框架主要包含 K-means 聚类的差分选择方法和样品利用策略,还在框架中嵌入了粒子更新模型的改进人工蜂群算法。该方法虽然降低了计算成本,减少了进化过程中使用的样本量,缩短了进化迭代的时间,但其在高维特征空间的应用效果不佳。

## 2 基础知识

**2.1 NSGA-II 算法** NSGA-II 是经典的多目标进化算法,在处理现实应用领域中的多目标优化问题时表现良好。其采用 Pareto 支配关系和拥挤距离<sup>[25-27]</sup> 机制来更新种群,并依据多个优化目标

度量选择多个折中解来构成 Pareto 前沿.

下面对 NSGA-II 算法的重要概念和基本算子进行描述.

**2.1.1 Pareto 支配关系** 多目标优化问题<sup>[28-33]</sup>中 Pareto 优化解是最常用的优化概念,其定义如下.

**定义 1 Pareto 支配** 设两个决策变量  $x_1, x_2 \in \Omega$ .  $x_1$  Pareto 占优  $x_2$ , 记作  $x_1 > x_2$ , 当且仅当  $\forall i \in \{1, 2, \dots, n\}, f_i(x_1) \leq f_i(x_2) \wedge \exists j \in \{1, 2, \dots, k\}, f_j(x_1) < f_j(x_2)$ , 记作  $x_1 < x_2$ .

**定义 2 Pareto 最优解** 解  $x^* \in \Omega$  为解集  $\Omega$  的 Pareto 最优解, 当且仅当  $\{x | x < x^*, x \in \Omega\} = \Phi$  ( $\Phi$  为空集).

**定义 3 Pareto 最优解集** 对于给定的多目标优化问题, 设其定义域为  $\Omega$ , 则 Pareto 最优解集  $X$  定义为  $X^* = \{x \in \Omega | \neg \exists x' \in \Omega, f(x') < f(x)\}$ . Pareto 最优解集在函数空间上对应的集合称为 Pareto 前沿<sup>[34-36]</sup>, 记作  $PF$ .

## 2.1.2 基本算子

**2.1.2.1 非支配排序算子** 首先, 对种群中的每个解  $p$  计算支配  $p$  的解的数量  $n_p$  和被  $p$  支配的解的集合  $S_p$ . 由于第一层非支配层级上解的支配数记为 0, 对于每一个  $n_p=0$  的解, 访问  $S_p$  中的成员  $q$ , 并将其支配数减 1, 如果  $q$  的支配数为 0, 则将其放入一个列表  $Q$  中. 因此,  $Q$  中包含了属于第二层非支配层级的解. 接着, 对集合  $Q$  中的解重复这一过程, 直到找到第三层非支配层级. 这个过程一直持续到找到所有的非支配层级为止.

**2.1.2.2 拥挤度距离算子** 首先根据每个目标函数的适应度对种群中的个体进行升序排序. 然后, 对于每个目标函数, 其边界解被分配一个无限的距离值, 每个中间解也被分配一个距离值, 该距离值等于两个相邻解差的绝对值, 该过程在不同的目标函数中重复进行. 因此, 拥挤的距离值即为在每个目标上的距离之和.

**2.2 熵权 TOPSIS 决策** TOPSIS 根据优化目标与 Lable 值之间的接近程度进行排序, 具有可观、可靠的优点, 但也存在权值法受决策者主观影响的问题. 熵权 TOPSIS 将信息熵引入 TOPSIS, 利用信息熵计算得到 TOPSIS 的权重, 减小

了主观权重赋值带来的偏差. 下面给出熵权 TOPSIS 决策的相关定义.

**定义 4 正则化矩阵** 根据解的个数  $m$  与优化目标的个数  $n$  确定相应的决策矩阵  $X = (x_{ij})_{m \times n}$ .  $r_{ij}$  表示第  $i$  个方案的第  $j$  个指标,  $i = 1, 2, \dots, m, j = 1, 2, \dots, n$ . 对决策矩阵进行正则化处理得到正则化矩阵  $R = (r_{ij})_{m \times n}$ , 其中,

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}$$

**定义 5 熵值** 第  $j$  个指标的熵值:

$$e_{ij} = -\frac{1}{\ln(m)} \sum_{i=1}^m p_{ij} \ln(p_{ij})$$

其中,  $p_{ij} = \frac{r_{ij}}{\sum_{i=1}^m r_{ij}}$ , 是对  $r_{ij}$  进行规范化处理后的结果.

利用熵值计算第  $j$  个指标的权重:

$$w_j = \frac{(1 - e_j)}{\sum_{j=1}^n (1 - e_j)}$$

利用第  $j$  个指标的权重  $w_j$  对正则化结果  $r_{ij}$  进行加权, 得到加权正则化后的结果  $\lambda_{ij} = w_j r_{ij}$ .

**定义 6 理想解** 理想解分正理想解和负理想解, 正理想解  $s_j^+$  取各指标的最大值  $s_j^+ = \max\{\lambda_{ij}\}$ , 负理想解  $s_j^-$  取各指标的最小值  $s_j^- = \min\{\lambda_{ij}\}$ .

**定义 7 解距** 决策值与正负理想解之间的解距可采用欧式距离计算, 表示为:

$$\phi_i^+ = \sqrt{\sum_{j=1}^n (s_j^+ - \lambda_{ij})^2}$$

$$\phi_i^- = \sqrt{\sum_{j=1}^n (s_j^- - \lambda_{ij})^2}$$

**定义 8 相对贴近度** 相对贴近度  $\eta_i$  表示各待决策值与正理想解之间的贴近程度, 表示为:

$$\eta_i = \frac{\phi_i^-}{\phi_i^+ + \phi_i^-}$$

对每个解的相对贴近度, 按从大到小的降序排序, 选取具有相对贴近度最大值的解作为最适解.



### 3 基于自适应环境因子熵权决策的多目标特征选择

为了解决 Pareto 解集中缺少合理最优解选择策略的问题,提出一种基于环境因子的熵权决策,减少决策过程中决策者主观因素的影响,增强合理最优解的选择依据,丰富选择的特征子集的可解释性.针对多目标进化搜索到的 Pareto 解集质量差的问题,提出基于环境因子的自适应算子,将个体交叉和变异过程依赖的个体适应度作为自适应依据,删除冗余特征,保留相关特征,增强特征子集的分类性能.

**3.1 基于环境因子改进的自适应算子** 提出一种作用于环境因子下的交叉变异算子的自适应算法环境因子,其形式化如式(1)所示:

$$\mu(g) = \frac{1}{1 + F(g)} \times \frac{n_g}{N} \quad (1)$$

对个体  $g$  在迭代过程中加入环境因子来控制整体适应度,  $n_g$  为训练样本中个体  $g$  的出现数目,  $N$  为迭代次数,  $fre(g)$  表示个体  $g$  在选择、交叉、变异过程中出现的频率.  $F(g)$  是训练样本数据中的类型  $g$  与所有类型数据在全体数据中出现频率总和之比,如式(2)所示:

$$F(g) = \frac{fre(g)}{\sum_{i=1}^m fre(g_i)} \quad (2)$$

**3.1.1 环境因子下的交叉算子** 在进化迭代中,随着进化迭代次数  $t$  的增加,遗传可分为前期和后期.进化前期个体的多样性强,需要快速找出较优的个体;进化后期的个体平均适应度已高度靠近最佳适应度,较大的交叉率可能会带来收敛不稳定性.当个体适应度较集中或较分散时,视为进化个体进入高度集中或高度分散两种极端情况,此时应适当调整交叉率.为此,改进交叉率如式(3)所示:

$$acr = \mu \sqrt{\frac{1}{N} \frac{f_{\max} - f'}{t^2(f_{\max} - f_{\arg})}} \quad (3)$$

其中,  $t$  为迭代次数.随着迭代的进行,种群逐渐向种群中的最优适应度靠拢,交叉率随着迭代而逐渐减小.  $(f_{\max} - f')$  表示种群中个体最大适应度

和两个交叉个体中适应度较大的个体的适应度之差,该项表示交叉个体在种群中的离散程度,两者之差越大即表示交叉的两个个体的适应度越低,交叉率随之增大,增加了引入优质样本的概率.  $(f_{\max} - f_{\arg})$  表示种群中个体最大适应度和种群中所有个体的平均适应度之差,其与  $(f_{\max} - f')$  的比值表示交叉的两个个体的适应度与种群平均适应度的差距,差值越大,意味着这两个个体越接近边界位置,为了增加引入优质样本的概率,交叉率需要随之增大.越大的种群个数  $N$  意味着越丰富的种群适应度,即有越大的可能含有优质解,故交叉率随着种群数  $N$  的增大而减小.该环境下样本的集中程度  $\mu$  变大,意味着种群适应度收敛程度变高,交叉率会随之上升,增加引入优质解的概率.

**3.1.2 环境因子下的变异算子** 随着遗传迭代次数  $t$  的增加,较大的变异率引入的新样本的波动会影响遗传算法的收敛性,故变异率应随迭代次数增加给予适当改变.个体适应度较集中时,往往意味着平均适应度函数逼近一个极值,但无法确定其是否为最优解,需对变异率进行适当调整,使其不易陷入局部最优.因此,修正变异率如式(4)所示:

$$mut = \sqrt{\frac{1}{t^2} \mu} \times \sqrt{\frac{1}{N}} \quad (4)$$

与交叉率算子的设计过程相似,综合考虑种群个数、样本集中程度和迭代次数变化的影响,变异率随着  $N$  的增大而减小,随着  $\mu$  的增加而增加,随着迭代次数  $t$  的增加而减小.

**3.2 基于环境因子熵权的 Pareto 最适解选取** 本节提出基于环境因子赋权重,将 AMFS 得到的 Pareto 最适解集通过熵权 TOPSIS 分析得出在此环境下的最适解,其中,设计的目标函数  $f_1$  为特征选择率,目标函数  $f_2$  为特征相关系数.目标函数如式(5)所示:

$$\begin{cases} \min f_1 = \frac{|F_s|}{|F_0|} \\ \min f_2 = \left| \frac{\sigma_{F_s} \sigma_{F_l}}{COV(F_s, F_l)} \right| \end{cases} \quad (5)$$

其中,  $|F_s|$  为最终优化特征个数,  $|F_0|$  为原始特征个数,  $|F_r|$  为所选特征子集,  $|F_l|$  为数据集的标签.

特征子集相关性最大化与特征选择率最小化通常被视为特征子集的两个优化目标,值得注意的是,这两个优化目标在大多数场景下存在相互冲突的情况,得到的序列值通常无法判定最优情况,可以通过对序列个体引入环境因子 $\mu$ 并对该环境下的 Pareto 解进行信息熵权判断来决策最适解. 具体步骤如下:

Step 1. 通过 AMFS 得到的 Pareto 序列构成正则化矩阵:

$$X = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \vdots & \vdots \\ x_{n1} & x_{n2} \end{bmatrix}$$

$$\text{令 } z_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^n x_{ij}^2}}, \text{ 得出正则化矩阵:}$$

$$Z = \begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \\ \vdots & \vdots \\ z_{n1} & z_{n2} \end{bmatrix}$$

Step 2. 计算概率矩阵  $P$ ,  $P$  中的每一个元素  $p_{ij} = \frac{z_{ij}}{\sum_{i=1}^n z_{ij}}$ .

Step 3. 通过  $e_j = -\frac{1}{\ln n} \sum_{i=1}^n p_{ij} \ln(p_{ij})$  ( $j=1, 2$ ),  $d_j = 1 - e_j$ , 则熵权  $W_j = \frac{d_j}{\sum_{j=1}^m d_j}$  ( $j=1, 2$ ).

Step 4. 对所得的熵权附加环境因子 $\mu$ , 计算个体适应度  $\lambda_{ij} = W_j \mu_{ij}$ . 取正理想解时,  $s_j^+ = \max\{\lambda_{ij}\}$ ; 取负理想解时,  $s_j^- = \min\{\lambda_{ij}\}$ . 通过理想解解距计算相对贴近度:

$$\phi_i^+ = \sqrt{\sum_{j=1}^n (s_j^+ - \lambda_{ij})^2}$$

$$\phi_i^- = \sqrt{\sum_{j=1}^n (s_j^- - \lambda_{ij})^2}$$

$$\eta_i = \frac{\phi_i^-}{\phi_i^+ + \phi_i^-}$$

最后, 选取相对贴近度的最大值  $a_\mu = \max\{\eta_i\}$ , 则  $a_\mu$  对应的个体为所得最适解.

**3.3 所提算法伪代码描述** 基于以上分析, 下面

给出基于自适应环境因子熵权决策的多目标特征选择的伪代码.

**算法 基于自适应环境因子熵权决策的多目标特征选择 (AMFS)**

**Input:** 原始数据集为  $G_{N \times M}$ , 其中, 样本数为  $N$ , 特征维度为  $M$ , 种群大小  $Pop = N$ , 迭代次数  $T = 100$

**Output:** Pareto 前沿  $PF$ , 最适解  $a_\mu$

Generate &. Initialize;

1. 初始化  $GN \times M$
2. 根据式(1)计算环境因子  $\mu(g)$
3. 初始化种群  $PN \times M$
4. 根据式(3)计算交叉算子
5. 根据式(4)计算变异算子
6. 计算拥挤距离与非支配排序
7. 更新  $PF$
8. 通过 Step 1 构造正则化矩阵
9. 通过 Step 2 构造概率矩阵
10. 通过 Step 3 计算熵权
11. 通过 Step 4 计算最大相对贴近度获取最适解  $a_\mu$

设  $n$  和  $m$  分别为原始数据集中的样本数和特征数,  $N$  是总体的大小,  $M$  是目标的数量,  $T$  是迭代次数,  $n^*$  为 Pareto 解集个数. 第 1~3 行计算每个特征的环境适应度, 使两个优化目标能够有效地衡量候选解决方案, 其时间复杂度主要依赖于环境因子  $\mu(g)$  的计算, 时间复杂度为  $O(n \times m)$ . 第 4~6 行计算拥挤距离与非支配排序, 其时间复杂度为  $O(N \lg N \times M)$ . 第 8~11 行计算最适解, 其时间复杂度为  $O(n^* \times M)$ . 因此, 本文算法 AMFS 的时间复杂度为:

$$O(T \times (n \times m) \times (N \lg N \times M) + (n^* \times M))$$

即为  $O((n \times m)(N \lg N \times M))$ .

**3.4 所提算法的收敛性分析** 根据上述定义,  $F$  表示算法可以搜索到的个体空间,  $B(t)$  是算法第  $t$  次迭代的种群,  $A(t)$  为算法归档集,  $F^* = M_{\setminus}(F, <)$  为空间  $F$  的所有非支配解. 下面给出本文算法 AMFS 的收敛性分析.

**定义 9** 定义  $|A|$  表示集合  $A$  的元素个数, 定义  $\delta(A, B) = |A| - |A \cap B|$ . AMFS 算法以概率 1 收敛于 Pareto 最优解集, 当  $t \rightarrow \infty$  时,  $\delta(A(t), F^*) \rightarrow 0$  的概率为 1.

**命题 1** 令  $s \in F$  为可行解集  $F$  中的任一元素. 如果概率  $P(s \in B(t))$  不依赖于进化代数  $t$  且存在  $\epsilon (0 < \epsilon < 1)$  满足  $P(s \in B(t)) \geq \epsilon$ , 则当  $t \rightarrow \infty$  时,  $\delta(A(t), F^*) \rightarrow 0$  的概率为 1.

**证明** 因为

$$A(t+1) = M_f(B(t+1) \cup A(t), <)$$

所以,  $A(t)$  非减, 即一旦个体  $s \in F^*$  被  $B(t)$  搜索到, 则该个体会保存到  $A(t)$  中, 而且因为该个体为非支配解, 它将一直被保存在  $A(t)$  中. 随着  $F^*$  中的个体不断被搜索到并被保存到  $A(t)$  中, 原本在  $A(t)$  中的被支配个体将从  $A(t)$  中删除. 因为概率  $P(s \in B(t))$  不依赖于代数  $t$  且存在  $0 < \epsilon < 1$ , 满足  $P(s \in B(t)) \geq \epsilon$ , 则对于任一个体  $s \in F^*$ , 有:

$$P(s \in A(t)) = 1 - \prod_{i=1}^t (1 - P(s \in B(i))) \geq 1 - (1 - \epsilon)^t$$

而

$$\lim_{t \rightarrow \infty} P(s \in A(t)) \geq \lim_{t \rightarrow \infty} (1 - (1 - \epsilon)^t) = 1$$

当  $t \rightarrow \infty$  时,  $\forall s \in F^*, P(s \in A(t)) \rightarrow 1$ . 此时,  $A(t)$  中支配解的个体数趋近于 0, 亦即  $A(t) \cap F^* \rightarrow A(t)$ , 可知  $\lim_{t \rightarrow \infty} \delta(A(t), F^*) = 0$ .

**命题 2** AMFS 以概率 1 收敛于 Pareto 最优解集.

**证明** 令  $s$  为解空间的任意个体,  $s$  可以表示为  $\{s_1, s_2, \dots, s_m\}$ , 其中  $s_k$  为 0 或 1,  $k = 1, 2, \dots, m$ . 设  $\{s'_1, s'_2, \dots, s'_m\}$  和  $\{s''_1, s''_2, \dots, s''_m\}$  为  $B(t)$  中的任意两个个体. 已知 AMFS 的变异概率  $\sigma = \sqrt{\frac{1}{t^2}} \mu \times \sqrt{\frac{1}{N}} \in (0, 1)$ , 且依据个体  $i$  的环境适应度  $\mu$ , 其被选择繁育下一代的概率为  $\epsilon_i (0 < \epsilon_i < 1)$ .

令:

$$\{s'''_1, s'''_2, \dots, s'''_m\} = \{s'_1|_{\sigma}, \dots, s'_{k-1}|_{\sigma}, s''_k|_{\sigma}, \dots, s''_m|_{\sigma}\}$$

为  $s'$  和  $s''$  的后代, 其中  $s''_k|_{\sigma}$  表示  $s''$  以概率  $\sigma$  发生变异,  $k$  为交叉点. 假设  $s_i = \begin{cases} s'_i, i < k \\ s''_i, i \geq k \end{cases}$  的概率为  $\tau_i$ ,  $0 \leq \tau_i \leq 1, i = 1, 2, \dots, m$ . 则两条基因  $s$  和  $s'''$  第  $i$  位相等的概率为:

$$P(s_i = s'''_i) = \tau_i(1 - \sigma) + (1 - \tau_i)\sigma = (1 - 2\sigma)\tau_i + \sigma$$

因为  $0 < \sigma < 1$  且  $\tau_i (0 \leq \tau_i \leq 1)$ , 所以  $P(s_i = s'''_i) \geq \sigma$ , 两条基因  $s$  和  $s'''$  相等的概率为:

$$P(s_i = s'''_i) = \prod_{i=1}^m (\tau_i(1 - \sigma) + (1 - \tau_i)\sigma) \geq \sigma^m$$

可知:

$$p(s_i \in B(t+1)) =$$

$$\sum_{j=1}^n \sum_{i=1}^m \left( \epsilon_j \epsilon_i \left( \prod_{i=1}^m (\tau_i(1 - \sigma) + (1 - \tau_i)\sigma) \right) \right) \geq n^2 \epsilon_{\min}^2 \sigma^m$$

其中,  $\epsilon_{\min}$  是  $\epsilon_i$  的最小值. 显然  $0 < n^2 \epsilon_{\min}^2 \sigma^m < 1$ , 取  $\epsilon = n^2 \epsilon_{\min}^2 \sigma^m$ , 则  $P(s \in B(t+1)) \geq \epsilon$ .

综上, AMFS 算法中, 概率  $P(s \in B(t))$  不依赖迭代次数  $t$  且存在  $\epsilon (0 < \epsilon < 1)$ , 满足  $P(s \in B(t)) \geq \epsilon$ . 根据定义 1 与命题 1, 本文算法 AMFS 以概率 1 收敛于 Pareto 最优解集.

## 4 实验与分析

为了验证提出的算法的有效性, 采用不同应用领域中的六个高维数据集进行对比实验, 其中, 最大特征数和最小特征数分别为 60 和 13, 最大样本数和最小样本数分别为 2310 和 178, 数据集的具体信息如表 1 所示. 实验平台为 Core i7-10510U, 2.30 GHz, 16 GB, 仿真算法均在 Matlab R2016a 中实现.

为了证明提出的算法表现优越, 采用五种现有的多目标特征选择算法与 AMFS 进行比较, 分别为 NSGAFS<sup>[37]</sup>, MOFSBDE<sup>[19]</sup>, FSMOPSO<sup>[20]</sup>, DCDREA<sup>[21]</sup> 和 NSGA-II<sup>[12]</sup>. 算法的相关参数如表 2 所示, 其中,  $N_f$  表示特征的数量, 所有比较算法均采用  $K$  近邻 (KNN) 分类器,  $K$  设置为 5. AMFS 的种群大小  $Pop = N_s$ , 迭代次数  $T = 100$ .

表 1 实验使用的数据集

Table 1 Datasets used in experiments

序号	数据集	特征数	样本数	类别
1	Vehicle	18	846	4
2	Wine	13	178	3
3	Sonar	60	208	2
4	Ionosphere	34	351	2
5	Vowel	13	990	11
6	Segmentation	19	2310	7

表2 算法的参数设置

Table 2 Algorithm parameter settings

算法	参数设置
NSGAFS	$Pop = 25, T = 100, Cr = 0.7, Mu = 0.1$
DCDREA	$Pop = N_f/20, T = 70, Mu = 0.1$
MOFSBDE	$Pop = 50, T = 100, Cr = 0.3, F = 0.5 \times rand$
FSMOPSO	$Pop = 30, T = 100$ $c_1 = 2.5 - 2 \times t/T, c_2 = 0.5 + 2 \times t/T$
NSGA-II	$Pop = 25, T = 100, Cr = 0.5, Mu = 0.1$
AMFS	$Pop = 50, T = 100$

#### 4.1 AMFS与不同EAs特征选择性能比较 为 了评估AMFS的性能,从最佳分类精度和平均分

类精度两方面进行度量,分别记为Best和Avg. 所有算法均独立运行10次,所有对比算法的Best的值是Pareto解集中距离原点欧几里得距离最近的解,AMFS的Best值是采用基于环境因子熵权计算所得. 另外,AD\_Best和AD\_Avg分别为对应算法的Best和Avg的平均值. 表3展示了AMFS与其他五个多目标进化算法的比较结果,表中黑体字表示所有算法中的最优值. 此外,采用 $t$ 检验来验证AMFS的显著性差异,表3中 $T$ -test\_Best和 $T$ -test\_Avg分别表示对应的 $P$ , $P < 0.05$ 表示差异显著,记作“+”, $P > 0.05$ 表示差异不显著,记作“-”.

表3 AMFS与五种多目标进化算法的性能比较

Table 3 Performance of AMFS and other five multi-objective evolutionary algorithms

数据集	性能指标	MOFSBDE	FSMOPSO	NSGAFS	DCDREA	NSGA-II	AMFS
Vehicle	Best	97.81%	95.31%	98.21%	97.82%	81.81%	<b>98.99%</b>
	Avg	93.48%	91.22%	97.11%	95.99%	82.82%	<b>97.66%</b>
Wine	Best	89.32%	90.89%	<b>97.92%</b>	96.21%	84.42%	<b>97.78%</b>
	Avg	86.56%	87.01%	94.24%	93.37%	83.32%	<b>96.63%</b>
Sonar	Best	83.29%	84.10%	91.59%	89.23%	77.32%	<b>93.31%</b>
	Avg	79.99%	80.11%	87.31%	88.42%	77.39%	<b>91.99%</b>
Ionosphere	Best	94.59%	<b>100%</b>	89.95%	<b>100%</b>	65.12%	99.82%
	Avg	91.70%	97.89%	86.67%	98.78%	64.31%	<b>98.99%</b>
Vowel	Best	97.57%	85.68%	75.79%	97.77%	66.83%	<b>99.13%</b>
	Avg	96.62%	84.70%	73.99%	96.10%	57.83%	<b>97.75%</b>
Segmentation	Best	97.73%	97.22%	98.91%	<b>100%</b>	89.80%	<b>100%</b>
	Avg	96.86%	96.61%	97.45%	97.65%	87.79%	<b>98.89%</b>
	AD_Best	93.82%	92.68%	93.97%	97.91%	77.55%	<b>98.22%</b>
	AD_Avg	91.35%	90.55%	91.41%	95.58%	75.57%	<b>97.11%</b>
$T$ -test_Best		0.02639 (+)	0.03083 (+)	0.16461 (-)	0.08581 (-)	0.00444 (+)	
$T$ -test_Avg		0.01919 (+)	0.01489 (+)	0.09646 (-)	0.01350 (+)	0.00851 (+)	

由表3可见,AMFS在各个数据集上均优于NSGA-II,这是由于环境因子加权的自适应算子对全局搜索最优解更具优势. 在Ionosphere数据集上,FSMOPSO的Best为100%,AMFS的Best为99.82%,比FSMOPSO低0.18%,但AMFS的Avg比FSMOPSO高1.10%. AMFS仅在一个数据集上表现稍弱,在其他数据集上的精度都高于FSMOPSO. 在Wine数据集上,AMFS的Best比NSGAFS低,但和其他算法相比是最高的,而且其Avg是最优的. AMFS和DCDREA在各个

数据集上的性能都比较接近,仅在个别数据集中可能存在因AMFS筛选的特征子集包含冗余特征而劣于DCDREA的情况,但整体来看,AMFS的AD\_Best与AD\_Avg均高于DCDREA.

综上,AMFS具有优越的性能,其AD\_Best和AD\_Avg分别为98.22%和97.11%,这得益于基于环境因子的自适应交叉算子和自适应变异算子可以挖掘更多的优秀的候选个体. 此外, $t$ 检验的测试结果表明,AMFS与NSGA-II,MOFSBDE和FSMOPSO三种多目标进化算法相比,两



种指标均有显著差异,与NSGAFS之间没有显著差距,与DCDREA仅有一个指标有显著差异.证明AMFS与现有的大多数多目标特征选择进化算法相比,有明显的竞争性.

**4.2 AMFS的HV和SP指标的表现** 为了进一步检验多目标特征选择算法中的Pareto解集,使用超体积(HV)和间距(SP)来分析不同多目标特征算法的性能,其中,HV为算法得到的解集中的个体与目标空间中的参考点围成的超立方体的体积.由于AMFS,MOFSBDE,FSMOPSO,NSGAFS,NSGA-II和DCDREA采用两个优化目标,因此参考点设置为(1,1).HV越大,算法的整体性能越好.假设Pareto P1的HV大于Pareto P2,表示P1的多样性和收敛性优于P2.对于SP指标,它测量从每个解决方案到其他解决方案的最小距离的标准偏差,SP越小,解集越均匀.表4和表5展示了AMFS和其他五种算法的HV和SP的比较结果,表中黑体字表示结果最优.

如表4所示,AMFS算法的HV远远大于FSMOPSO,NSGAFS,NSGA-II.在Vowel数据集

上,FSMOPSO,NSGAFS,NSGA-II的HV分别为0.6472,0.6109,0.6019,AMFS的HV是0.9331,分别高0.2859,0.3222,0.3312.在Wine数据集上,MOFSBDE的HV超过了AMFS,但AMFS的HV整体上大于MOFSBDE.DCDREA与AMFS的结果相近,甚至在Segmentation数据集上,DCDREA的HV超过了AMFS,但是AMFS的HV在其他五个数据集上都大于DCDREA.*t*检验的结果表明,AMFS的HV明显优于大多数算法,证明AMFS得到的Pareto解集与其他算法相比,具有更好的分集和收敛性能.

如表5所示,AMFS的SP优于FSMOPSO,MOFSBDE,NSGA-II.在Wine数据集上,NSGAFS的SP为0.0021,比AMFS略低了0.0052.类似地,在Ionosphere与Segmentation数据集上,DCDREA的SP比AMFS算法略低0.0008和0.0006,但在其余四个数据集上,AMFS的表现更优.*t*检验的结果表明,AMFS的SP明显优于大多数算法,证明由AMFS获得的Pareto最优解集具有良好的分布均匀性.

表4 AMFS算法与五种多目标特征选择进化算法的HV的比较

Table 4 HV of AMFS and other five multi-objective feature selection evolutionary algorithms

数据集	MOFSBDE	FSMOPSO	NSGAFS	DCDREA	NSGA-II	AMFS
Vehicle	0.6533	0.5887	0.4733	0.8384	0.4433	<b>0.8536</b>
Wine	<b>0.8859</b>	0.7782	0.7059	0.8223	0.6679	0.8552
Sonar	0.6723	0.5995	0.5823	0.7935	0.5413	<b>0.8979</b>
Ionosphere	0.8585	0.4241	0.4786	0.9101	0.4585	<b>0.9121</b>
Vowel	0.8209	0.6472	0.6109	0.9211	0.6019	<b>0.9331</b>
Segmentation	0.8613	0.8484	0.7513	<b>0.9685</b>	0.7522	0.9563
<i>t</i> -test	0.03660 (-)	0.00861 (+)	0.00097 (+)	0.18799 (-)	0.00074 (+)	

表5 AMFS算法与五种多目标特征选择进化算法的SP的比较

Table 5 SP of AMFS and other five multi-objective feature selection evolutionary algorithms

数据集	MOFSBDE	FSMOPSO	NSGAFS	DCDREA	NSGA-II	AMFS
Vehicle	0.2369	0.0182	0.0172	0.0078	0.1272	<b>0.0028</b>
Wine	0.3326	0.0103	<b>0.0021</b>	0.0132	0.1113	0.0073
Sonar	0.2137	0.0328	0.0129	0.0216	0.2282	<b>0.0067</b>
Ionosphere	0.1879	0.0356	<b>0.0056</b>	0.0081	0.2049	0.0089
Vowel	0.3691	0.0798	0.0139	0.0154	0.2051	<b>0.0075</b>
Segmentation	0.3189	0.0289	0.0098	<b>0.0079</b>	0.1872	0.0085
<i>t</i> -test	0.00028 (+)	0.03692 (+)	0.31411 (-)	0.07404 (-)	0.00027 (+)	

由以上分析可知,本文提出的 AMFS 算法在高维数据集上能获得分类精度更高的 Pareto 解集并从中选取合理的唯一解,其主要原因:一是将进化过程中的环境因子嵌入改进的自适应交叉和变异算子,有效地动态监测进化个体中不同特征组合的性能表现,避免了主观因素对搜索特征子集的影响,以自适应的方式获得综合性能较好的 Pareto 前沿解;二是引入了环境因子熵权,能在不同优化目标下计算信息熵权,自动判别 Pareto 前沿中的最佳折中解,减少包含重要特征的进化个

体因被支配而错误剔除的可能性,保证全局最优解的搜索效率和精度。

表 6 给出了 AMFS 与五种对比算法的时间复杂度,由表可见,AMFS 略占优势。这是由于时间复杂度主要依赖于环境因子  $\mu(g)$  的计算,其环境因子的自适应算子在寻找全局最优解时可加速收敛过程,总体时间复杂度略优于其他五种多目标特征选择算法,但进一步减少计算环境因子的时间开销仍需继续研究。

表 6 AMFS 与五种多目标特征选择进化算法的时间复杂度的比较

Table 6 Time complexity of AMFS and other five multi-objective feature selection evolutionary algorithms

MOFSBDE	FSMOPSO	NSGAFS	DCDREA	NSGA-II	AMFS
$O(N^2 \times M)$	$O(N^3 \times M)$	$O(n \times (N^2 \times M))$	$O((n \times m)(N^2 \times M))$	$O(N^2 \times M)$	$O((n \times m)(N \lg N \times M))$

#### 4.3 AMFS 与不同最适解选择策略的性能比较

为了检验 AMFS 选择最适解的性能,进行了消融实验,采用 TOPSIS、熵权 TOPSIS (ETOPSIS) 和基于环境因子熵权 TOPSIS (BETOPSIS) 来赋值权重,根据不同权重对同一实验结果进行选择,判断三种方法选择最适解的能力。使用的数据集为 Vehicle,  $Pop$  设置为 50, 迭代次数  $T=100$ , 权重设置如表 7 所示。

表 7 AMFS 算法消融实验权重设置

Table 7 Weight settings for ablation experiments of AMFS algorithm

TOPSIS	$W_j = 0.5$
ETOPSIS	$W_j = \frac{d_j}{\sum_{j=1}^m d_j} (j=1, 2)$
BETOPSIS	$W_j = \mu \frac{d_j}{\sum_{j=1}^m d_j} (j=1, 2)$

图 2a 为通过 AMFS 获得的 Pareto 解集, 图 2b 为用三种不同的获取权重指标方法所选择的最适解, 其中青色解为 Pareto 最优解, 紫色解为 TOPSIS 选择解, 蓝色解为熵权 TOPSIS 选择解, 红色解为基于环境因子熵权 TOPSIS 选择解。实验结果如表 8 所示。由表可见, 与 TOPSIS 选出的结果相比, BETOPSIS 的特征相关系数仅升高 0.2, 但特征选择率大幅度缩减了 20%; 和 ETOPSIS 选

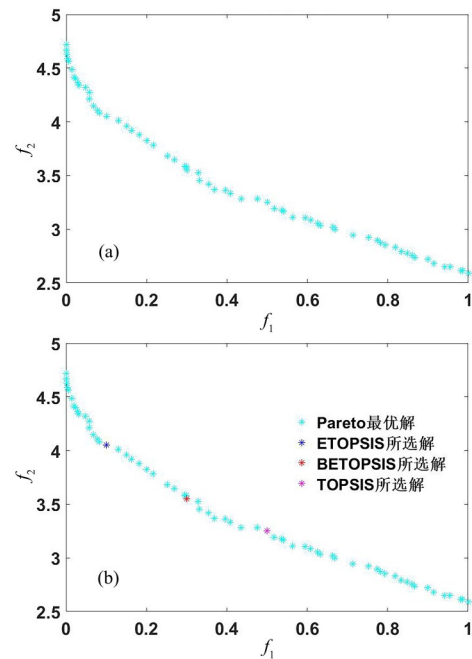


图 2 三种方法选择 Pareto 解集最适解

Fig. 2 Three methods for selecting the most suitable Pareto solution set

表 8 AMFS 算法的消融实验结果

Table 8 Ablation experimental results of AMFS

	$f_1$ : 特征选择率	$f_2$ : 特征相关系数
TOPSIS	0.5	3.3
ETOPSIS	0.15	4
BETOPSIS	0.3	3.5

出的结果相比,BETOPSIS的特征相关系数降低0.5,特征选择率升高15%。这是因为TOPSIS权重是人为设置的,偏离值相对较大,且主观因素占比较高;ETOPSIS在TOPSIS的基础上去除了主观因素,获得的最适解相对较好;BETOPSIS在ETOPSIS的基础上增加了对数据集特征依赖度的判定,选择出的结果既消除了主观因素,又增加了选择权重的可解释性,获得的最适解最优。证明BETOPSIS对选择最适解有一定幅度的提升。

## 5 结论

本文提出一种基于自适应环境因子熵权决策的多目标特征选择算法,首先通过加权环境因子来自适应地筛去不相关特征,以提供高质量的候选特征子集空间;然后,将环境因子嵌入改进的交叉算子和变异算子,加速最优特征子集的自适应搜索;最后,利用基于环境因子的熵权分析决策出Pareto最优解集中的最适解。实验结果表明,针对高维特征选择问题,本文提出的AMFS的分类精度表现较好。未来将关注多目标优化特征选择搜索策略的高效性,研究低开销的Pareto求解算子,此外,特征子集的稳定性分析是另一个值得考虑的研究方向。

### 参考文献

- [1] 梁正平,林万鹏,胡凯峰,等. 基于帕累托前沿面曲率预估的超多目标进化算法. 软件学报, 2023, 34(9):4096—4113. (Liang Z P, Lin W P, Hu K F, et al. Many-objective evolutionary algorithm based on curvature estimation of Pareto front. Journal of Software, 2023, 34(9):4096—4113.)
- [2] Bao C T, Gao D J, Gu W, et al. A new adaptive decomposition-based evolutionary algorithm for multi- and many-objective optimization. Expert Systems with Applications, 2023(213):119080.
- [3] 谢承旺,余伟伟,郭华,等. DAV-MOEA:一种采用动态角度向量支配关系的高维多目标进化算法. 计算机学报, 2022, 45(2):317—333. (Xie C W, Yu W W, Guo H, et al. DAV-MOEA: A many-objective evolutionary algorithm adopting dynamic angle vector based dominance relation. Chinese Journal of Computers, 2022, 45(2):317—333.)
- [4] 廉杰,姚鑫,李占山. 用于特征选择的乌鸦搜索算法的研究与改进. 软件学报, 2022, 33(11):3903—3916. (Lian J, Yao X, Li Z S. Research and improvements on crow search algorithm for feature selection. Journal of Software, 2022, 33(11):3903—3916.)
- [5] Xue Y, Zhu H K, Neri F. A feature selection approach based on NSGA-II with ReliefF. Applied Soft Computing, 2023(134):109987.
- [6] Wang P, Xue B, Liang J, et al. Feature selection using diversity-based multi-objective binary differential evolution. Information Sciences, 2023(626):586—606.
- [7] Robindro K, Clinton U B, Hoque N, et al. JoMIC: A joint MI-based filter feature selection method. Journal of Computational Mathematics and Data Science, 2023(6):100075.
- [8] Al-Yaseen W L, Idrees A K, Almasoudy F H. Wrapper feature selection method based differential evolution and extreme learning machine for intrusion detection system. Pattern Recognition, 2022(132):108912.
- [9] Kermani F Z, Eslami E, Sadeghi F. Global Filter-Wrapper method based on class-dependent correlation for text classification. Engineering Applications of Artificial Intelligence, 2019(85):619—633.
- [10] Tarkhaneh O, Nguyen T T, Mazaheri S. A novel wrapper-based feature subset selection method using modified binary differential evolution algorithm. Information Sciences, 2021(565):278—305.
- [11] Yaman O, Yol F, Altinors A. A fault detection method based on embedded feature extraction and SVM classification for UAV motors. Micro-processors and Microsystems, 2022(94):104683.
- [12] 樊田田,许蕾,陈林. 基于多目标优化算法NSGA-II推荐相似缺陷报告. 计算机学报, 2019, 42(10):2175—2189. (Fan T T, Xu L, Chen L. Recommending similar bug reports based on multi-targets optimization algorithm NSGA-II. Chinese Journal of Computers, 2019, 42(10):2175—2189.)
- [13] 齐琦,毋涛. 基于改进NSGA-II算法的多目标生产智能调度. 计算机技术与发展, 2021, 31(8):162—168. (Qi Q, Wu T. Multi-objective intelligent production optimal scheduling based on improved

- NSGA - II algorithm. Computer Technology and Development, 2021, 31(8): 162—168.)
- [14] Ghorbani B, Shirmohammadi R, Mehrpooya M, et al. Structural, operational and economic optimization of cryogenic natural gas plant using NSGAII two - objective genetic algorithm. Energy, 2018(159): 410—428.
- [15] Xu Z, Ning X, Yu Z L, et al. Design optimization of a shell - and - tube heat exchanger with disc - and - doughnut baffles for aero - engine using one hybrid method of NSGA - II and MOPSO. Case Studies in Thermal Engineering, 2023(41): 102644.
- [16] Liu T, Yuan Q Y, Ding X M, et al. Multi-objective optimization for greenhouse light environment using Gaussian mixture model and an improved NSGA - II algorithm. Computers and Electronics in Agriculture, 2023(205): 107612.
- [17] Zhang P, Qian Y Y, Qian Q. Multi - objective optimization for materials design with improved NSGA - II. Materials Today Communications, 2021 (28): 102709.
- [18] Chamlal H, Ouaderhman T, Rebbah F E. A hybrid feature selection approach for Microarray datasets using graph theoretic - based method. Information Sciences, 2022(615): 449—474.
- [19] Zhang Y, Gong D W, Gao X Z, et al. Binary differential evolution with self - learning for multi - objective feature selection. Information Sciences, 2020(507): 67—85.
- [20] Zhang Y, Gong D W, Cheng J. Multi - objective particle swarm optimization approach for cost-based feature selection in classification. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2017, 14(1): 64—75.
- [21] 孙哲人, 黄玉划, 陈志远. 基于多样性分类和距离回归的进化算法. 软件学报, 2022, 33(10): 3700—3716. (Sun Z R, Huang Y H, Chen Z Y. Diversity classification and distance regression assisted evolutionary algorithm. Journal of Software, 2022, 33(10): 3700—3716.)
- [22] Zhou Y, Kang J H, Guo H N. Many - objective optimization of feature selection based on two-level particle cooperation. Information Sciences, 2020 (532): 91—109.
- [23] Rashno A, Shafipour M, Fadaei S. Particle ranking: An Efficient method for multi - objective particle swarm optimization feature selection. Knowledge - Based Systems, 2022(245): 108640.
- [24] 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. Applied Soft Computing, 2020(8): 106041
- [25] Zhao W G, Zhang Z X, Mirjalili S, et al. An effective multi-objective artificial hummingbird algorithm with dynamic elimination - based crowding distance for solving engineering design problems. Computer Methods in Applied Mechanics and Engineering, 2022(398): 115223.
- [26] Chen L L, Li Q, Zhao X H, et al. Multi-population coevolutionary dynamic multi - objective particle swarm optimization algorithm for power control based on improved crowding distance archive management in CRNs. Computer Communications, 2019(145): 146—160.
- [27] 马畅畅, 汪坤, 鹿晓梦, 等. 一种确定目标域多目标优化算法 NSGA/P. 计算机技术与发展, 2022, 32(5): 15—21, 28. (Ma C C, Wang K, Lu X M, et al. A multi-objective optimization algorithm for determinant objective domain NSGA/P. Computer Technology and Development, 2022, 32(5): 15—21, 28.)
- [28] Yue C T, Suganthan P N, Liang J, et al. Differential evolution using improved crowding distance for multimodal multiobjective optimization. Swarm and Evolutionary Computation, 2021(62): 100849.
- [29] 郑金华, 董南江, 阮干, 等. 决策空间定向搜索的高维多目标优化策略. 软件学报, 2019, 30(9): 2686—2704. (Zheng J H, Dong N J, Ruan G, et al. High-dimensional multi - objective optimization strategy based on decision space oriented search. Journal of Software, 2019, 30(9): 2686—2704.)
- [30] Liu J H, Lin Y J, Ding W P, et al. Multi-label feature selection based on label distribution and neighborhood rough set. Neurocomputing, 2023 (524): 142—157.
- [31] Kashef S, Nezamabadi-pour H. A label-specific multi-label feature selection algorithm based on the Pareto dominance concept. Pattern Recognition, 2019(88): 654—667.



- [32] Voorneveld M. Characterization of Pareto dominance. *Operations Research Letters*, 2003, 31(1):7—11.
- [33] Aziz H, Brandl F, Brandt F. Universal Pareto dominance and welfare for plausible utility functions. *Journal of Mathematical Economics*, 2015(60):123—133.
- [34] Rao R V, Lakshmi R J. Ranking of Pareto-optimal solutions and selecting the best solution in multi- and many - objective optimization problems using R - method. *Soft Computing Letters*, 2021(3):100015.
- [35] Bryan J A, Zhang Y L, Wang H L, et al. Multi-objective design optimization of an integrated regenerative transcritical cycle considering sensitivity in Pareto-optimal solutions. *Energy Conversion and Management*;X, 2023(18):100364.
- [36] Ponsi F, Bassoli E, Vincenzi L. A multi-objective optimization approach for FE model updating based on a selection criterion of the preferred Pareto - optimal solution. *Structures*, 2021(33):916—934.
- [37] Hamdani T M, Won J M, Alimi A M, et al. Multi-objective feature selection with NSGA II //The 8<sup>th</sup> International Conference on Adaptive and Natural Computing Algorithms. Springer Berlin Heidelberg, 2007:240—247.

(责任编辑 杨可盛)