A multi-objective evolutionary algorithm with mutual-information-guided improvement phase for feature selection in complex manufacturing processes

An-Da Li[a,b], Zhen He[c], Qing Wang[a,b], Yang Zhang[a,b,*], Yanhui Ma[d,*]

[a] *School of Management, Tianjin University of Commerce, Tianjin 300134, China*
[b] *Research Center for Management Innovation and Evaluation, Tianjin University of Commerce, Tianjin 300134, China*
[c] *College of Management and Economics, Tianjin University, Tianjin 300072, China*
[d] *School of Management, Tianjin University of Technology, Tianjin 300384, China*

Abstract

Complex manufacturing processes (CMP) involve numerous features that impact product quality. Therefore, selecting key process features (KPF) is crucial for effective quality prediction and control in CMPs. This paper proposes a KPF (feature) selection method for the high-dimensional CMP data. The KPF selection problem is formulated as a bi-objective combinatorial optimization task of maximizing the geometric mean measure and minimizing the number of selected features. To solve this challenging high-dimensional KPF selection problem, we propose a novel multi-objective evolutionary algorithm (MOEA) called NSGAII-MIIP. NSGAII-MIIP applies an improvement phase (called MIIP) to purify the non-dominated solutions obtained by genetic operators during the iteration process to improve the FS performance. The improvement phase is guided by a mutual-information-based feature importance measure considering both a feature's relevance degree to class (product quality level) and its redundancy degree to selected features. This allows MIIP to efficiently update non-dominated solutions by selecting relevant features and eliminating redundant features. Moreover, MIIP is seamlessly integrated into the solution ranking process of NSGAII-MIIP so that solutions from the improvement phase can be ranked together with original solutions in the population efficiently. Experiments on eight datasets show that NSGAII-MIIP has better KPF selection performance than eight state-of-the-art multi-objective FS methods. Moreover, NSGAII-MIIP exhibits superior search performance compared to eight typical multi-objective optimization algorithms.

Keywords: Evolutionary computations, Multi-objective optimization, Feature selection, Improvement phase, Quality prediction

*Please cite as: Li, A.-D., He, Z., Wang, Q., Zhang, Y., & Ma, Y. (2025). A multi-objective evolutionary algorithm with mutual-information-guided improvement phase for feature selection in complex manufacturing processes. European Journal of Operational Research, 323(3), 952–965. https://doi.org/10.1016/j.ejor.2024.12.036

*Corresponding authors

Email addresses: adli@tjcu.edu.cn (An-Da Li), zhhe@tju.edu.cn (Zhen He), wangqing@tjcu.edu.cn (Qing Wang), yzhang@tcju.edu.cn (Yang Zhang), mayanhui@email.tjut.edu.cn (Yanhui Ma)

1. Introduction

With the wide implementation of sensors, the process parameters or control characteristics (called process features) affecting product quality can be recorded in the data collected from manufacturing processes, which is essential for data-driven quality techniques such as quality prediction (Zhang et al., 2022a; Deng et al., 2023), statistical process control (SPC) (Li et al., 2020b), advanced quality control (AQC) (Chien et al., 2022), and in-process quality improvement (Shi, 2023). Complex manufacturing processes (CMP) in modern industries contain numerous process features (PF), which significantly increases the dimensionality of CMP data. It is required to select the key PFs (KPF) strongly related to product quality for data dimensionality reduction (Anzanello et al., 2012; Li et al., 2019), so that the performance of data-driven quality techniques can be improved. In this paper, we focus on building a data-driven method to select KPFs in CMPs with good predictive performance for product quality.

KPF selection is naturally a feature selection (FS) problem (Guyon & Elisseeff, 2003) in machine learning that aims to identify the most informative features (i.e., PFs) related to the response variable (product quality). Generally, FS methods can be classified into filter and wrapper categories. Filter methods adopt a measure based on distance, information theory, or statistical theory to evaluate feature importance, which guides the selection of informative features (Yu & Liu, 2004). In recent years, measures based on the partial least squares regression (PLSR) model (Anzanello et al., 2009), mutual information (MI) (Guo & Banerjee, 2017), etc., have been proposed to build filter-based KPF selection methods. Wrapper methods directly use the classification performance of a learning model to evaluate a feature subset's goodness. Classifiers such as K nearest neighbors (KNN) (Pandiyan et al., 2018) and naive Bayesian (NB) (Li et al., 2019) have been applied to build wrapper-based KPF selection methods in the literature.

FS has shown to be an NP-hard combinatorial problem with a decision space of 2^N (N is the number of original features) (Amaldi & Kann, 1998), which is enormous given a great value of N. Different heuristic search algorithms, including sequential forward selection (SFS) (Kohavi & John, 1997) and sequential backward selection (SBS) (Kohavi & John, 1997), have been applied to FS for finding a desirable feature subset in an acceptable running time. Moreover, evolutionary computation (EC) techniques (Cosson et al., 2024) are well-suited for NP-hard FS problems because they offer good global search performance, are derivation-free, and are easy to implement. In recent years, the EC algorithms such as genetic algorithms (GAs) (Oztekin et al., 2018; Liu et al., 2022; Jiao et al., 2024), particle swarm optimization (PSO) (Mistry et al., 2017; Li et al., 2021; Nguyen et al., 2024), differential evolution (DE) (Zhang et al., 2020; Wang et al., 2023a; Hancer et al., 2023), and ant colony optimization (Wang et al., 2023b), have been successfully applied to FS.

An FS task generally involves two objectives, i.e., maximizing the classification performance and minimizing the size of feature subsets (solutions) (Xue et al., 2016). To optimize this bi-objective FS problem, single-objective EC algorithms need a pre-processing step to aggregate the two objectives, which requires

much domain knowledge. Multi-objective EC algorithms can handle conflicted multiple objectives simultaneously and obtain a set of non-dominated solutions approximating the Pareto-optimal set (Sahinkoc & Ümit Bilge, 2022). Thus, multi-objective EC algorithms, such as multi-objective GA (Li et al., 2016; Xue et al., 2022), multi-objective DE (Zhang et al., 2020), and multi-objective PSO algorithms (Xue et al., 2013), have been increasingly proposed to build FS methods. These algorithms aim to maximize the classification accuracy and minimize the feature subset size for FS. However, on unbalanced data, accuracy may lead to undesirable FS results as it is not a good classification performance measure. Therefore, performance measures designed for unbalanced data, including balanced accuracy (Tran et al., 2018), expected maximum profit (EMP) (Kozodoi et al., 2019; Simumba et al., 2022), and geometric mean (GM) (Li et al., 2020a, 2023) have been used instead of accuracy to address the data imbalance problem in many FS methods.

Although the EC-based FS methods have been widely studied, FS is still a challenging task for EC algorithms, especially on high-dimensional data with numerous features, because of the huge decision space and complicated feature interactions (Ahadzadeh et al., 2023). Local improvement strategies have been embedded in some EC algorithms to improve the FS performance. Zhang et al. (2020) applied a one-bit purifying step in a multi-objective DE algorithm. This purifying step locally updates the non-dominated solutions based on the feature importance result from feature comparison. He et al. (2022) proposed a decomposition-based multi-objective PSO algorithm for KPF selection. An improvement phase is used to update the non-dominated solutions from the PSO mechanism by randomly adding and eliminating a feature during the iteration process. Hancer (2022) proposed a cost-sensitive multi-objective artificial bee colony algorithm for FS, where a local improvement strategy that randomly eliminates features with high costs is used. The random search-based improvement strategies are used in the above methods. However, these random strategies do not consider feature relevance (correlations between features and the class) or redundancy (correlations among features) information, which can be very beneficial for improving FS performance. To address this issue, some studies have constructed feature relevance or redundancy measures to build an improvement phase in EC-based FS methods. Moradi & Gholampour (2016) divided features into similar and dissimilar groups considering feature redundancy first. Then, an improvement phase is used to update a solution by selecting/removing features in the dissimilar/similar group. Hancer (2024) adopted a local improvement mechanism in a DE algorithm that adds or eliminates features regarding the best solution in a probability manner. The features to be added or eliminated are selected by MI-based feature relevance and redundancy measures. In Song et al. (2021) and Zhang et al. (2022b), improvement strategies using MI-based feature relevance and redundancy measures are used to update the best and worst particles in PSO. However, the feature relevance/redundancy-based improvement strategies are designed for single objective EC algorithms. Building an effective improvement strategy in the multi-objective FS scenario, which involves multiple best solutions (i.e., non-dominated solutions) and worst solutions in the swarm/population, is needed.

1.1. Motivation

As mentioned above, filter and wrapper-based KPF selection methods have been studied in the literature. Compared to filters, wrappers can select KPFs with better predictive performance for product/process quality since a learning model is involved. However, wrappers require substantially more computational time than filters because they need to search in a huge decision space to find the best PF subset, and evaluating the effectiveness of a candidate PF (feature) subset is time-consuming. Therefore, a more time-efficient wrapper-based KPF selection method is required.

Multi-objective EC algorithms have been widely applied to build wrapper-based FS methods for their good search abilities. However, the slow convergence speed on high-dimensional CMP data (with numerous PFs) may limit their performance in finding the KPF set in an acceptable time. Although some studies have embedded an improvement phase in multi-objective EC algorithms to improve the convergence speed for FS, these improvement strategies do not comprehensively consider the intrinsic property of data, i.e., feature relevance and redundancy, which limits their FS performance. This drives us to propose a new improvement strategy considering feature relevance and redundancy and embed this strategy in a multi-objective EC algorithm for KPF selection.

1.2. Goals and contributions

The goal of this paper is to design a new multi-objective evolutionary algorithm (MOEA) called NSGAII-MIIP (NSGA-II with MI-guided improvement phase) to build a wrapper-based FS method for selecting KPFs in CMPs. In NSGAII-MIIP, an improvement phase to purify non-dominated solutions (feature/PF subset) is used, enabling the algorithm to search for KPFs efficiently. The improvement strategy considers both a feature's relevance and redundancy when updating a solution. We select NSGA-II (non-dominated sorting GA II) as the base algorithm to build NSGAII-MIIP. This is because the sequential solution ranking manner of NSGA-II makes it easy and efficient to embed an improvement phase in the solution ranking process (see Section 4.4 for details). The contributions of the proposed method are summarized as follows:

1. An MI-based feature importance measure is proposed to evaluate features. It evaluates the feature (PF) importance considering both the relevance degree between a feature and the class label (product quality) and the redundancy degree between a feature and a given feature subset.
2. An improvement strategy is established to purify a solution (feature/PF subset). This strategy contains three improvement operations, which add, eliminate, or interchange features in terms of a solution. The proposed MI-based feature importance measure is used to select the features to perform the improvement operations. It enables the improvement strategy to effectively explore promising local areas most likely to update a solution to a new one with a higher relevance level to class and a lower interior feature redundancy.

3. An improvement-phase-embedded ranking approach is proposed for NSGAII-MIIP. It adopts the proposed improvement strategy to purify current non-dominated solutions during the solution ranking process. Specifically, the ranking process pauses when the non-dominated solutions are found and the improvement phase is conducted to search for new solutions based on the non-dominated solutions. Then, the ranking process continues the original ranking operations to sort these new solutions along with the original solutions in the population. This embedding strategy is efficient because the time complexity of the ranking process does not change compared with the ranking process with no improvement phase.

We have verified the KPF selection performance of NSGAII-MIIP on four real CMP datasets and the synthetic datasets with additional features generated with these real datasets with a set of experiments. First, experimental results indicate NSGAII-MIIP can quickly obtain a few KPFs with good predictive performance for product quality with a limited computational budget. Second, from a multi-objective optimization (MOO) perspective, the overall quality prediction performance of non-dominated solutions found by NSGAII-MIIP is better than benchmark multi-objective FS methods. Finally, NSGAII-MIIP shows better search performance than eight benchmark MOO algorithms, which demonstrates the effectiveness of the improvement phase.

The rest of this paper is organized as follows. Section 2 introduces the KPF selection problem addressed in this paper. Section 3 first defines the MI-based feature importance measure and then proposes the improvement strategy. Section 4 proposes the NSGAII-MIIP algorithm for the KPF selection problem. Section 5 introduces the experimental settings for verifying the proposed method. Section 6 presents the KPF selection results of NSGAII-MIIP and benchmark FS methods. Further analysis on the KPF selection performance and search performance of NSGAII-MIIP is shown in Section 7. Section 8 presents the conclusions of this paper and future research interests. Finally, the supplementary material evaluates the time complexity of NSGAII-MIIP in detail and presents the additional results of the performance analysis of NSGAII-MIIP.

2. KPF selection problem

Let $\mathcal{D} = \{\mathbf{F}, \mathbf{y}\}$ be a dataset from a CMP with M products (instances), where $\mathbf{F}$ is a $M \times N$ matrix recording the measurements N PFs ($\mathbb{F} = \{f_1, f_2, ..., f_N\}$) and $\mathbf{y}$ is a M-dimensional vector recording the measurements of the product quality level (class label) $Y \in \{+1, -1\}$. In this paper, we assume that $Y = +1$ and $Y = -1$ denote the quality levels of the minority and majority class instances. The objective of KPF selection is to select a subset $\mathbb{X} \subseteq \mathbb{F}$ with which we can build a concise learning model to predict the product quality level Y by eliminating redundant and irrelevant PFs. As suggested in Li et al. (2019), the KPF

selection problem is constructed as a bi-objective FS problem based on the wrapper framework as

$$\begin{aligned} &\text{minimize} && \mathbf{F}_{obj} = (1 - GM(\mathbb{X}), |\mathbb{X}|)^T \\ &\text{subject to} && \mathbb{X} \subseteq \mathbb{F} \end{aligned} \tag{1}$$

where $\mathbb{X}$ denotes a feasible solution (PF subset), $GM(\mathbb{X})$ denotes the GM value (geometric mean of sensitivity and specificity) obtained by $\mathbb{X}$ for predicting Y, and $|\mathbb{X}|$ denotes the number of PFs in $\mathbb{X}$. Eq. (1) defines a KPF selection problem with two objectives, i.e., maximizing the classification performance (i.e., GM) of and minimizing the number of selected PFs. Therefore, a small-size PF subset with good prediction performance for product quality can be obtained. To evaluate the value of $GM(\mathbb{X})$, as suggested in Kohavi & John (1997), an inner-loop 5-fold cross-validation (CV) is applied on the training set, dividing the training set into 5 pairs of the inner-loop training and test sets. Then, the average GM value over the 5 inner-loop test sets is obtained as the evaluated $GM(\mathbb{X})$.

3. Mutual-information-guided improvement strategy

In this section, an MI-based feature importance measure is first defined. Then, a clustering algorithm to divide the features into different groups is proposed. Finally, three improvement operations based on the importance measure and the clustering results are proposed.

3.1. Mutual-information-based feature importance measure

FS aims to maintain a few informative features by eliminating irrelevant and redundant features. Feature relevance and redundancy are two major concerns in FS. Feature relevance measures the correlation degree between a feature and the class, while feature redundancy measures the interaction degrees among the predictive features. A feature f is redundant to a feature subset $\mathbb{X}$ if f is related to $\mathbb{X}$ but the class-discrimination power of $\mathbb{X} \cup \{f\}$ is not significantly different from $\mathbb{X}$. Feature relevance and redundancy can be effectively measured based on the information theory.

Entropy is a basic concept in Shannon's information theory that measures the uncertainty of a random variable. The entropy of a variable X is defined as

$$H(X) = -\int p(x) \log p(x) \mathrm{d}x. \tag{2}$$

MI is also an important measure in information theory that measures the degree of correlation between two variables/features. It captures both the linear and non-linear relations between two variables. Let X and Y be two variables, $p(x)$ or $p(y)$ be the prior probability of $X = x$ or $Y = y$, and $p(y, x)$ be the joint probability

of $X = x$ and $Y = y$. The MI of X and Y is

$$I(Y;X) = \iint p(y,x) \log \frac{p(y,x)}{p(y)p(x)} \mathrm{d}y\mathrm{d}x. \tag{3}$$

Based on the definition of entropy and MI in Eqs. (2) and (3), the feature relevance and redundancy measures can be further defined as follows.

Definition 1. (*Feature relevance*) Let f be a feature and Y be the class label. The feature relevance of f is defined as the correlation degree of f to Y. It is calculated as the MI of f and Y, i.e.,

$$Relevance(f) = I(Y;f). \tag{4}$$

Feature redundancy measures the degree of correlation among selected features. Different feature redundancy measures based on MI have been proposed in the literature. Battiti (1994) defines a redundancy measure that measures the correlation degree between a feature f and a given feature subset $\mathbb{X}$ as the average of MI values between f and the features in $\mathbb{X}$. This measure has been widely used in the literature (Peng et al., 2005; Song et al., 2021; Hancer, 2024) to build FS methods. However, Battiti's redundancy measure does not consider the interaction between features and the class, which yields biased redundancy evaluation results. To this end, Kwak & Choi (2002) proposed a class-dependent measure to estimate the redundancy of a feature concerning a feature subset. This measure evaluates the redundancy degree of two features considering their interactions in predicting the class label. In this paper, we take this class-dependent measure as the feature redundancy measure, which is introduced below.

Definition 2. (*Feature redundancy*) Let $\mathbb{X}$ be a feature subset, f be a feature ($f \notin \mathbb{X}$), and Y be the class label. The class-dependent feature redundancy of f with respect to $\mathbb{X}$ is defined as $I(f;\mathbb{X}) - I(f;\mathbb{X}|Y)$, where $I(Y;\mathbb{X}|Y)$ is the conditional MI. With the assumption that the information is uniformly distributed over the region of a feature $f_j \in \mathbb{X}$, the feature redundancy of f with respect to $\mathbb{X}$ (i.e., $I(f;\mathbb{X}) - I(Y;\mathbb{X}|Y)$) can be estimated as

$$Redundancy(f,\mathbb{X}) = \sum_{f_j \in \mathbb{X}} \frac{I(Y;f_j)}{H(f_j)} I(f_j;f). \tag{5}$$

According to Definition 2, the redundancy between f and $\mathbb{X}$ is calculated as the sum of MI values (i.e., $I(f_j;f)$) between f and each feature $f_j \in \mathbb{X}$, where $\frac{I(Y;f_j)}{H(f_j)}$ is the weight given to each $I(f_j;f)$ to calculate the sum of MI values. This weight involves the class Y and it will be high if f_j is strongly related to Y.

Definition 3. (*Feature importance*) Let $\mathbb{X}$ be a feature subset, f be a feature ($f \notin \mathbb{X}$), Y be the class label,

and $\omega > 0$ be a weight parameter. The feature importance of f with respect to $\mathbb{X}$ is defined as

$$W(f, \mathbb{X}) = \frac{(Relevance(f))^{\omega}}{Redundancy(f, \mathbb{X})} = \frac{I(Y; f)^{\omega}}{\sum_{f_j \in \mathbb{X}} \frac{I(Y;f_j)}{H(f_j)} I(f_j; f)}. \tag{6}$$

According to Definition 3, the importance of a feature is calculated as the ratio of the weighted feature relevance to feature redundancy, where ω is a predefined parameter. This importance measure can evaluate the importance of a feature considering its both relevance and redundancy properties. A feature with a high degree of relevance and a low degree of redundancy will have a high importance value.

3.2. Clustering features

We propose a k-medoids algorithm to cluster features into different groups. The features strongly related to each other are clustered in the same group. This is beneficial for building an improvement strategy that effectively handles the feature redundancy problem.

The proposed k-medoids algorithm is shown in Algorithm 1. First, k-means++ (Arthur & Vassilvitskii, 2007) is used to initialize k medoids. During the iteration step of k-medoids, we update the medoid of each cluster to be the feature with the minimal sum of distances between it and other features in the same cluster. The uniqueness of the proposed k-medoids algorithm is that the property of FS is considered by using a new distance measure based on MI to evaluate feature similarities. The distance between two features f_1 and f_2 is defined as

$$d(f_1, f_2) = 1 - \frac{I(f_1; f_2)}{\min\{H(f_1), H(f_2)\}}. \tag{7}$$

In Eq. (7), $\frac{I(f_1;f_2)}{\min\{H(f_1),H(f_2)\}}$ is a normalized MI measure which scales the original MI value $I(f_1; f_2)$ into $[0, 1]$ so that the final distance value $d(f_1, f_2)$ is in $[0, 1]$. The proposed distance measure has a good property that the distance of a feature to itself is 0. The MI of a feature f with respect to itself is $I(f; f) = H(f)$. Therefore, according to Eq.(7), f's distance to itself is calculated as $1 - \frac{I(f;f)}{\min\{H(f),H(f)\}} = 0$. The k in k-medoids is a user-defined parameter denoting the number of clusters. The experience method is a simple strategy to estimate a desirable k value. It sets k as $\sqrt{N}$, where N is the number of features (Zhang et al., 2022b). In this paper, we set $k = \sqrt{N}$ for k-medoids with the experience method.

3.3. Improvement operations

In an FS task, a feature subset can be locally updated by three operations, *add*, *eliminate*, and *interchange* (Nguyen et al., 2016). For a feature subset $\mathbb{X}$, the add operation inserts a new feature $f \notin \mathbb{X}$ to $\mathbb{X}$, the eliminate operation removes a feature $f \in \mathbb{X}$ from $\mathbb{X}$, and the interchange operation replace a feature $f_1 \in \mathbb{X}$ with another feature $f_2 \notin \mathbb{X}$. In this section, we propose an improvement strategy with the three operations. This strategy considers feature relevance and redundancy to select the features to perform an improvement operation. In each operation, the feature space defined by a feature cluster (obtained by the

Algorithm 1: The proposed k-medoids algorithm for feature clustering.

Input : The training set $\mathcal{D}^{tr}$ which contains a set $\mathbb{F} = \{f_1, f_2, ..., f_N\}$ of features, the number of clusters k;
Output : Clusters $\mathbb{C}_1, \mathbb{C}_2, ..., \mathbb{C}_k$ with assigned features;

Initialize k cluster medoids $c_1, c_2, ..., c_k$ with $\mathcal{D}^{tr}$ using the k-means++ algorithm;
Calculate the distances between any two features in $\mathbb{F}$;
Assign each feature $f \in \mathbb{F}$ to the nearest cluster medoid and obtain k clusters $\mathbb{C}_1, \mathbb{C}_2, ..., \mathbb{C}_k$. (Cluster $\mathbb{C}_i$ contains the features assigned to the medoid c_i.) ;
repeat
    foreach *Cluster* $\mathbb{C}_i$ ($i \in \{1, 2, ..., k\}$) **do**
        Update the medoid c_i as the feature with the minimal sum of distances between it and other features in cluster $\mathbb{C}_i$;
    Assign each feature $f \in \mathbb{F}$ to the nearest cluster medoid and obtain k clusters $\mathbb{C}_1, \mathbb{C}_2, ..., \mathbb{C}_k$;
until *The clusters do not change*;
return *Clusters* $\mathbb{C}_1, \mathbb{C}_2, ..., \mathbb{C}_k$;

k-medoids algorithm) is considered. Specifically, features in the feature cluster are evaluated with the feature importance measure proposed in Section 3.1. The considered feature subset is updated according to the evaluation result so that it can hopefully update to one that has a higher relevance degree to the class and a lower redundancy degree among selected features. The detailed procedure of the proposed improvement operations, i.e., add, eliminate, and interchange, is given in Algorithm 2 and explained below.

The add operation: Let $\mathbb{C}_{p_i}$ be the considered feature cluster. The add operation first divides $\mathbb{C}_{p_i}$ into a selection set $\mathbb{C}_s$ and an elimination set $\mathbb{C}_e$, where features in $\mathbb{C}_s$ are selected by $\mathbb{X}$ and features in $\mathbb{C}_e$ are not selected by $\mathbb{X}$. The feature importance value $W(f, \mathbb{C}_s)$ (Eq. (6)) is calculated for each $f \in \mathbb{C}_e$ and the feature f_a with the maximum $W(f, \mathbb{C}_s)$ is added to $\mathbb{X}$. $W(f, \mathbb{C}_s)$ considers the relevance degree between f and the class as well as the redundancy degree between f and the features in $\mathbb{C}_s$. Therefore, an informative feature f_a with a high relevance degree to the class and a low redundancy degree to already selected features can be further added to $\mathbb{X}$.

The eliminate operation: Similar to the add operation, the considered feature cluster $\mathbb{C}_{p_i}$ is first divided into $\mathbb{C}_s$ and $\mathbb{C}_e$. Then, we remove the most undesirable feature $f_e \in \mathbb{C}_s$ with the minimum $W(f, \mathbb{C}_s \backslash \{f\})$. $W(f, \mathbb{C}_s \backslash \{f\})$ evaluates the relevance degree of f and its redundancy degree to remaining features in $\mathbb{C}_s$. Therefore, the eliminate operation can effectively remove an undesirable feature f_e from $\mathbb{X}$ considering feature relevance and redundancy.

The interchange operation: We propose the following heuristic procedure to perform the interchange operation. First, we determine the feature in $\mathbb{X}$ most likely to be unimportant with respect to $\mathbb{C}_{p_i}$. $W(f, \mathbb{C}_s \backslash \{f\})$ is calculated for each feature $f \in \mathbb{C}_s$ with Eq. (6). The feature f_e with the minimum $W(f, \mathbb{C}_s \backslash \{f\})$ is selected for the replacement. Next, we select the feature $f_a \in \mathbb{C}_e$ to replace f_e. The feature $f_a \in \mathbb{C}_e$ with the maximum importance value $W(f, \mathbb{C}_s \backslash \{f_e\})$ is selected for the replacement. Finally, the interchange operation is conducted by replacing f_e with f_a if $W(f_a, \mathbb{C}_s \backslash \{f_e\}) \geq W(f_e, \mathbb{C}_s \backslash \{f_e\})$. By interchanging f_a and f_e selected with the MI-based feature importance measure, $\mathbb{X}$ can be hopefully updated to

Algorithm 2: The add, eliminate, and interchange operations.

Input : A feature subset (solution) $\mathbb{X}$, a set of feature clusters $\{\mathbb{C}_1, \mathbb{C}_2, ..., \mathbb{C}_k\}$;
Output : Updated feature subset $\mathbb{X}$;

Function `addOp`($\mathbb{X}$, $\{\mathbb{C}_1, \mathbb{C}_2, ..., \mathbb{C}_k\}$)**:**
  $\{p_1, p_2, ..., p_k\} \leftarrow$ A random permutation of the integers from 1 to k;
  for $i \leftarrow 1$ *to* k **do**
    For each feature $f \in \mathbb{C}_{p_i}$, add f to $\mathbb{C}_s$ if $f \in \mathbb{X}$, otherwise add f to $\mathbb{C}_e$;
    if $\mathbb{C}_e \neq \emptyset$ **then**
      $\mathbb{X} \leftarrow \mathbb{X} \cup \{f_a = \arg\max_{f \in \mathbb{C}_e} W(f, \mathbb{C}_s)\}$;
      break;
  return $\mathbb{X}$;

Function `eliminateOp`($\mathbb{X}$, $\{\mathbb{C}_1, \mathbb{C}_2, ..., \mathbb{C}_k\}$)**:**
  $\{p_1, p_2, ..., p_k\} \leftarrow$ A random permutation of the integers from 1 to k;
  for $i \leftarrow 1$ *to* k **do**
    For each feature $f \in \mathbb{C}_{p_i}$, add f to $\mathbb{C}_s$ if $f \in \mathbb{X}$, otherwise add f to $\mathbb{C}_e$;
    if $\mathbb{C}_s \neq \emptyset$ **then**
      $\mathbb{X} \leftarrow \mathbb{X} \backslash \{f_e = \arg\min_{f \in \mathbb{C}_s} W(f, \mathbb{C}_s \backslash \{f\})\}$;
      break;
  return $\mathbb{X}$;

Function `interchangeOp`($\mathbb{X}$, $\{\mathbb{C}_1, \mathbb{C}_2, ..., \mathbb{C}_k\}$)**:**
  $\{p_1, p_2, ..., p_k\} \leftarrow$ A random permutation of the integers from 1 to k;
  for $i \leftarrow 1$ *to* k **do**
    For each feature $f \in \mathbb{C}_{p_i}$, add f to $\mathbb{C}_s$ if $f \in \mathbb{X}$, otherwise add f to $\mathbb{C}_e$;
    if $\mathbb{C}_s \neq \emptyset$ *and* $\mathbb{C}_e \neq \emptyset$ **then**
      $f_e = \arg\min_{f \in \mathbb{C}_s} W(f, \mathbb{C}_s \backslash \{f\})$, $f_a = \arg\max_{f \in \mathbb{C}_e} W(f, \mathbb{C}_s \backslash \{f_e\})$;
      if $W(f_a, \mathbb{C}_s \backslash \{f_e\}) \geq W(f_e, \mathbb{C}_s \backslash \{f_e\})$ **then**
        $\mathbb{X} \leftarrow \mathbb{X} \cup \{f_a\} \backslash \{f_e\}$;
        break;
  return $\mathbb{X}$;

one with a higher relevance degree to class and a lower redundancy degree of selected features.

4. Proposed optimization approach: NSGAII-MIIP

In this section, an MOEA called NSGAII-MIIP is proposed for the KPF selection model in Eq. (1). Except for genetic operations, NSGAII-MIIP applies an improvement phase with the proposed MI-guided improvement strategy to update solutions. The details of NSGAII-MIIP are given below.

4.1. Overall procedure

The overall procedure of NSGAII-MIIP is shown in Algorithm 3. NSGAII-MIIP obtains a set Ω of non-dominated solutions (i.e., non-dominated set) by optimizing the bi-objective KPF selection model. It is worth noting that, as suggested in Li et al. (2016), the ideal point method (IPM) is applied to select the best compromise solution $\mathbf{X}^*$ (i.e., the final KPF set) from Ω, which is beneficial for practical applications

Algorithm 3: The proposed NSGAII-MIIP algorithm.

Input : The training set $\mathcal{D}^{tr}$ (which contains M instances, a set $\mathbb{F} = \{f_1, f_2, ..., f_N\}$ of N features, and a class label Y), the population size S;
Output : The found solution (feature subset) $\mathbf{X}^*$;
Calculate the entropy $H(f)$ for each feature $f \in \mathbb{F}$ using Eq. (2);
Calculate the MI $I(f_1; f_2)$ between each pair of features $f_1 \in \mathbb{F} \cup \{Y\}$ and $f_2 \in \mathbb{F} \cup \{Y\}$ using Eq. (3);
Cluster features in $\mathbb{F}$ into k clusters $\{\mathbb{C}_1, \mathbb{C}_2, ..., \mathbb{C}_k\}$ based on the *k-medoids algorithm* in Algorithm 1;
Set the iteration counter as $t = 0$ and set the no-dominated set as $\Omega = \emptyset$;
Initialize a population $\mathbb{P}^0 = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_S\}$ of S solutions;
Evaluate the objective function values for each $\mathbf{X} \in \mathbb{P}^0$ with $\mathcal{D}^{tr}$ using Eq. (1);
Rank the solutions in $\mathbb{P}^0$ based on the *fast non-dominated sorting approach* and the *crowding distance measure* (see Section 4.4);
repeat
    Generate offspring population $\mathbb{P}'$ based on $\mathbb{P}^t$ using genetic operators;
    Generate ranking pool $\mathbb{R}$ by combining $\mathbb{P}'$ and $\mathbb{P}^t$ (i.e., $\mathbb{R} = \mathbb{P}' \cup \mathbb{P}^t$);
    Obtain the offspring population $\mathbb{P}^{t+1}$ based on $\mathbb{R}$ using the *improvement-phase-embedded ranking approach* (see Algorithm 4), where a set Γ of new solutions are further generated by the improvement strategy and then ranked together with $\mathbb{R}$;
    $t \leftarrow t + 1$;
until *termination condition*;
Add non-dominated solutions in $\mathbb{P}^t$ to Ω;
Select the best compromise solution $\mathbf{X}^*$ from Ω using the *IPM*.
return $\mathbf{X}^*$;

of the KPF selection method. IPM first defines an ideal point in the objective space based on Ω and then selects the solution closest to the ideal point as the best compromise solution.

NSGAII-MIIP is built based on NSGA-II with some new components to improve the FS performance. First, the entropies of features and the MI between any two features are evaluated on the training set $\mathcal{D}^{tr}$ using Eqs. (2) and (3). The entropy and MI results will be used by k-medoids and the improvement phase. Then, the proposed k-medoids algorithm is used to categorize the features into k clusters $\{\mathbb{C}_1, \mathbb{C}_2, ..., \mathbb{C}_k\}$, which are required by the improvement operations. Next, during the iteration process, the genetic operators are used to generate the offspring population $\mathbb{P}'$, which are then combined with the parent population $\mathbb{P}^t$ to construct the ranking pool $\mathbb{R}$. After that, the solutions in $\mathbb{R}$ are ranked so that a high-quality population $\mathbb{P}^{t+1}$ can be obtained. Different from NSGA-II, we propose an improvement-phase-embedded solution ranking approach in NSGAII-MIIP. This ranking approach applies an improvement phase within the ranking process to purify current non-dominated solutions. Specifically, the MI-guided improvement strategy is used to generate a set Γ of new solutions based on the non-dominated solutions in $\mathbb{R}$ once the ranking approach finds the non-dominated solutions from $\mathbb{R}$, and then the ranking process continues to rank solutions in the union of $\mathbb{R}$ and Γ. Finally, after the iteration process, NSGAII-MIIP outputs the non-dominated solutions, from which the IPM is applied to obtain the final solution $\mathbf{X}^*$(KPF set).

4.2. Solution representation and initialization

Let the original feature set be $\mathbb{F} = \{f_1, f_2, ..., f_N\}$. A solution (feature subset) $\mathbb{X}$ for the KPF selection problem is encoded as a binary vector $\mathbf{X} = (x_1, x_2, ..., x_N)$, where each $x_i \in \{0, 1\}(i = 1, 2, ..., N)$. $x_i = 1$ or $x_i = 0$ denotes the ith feature is selected or not selected by $\mathbf{X}$.

The opposition-based learning (OBL) concept (Tizhoosh, 2005) has shown to have good performance in generating a set of diversified solutions. Therefore, we adopt an OBL-based strategy in NSGAII-MIIP to initialize population $\mathbb{P}^0$ of S solutions. First, 50% of solutions are randomly initialized. Then, the opposite solutions for these initialized solutions are generated as the remaining 50% of solutions of the population. The opposite solution of a solution $\mathbf{X} = (x_1, x_2, ..., x_N)$ is defined as $\tilde{\mathbf{X}} = (1 - x_1, 1 - x_2, ..., 1 - x_N)$.

4.3. Genetic operators

The binary tournament selection is used to select S parents from $\mathbb{P}^t$. These selected parents are grouped into $S/2$ pairs to perform the crossover and mutation operations.

A modified single point crossover operator (Li et al., 2020a) is used in NSGAII-MIIP. Let $\mathbf{X}_a = (x_{a1}, x_{a2}, ..., x_{aN})$ and $\mathbf{X}_b = (x_{b1}, x_{b2}, ..., x_{bN})$ be two paired parents. A index set $\Phi = \{i_1, i_2, ..., i_d\}$ is first obtained based on $\mathbf{X}_a$ and $\mathbf{X}_b$, where $x_{ai} \neq x_{bi}$ for each $i \in \Phi$. Then, a crossover point is generated as $i_c \in \Phi$, where c is a random integer in $[2, d]$. Next, the offspring are generated by the crossover operation as $\mathbf{X}_a^c = (x_{a1}, x_{a2}, ..., x_{a(i_c-1)}, x_{bi_c}, ..., x_{bN})$ and $\mathbf{X}_b^c = (x_{b1}, x_{b2}, ..., x_{b(i_c-1)}, x_{ai_c}, ..., x_{aN})$. Compared with the standard single point crossover operator, the modified one uses a set $\Phi = \{i_1, i_2, ..., i_d\}$ to record at which index the two parents have different values. As the crossover point is generated from the set Φ, the generated offspring are different from their parents, which guarantees the success of the crossover operation.

The subset size-oriented mutation (SSOM) operator (Oh et al., 2004) designed for FS is adopted in NSGAII-MIIP to update solutions after crossover further. In SSOM, the mutation rate for the elements of 1 is different from that of 0 for a solution $\mathbf{X}^c$. The mutation rate p_m for the elements of 1 is a user-defined constant, while a different mutation rate is used for the elements of 0. Assume that the number of elements of 1 and 0 is $N_1(\mathbf{X}^c)$ and $N_0(\mathbf{X}^c)$ for $\mathbf{X}^c$. The mutation rate of elements of 0 is calculated as $p_{m0} = \frac{N_1(\mathbf{X}^c)}{N_0(\mathbf{X}^c)} \cdot p_m$. The SSOM operator has a good property that the expected number of selected features after mutation does not change, which is beneficial for building an effective EC-based FS method.

4.4. The improvement-phase-embedded ranking approach

In NSGAII-MIIP, the fast non-dominated sorting approach and crowding distance measure (Deb et al., 2002) are used to rank the solutions in the initial population as shown in Line 7 of Algorithm 3. The fast non-dominated sorting approach uses an iterative process to divide the solutions in the ranking pool $\mathbb{R}$ into different non-dominated fronts using the Pareto dominance concept, indicating different levels of solutions' goodness. For two solutions $\mathbf{X}_1$ and $\mathbf{X}_2$, $\mathbf{X}_1$ is better than $\mathbf{X}_2$ if the former is in a lower non-dominated

front. In the ranking process, the non-dominated solutions in $\mathbb{R}$ are assigned to a non-domination rank $Rank(\mathbf{X}) = 1$, which means these solutions are added to the first non-dominated front $\mathcal{F}_1$. Then, the non-dominated solutions for remaining solutions in the ranking pool (i.e., $\{\mathbb{R} \backslash \mathcal{F}_1\}$) are assigned to a $Rank(\mathbf{X}) = 2$ and added to $\mathcal{F}_2$. The above steps continue until all solutions in $\mathbb{R}$ are ranked. The fast non-dominated sorting approach uses a domination count $n(\mathbf{X})$ recording the number of solutions dominating $\mathbf{X}$ and a domination set $\mathbb{S}(\mathbf{X})$ recording the solutions dominated by $\mathbf{X}$. A $n(\mathbf{X}) = 0$ means that $\mathbf{X}$ is a non-dominated solution. The $\mathbb{S}(\mathbf{X})$ can be used to update the domination count of solutions after eliminating current non-dominated solutions from $\mathbb{R}$ so that the solutions in the next front can be found. To further compare the solutions in the same non-dominated front, a crowding distance measure is used. This measure estimates the density of solutions around each solution, and a larger crowding distance denotes a better solution quality. Therefore, with the fast non-dominated sorting approach and the crowding distance measure, all solutions in the population can be ranked.

In this paper, an improvement phase is used in NSGAII-MIIP to purify non-dominated solutions found so far further and generate a set Γ of new solutions. The solutions in Γ are ranked together with solutions in the original ranking pool $\mathbb{R}$. Based on the ranked solutions, we can select the best S solutions to compose the population of the next generation. The improvement phase requires the information of non-dominated solutions in the population. Intuitively, we can first use the fast non-dominated sorting approach to obtain non-dominated solutions in $\mathbb{R}$ and then generate Γ by the improvement operations. After that, another ranking process can be applied on $\Gamma \cup \mathbb{R}$. However, this straightforward strategy requires two rounds of the ranking process, which improves the time complexity substantially. To address this problem, we propose an improvement-phase-embedded ranking approach. The improvement operations proposed in Section 3.3 are applied to generate new solutions Γ once the ranking process finds the non-dominated solutions from $\mathbb{R}$. The solutions in Γ are then combined with the original solutions in $\mathbb{R}$ to continue the ranking process.

The procedure of the improvement-phase-embedded ranking approach is shown in Algorithm 4. First, the domination count $n(\mathbf{X})$ and domination set $\mathbb{S}(\mathbf{X})$ are obtained for each solution $\mathbf{X}$ in the ranking pool $\mathbb{R}$, which is the same as the fast non-dominated sorting approach. The non-dominated solutions with $n(\mathbf{X}) = 0$ are appended to $\mathcal{F}_1$. Then, we apply the add, eliminate, and interchange operations shown in Algorithm 2 on each $\mathbf{X} \in \mathcal{F}_1$ to generate a set Γ of new solutions. Next, we obtain $n(\mathbf{X})$ and $\mathbb{S}(\mathbf{X})$ for each $\mathbf{X} \in \mathbb{R}^{new}$ ($\mathbb{R}^{new} = \mathbb{R} \cup \Gamma$), which are required for ranking the solutions in $\mathbb{R}^{new}$. Because we have already obtained $n(\mathbf{X})$ and $\mathbb{S}(\mathbf{X})$ in terms of $\mathbb{R}$ by comparing any two solutions in $\mathbb{R}$ as shown in Line 1 of Algorithm 4, $n(\mathbf{X})$ and $\mathbb{S}(\mathbf{X})$ for $\mathbf{X} \in \mathbb{R}^{new}$ can be obtained by just further comparing solutions between Γ and $\mathbb{R}$ and by comparing solutions within Γ as shown in Lines 10 to 20 of Algorithms 4. This means that we do not need to thoroughly compare any two solutions in $\mathbb{R}^{new}$ to obtain $n(\mathbf{X})$ and $\mathbb{S}(\mathbf{X})$. After that, we can obtain the non-domination rank $Rank(\mathbf{X})$ for $\mathbf{X} \in \mathbb{R}^{new}$ with the obtained $n(\mathbf{X})$ and $\mathbb{S}(\mathbf{X})$. Finally, we can rank the solutions $\mathbf{X} \in \mathbb{R}^{new}$ with the obtained $Rank(\mathbf{X})$ and the crowding distance value $cd(\mathbf{X})$ and output the

Algorithm 4: The improvement-phase-embedded ranking approach.

Input : The training set $\mathcal{D}^{tr}$, the ranking pool $\mathbb{R}$, a set of feature clusters $\{\mathbb{C}_1, \mathbb{C}_2, ..., \mathbb{C}_k\}$;
Output : The population $\mathbb{P}^{new}$;

Obtain domination count $n(\mathbf{X})$ and domination set $\mathbb{S}(\mathbf{X})$ for each $\mathbf{X} \in \mathbb{R}$;
$\mathcal{F}_1 \leftarrow \{\mathbf{X} | n(\mathbf{X}) = 0, \mathbf{X} \in \mathbb{R}\}$; /* Obtain the non-dominated set for $\mathbb{R}$ */
$\Gamma_a \leftarrow \emptyset$, $\Gamma_e \leftarrow \emptyset$, $\Gamma_i \leftarrow \emptyset$;
/* The MI-guided improvement phase */
foreach $\mathbf{X} \in \mathcal{F}_1$ **do**
    $\Gamma_a \leftarrow \Gamma_a \cup \{\mathbf{X}^{new} =$ `addOp(`$\mathbf{X}$, $\{\mathbb{C}_1, \mathbb{C}_2, ..., \mathbb{C}_k\}$`)`$\}$; /* See Algorithm 2 */
    $\Gamma_e \leftarrow \Gamma_e \cup \{\mathbf{X}^{new} =$ `eliminateOp(`$\mathbf{X}$, $\{\mathbb{C}_1, \mathbb{C}_2, ..., \mathbb{C}_k\}$`)`$\}$; /* See Algorithm 2 */
    $\Gamma_i \leftarrow \Gamma_i \cup \{\mathbf{X}^{new} =$ `interchangeOp(`$\mathbf{X}$, $\{\mathbb{C}_1, \mathbb{C}_2, ..., \mathbb{C}_k\}$`)`$\}$; /* See Algorithm 2 */
$\Gamma \leftarrow \Gamma_a \cup \Gamma_e \cup \Gamma_i$, $\mathbb{R}^{new} \leftarrow \mathbb{R} \cup \Gamma$;
Evaluate objective function values for each solution in Γ with $\mathcal{D}^{tr}$ using the optimization model in Eq. (1);
/* Update the domination count $n(\mathbf{X})$ and set $\mathbb{S}(\mathbf{X})$ for each $\mathbf{X} \in \mathbb{R}^{new}$ */
foreach $\mathbf{X} \in \Gamma$ **do**
    $\mathbb{S}(\mathbf{X}) \leftarrow \emptyset$, $n(\mathbf{X}) \leftarrow 0$;
    foreach $\mathbf{Z} \in \mathbb{R} \cup \Gamma$ **do**
        if $\mathbf{X} \prec \mathbf{Z}$ **then** /* $\mathbf{X}$ dominates $\mathbf{Z}$ */
            $\mathbb{S}(\mathbf{X}) \leftarrow \mathbb{S}(\mathbf{X}) \cup \mathbf{Z}$; /* Update domination sets for solutions from improvement phase */
            if $\mathbf{Z} \in \mathbb{R}$ **then**
                $n(\mathbf{Z}) \leftarrow n(\mathbf{Z}) + 1$; /* Update domination counts for original solutions in $\mathbb{R}$ */
        else if $\mathbf{Z} \prec \mathbf{X}$ **then** /* $\mathbf{Z}$ dominates $\mathbf{X}$ */
            $n(\mathbf{X}) \leftarrow n(\mathbf{X}) + 1$; /* Update domination counts for solutions from improvement phase */
            if $\mathbf{Z} \in \mathbb{R}$ **then**
                $\mathbb{S}(\mathbf{Z}) \leftarrow \mathbb{S}(\mathbf{Z}) \cup \mathbf{X}$; /* Update domination sets for original solutions in $\mathbb{R}$ */
Obtain non-domination rank $Rank(\mathbf{X})$ for each $\mathbf{X} \in \mathbb{R}^{new}$ using the information of $n(\mathbf{X})$ and $\mathbb{S}(\mathbf{X})$;
$cd(\mathbf{X}) \leftarrow$ Calculate the crowding distance for each $\mathbf{X} \in \mathbb{R}^{new}$;
$\mathbb{R}^{new} \leftarrow$ Rank the solutions in $\mathbb{R}^{new}$ based on the non-domination rank $Rank(\mathbf{X})$ and crowding distance $cd(\mathbf{X})$ of solutions;
$\mathbb{P}^{new} \leftarrow$ The first S solutions in the sorted $\mathbb{R}^{new}$;
return $\mathbb{P}^{new}$;

best S solutions to the next generation.

It is worth noting that as the proposed improvement strategy is a general solution purifying procedure, it can be as well embedded in typical MOEAs, such as MOEA/D (Zhang & Li, 2007) and SPEA2 (Zitzler et al., 2001), to purify non-dominated solutions during the iterations. The reason we selected NSGA-II as the base algorithm to build NSGAII-MIIP is because the sequential solution ranking mechanism of NSGA-II makes it easy to embed the improvement phase in the ranking process without additional efforts as we analyzed. It would be interesting to investigate efficient strategies to embed the proposed improvement strategy in other typical MOEAs in the future.

4.5. Time complexity

Being a wrapper-based FS method, the objective function evaluation of feature subsets is one of the major time-consuming parts of NSGAII-MIIP. However, theoretically estimating the function evaluation

time is hard because the it is determined by the learning algorithm involved in the wrapper method and the cardinality of the feature subset. Therefore, we evaluate the overall time complexity of NSGAII-MIIP in Algorithm 3 without considering the function evaluation time. A detailed analysis of the time complexity of NSGAII-MIIP is shown in the supplementary material. As analyzed in the supplementary material, the three parts that contribute to the time complexity of NSGAII-MIIP are: a) the calculation of entropy and MI, b) the k-medoid algorithm, and c) the genetic operations and the improvement-phase-embedded ranking approach. The time complexities of the three parts are $O(MN)+O(MN^2)$, $O(N^2)$, and $O(TSN)+O(TVS^2)$ respectively, where M is the number of instances in the training set, N is the number of features in $\mathbb{F}$, T is the number of iterations of NSGAII-MIIP, S is the population size, and V is the number of objectives ($V = 2$ in the KPF selection model). Thus, the overall time complexity of NSGAII-MIIP is $[O(MN) + O(MN^2)] + O(N^2) + [O(TSN) + O(TVS^2)] \cong O(MN^2) + O(TSN) + O(TVS^2)$. It is worth noting that the time complexity of the proposed ranking approach is $O(VS^2)+O(SN)$, which is composed of the complexity of the improvement operations ($O(SN)$) and the complexity of solution ranking ($O(VS^2)$). This means the solution ranking part of the proposed ranking approach has the same complexity as the fast non-dominated sorting approach. This is because even the improvement phase breaks the solution ranking process into two parts, no additional operations are required to rank solutions in $\mathbb{R}^{new}$ compared with directly ranking a set of solutions with the same size as $\mathbb{R}^{new}$. This shows the efficiency of the proposed ranking approach.

5. Experimental design

In this section, the datasets, benchmark algorithms, and experimental settings to verify the FS performance of the proposed NSGAII-MIIP algorithm are presented. The source code of NSGAII-MIIP is available at `https://github.com/andali89/NSGAII-MIIP`.

5.1. Datasets

We use four CMP datasets, LATEX (Gauchi & Chagnon, 2001), ADPN (Gauchi & Chagnon, 2001), SPIRA (Gauchi & Chagnon, 2001), and PAPER (Wold et al., 2001), to verify the proposed NSGAII-MIIP algorithm. The data of LATEX are taken from the manufacturing process of latex. The monomer input rate, temperature, level time, catalyst level, reactive concentration, etc., are the PFs. The data of ADPN are taken from the manufacturing process of adiponitrile, which is an intermediary of Nylon 6-6. The flow, pressure, temperature, etc., are the PFs. The data of SPIRA are taken from the fermentation process for manufacturing an antibiotic called Spiramycine. The oxygen consumption peaks, stirring power, temperature level, etc., are the PFs. The four datasets have a continuous quality variable measuring the quality of products (instances). Based on the thresholds given in the original studies, Anzanello et al. (2012) converted the quality variable of these datasets into a discrete class label, which indicates different quality

Table 1: Details of the datasets.

Dataset	#Features (PFs)	#Instances	#Positives (premium quality)	#Negatives (regular quality)
LATEX	117	262	78	184
ADPN	100	71	20	51
SPIRA	96	145	50	95
PAPER	54	384	33	351
LATEX-F	176	262	78	184
ADPN-F	150	71	20	51
SPIRA-F	144	145	50	95
PAPER-F	81	384	33	351

levels of the products, i.e., premium quality (+1) and regular quality (-1). Because the proposed method is designed on classification models, the converted datasets are used in the experiments.

To comprehensively evaluate the KPF selection performance of NSGAII-MIIP, we generate four synthetic datasets, referred to as LATEX-F, ADPN-F, SPIRA-F, and PAPER-F, by adding noisy PFs following the standard Gaussian distribution $N(0, 1)$ to LATEX, ADPN, SPIRA, and PAPER. The number of features in synthetic datasets increased by 50% compared with the original ones. Details of the original CMP datasets and the synthetic datasets used in the experiments are shown in Table 1.

5.2. Benchmark algorithms

We use eight multi-objective FS methods, which include GADMS-IPM (Li et al., 2020a), MOPSO-LS (He et al., 2022), MOFS-BDE (Zhang et al., 2020), NSGAII-IPM (Li et al., 2016), IDMS-IPM (Li et al., 2019), NSGA-II/SDR (Tian et al., 2019), MOEA/D (Zhang & Li, 2007), and SPEA2 (Zitzler et al., 2001), as the benchmark algorithms.

GADMS-IPM (Li et al., 2020a) combines a GA with direct multi-search (DMS) for KPF selection. The DMS step in GADMS-IPM is an improvement step updating the non-dominated solutions during the iteration process. MOPSO-LS (He et al., 2022) is a multi-objective PSO-based KPF selection method. An improvement step randomly adding and eliminating a feature for each non-dominated solution is used. MOFS-BDE (Zhang et al., 2020) is a multi-objective binary DE algorithm proposed for FS. It adopts an improvement step called one-bit purifying to update non-dominated solutions. NSGAII-IPM (Li et al., 2016) is a KPF selection method based on a modified NSGA-II algorithm, in which an improved sorting approach is used to handle duplicate solutions to improve population diversity. IDMS-IPM (Li et al., 2019) improves DMS by adding a mutation operation to solve the KPF selection problems. GADMS-IPM, MOPSO-LS, and IDMS-IPM adopt the same FS model as NSGAII-MIIP which maximizes the GM measure and minimizes the number of selected features. In comparison, MOFS-BDE and NSGAII-IPM aim to maximize accuracy and minimize the number of selected features.

Moreover, the MOEAs including NSGA-II/SDR (Tian et al., 2019), MOEA/D (Zhang & Li, 2007), and SPEA2 (Zitzler et al., 2001) are used to build three benchmark FS methods. NSGA-II/SDR is a recently proposed variant of NSGA-II that adopts a strengthened dominance relation considering both convergence and diversity to rank solutions. MOEA/D and SPEA2 are popular MOEAs widely applied to different

optimization tasks. The three MOEAs are applied to the KPF selection model defined in Eq. (1). The genetic operators used in the three MOEAs are the commonly used binary tournament selection, single-point crossover, and bitwise mutation.

In NSGAII-IPM, GADMS-IPM, MOPSO-LS, NSGAII-IPM and IDMS-IPM, the IPM was used to obtain the final KPF set from the non-dominated solutions found by these MOO algorithms. For a comparison purpose, in MOFS-BDE, NSGAII/SDR, MOEA/D, and SPEA2, the IPM is used as well to find the final KPF set. The final KPF sets obtained by these algorithms are used to evaluate the KPF selection performance.

5.3. Experimental settings

The experiments are conducted on an Intel Core PC with a 3.6 GHz CPU and a 16 GB main memory. The main procedure of NSGAII-MIIP and benchmark algorithms is implemented in MATLAB. All FS methods adopt the NB classifier (John & Langley, 1995) as the learning model, which is implemented in the Weka machine learning platform (Hall et al., 2009). Note that PAPER and PAPER-F are highly unbalanced because their imbalance ratio (proportion of negative instances to positive instances) is over 10, which is much higher than the imbalance ratios of other datasets. The high imbalance ratio can result in an unreliable trained classifier. To address this issue, as suggested in Li et al. (2020a), we modify the training process of the NB classifier for PAPER and PAPER-F, where the training sets used to build the NB classifier are balanced by duplicating the negative instances for $[\#Negatives/\#Positives] - 1$ times.

We apply the 10-fold stratified CV method to verify the FS methods. A 10-fold CV generates 10 pairs of training and test sets, which yield 10 experimental runs. In each run, a training set is input to the FS methods to find the KPFs, with which a learning model is built. The classification results of the learning model on the paired test set are used to verify the effectiveness of selected KPFs. Since NSGAII-MIIP and benchmark algorithms are stochastic algorithms, we repeat 10-fold CV 3 times, which yields $3 \times 10 = 30$ experimental runs. Moreover, in all these algorithms, accuracy or GM is adopted as one objective function of the FS model to evaluate the goodness of a feature subset. The inner-loop 5-fold CV process is adopted on the training set to estimate the value of accuracy or GM of a feature subset as introduced in Section 2.

The parameter settings of NSGAII-MIIP and benchmark algorithms are given as follows. In NSGAII-MIIP, we set crossover and mutation rates as $p_c = 0.9$ and $p_m = 1/N$ (N is the number of original features) as suggested in Li et al. (2020a). The weight parameter in the feature importance measure of NSGAII-MIIP is set as $\omega = 2$ based on the tuning experiments (see supplementary material for details). Moreover, to facilitate the calculation of entropy or MI in the feature importance measure, the numeric features in the training set are discretized into 10 bins (the default setting in Weka) of uniform width. The discretized features are just used for feature importance calculation. In GADMS-IPM, as suggested in Li et al. (2020a), the crossover and mutation rates are set as $p_c = 0.9$ and $p_m = 1/N$, the initial step size value is set as $\alpha_0 = 1$, and the step size updating parameter is set as $\beta = 0.9$. In MOPSO-LS, the inertia weight and

accelerate constants are set as $\omega = 0.7298$ and $c_1 = c_2 = 1.49618$, the size of neighbors is set as $T = 10$, and the mutation rate is set as $p_m = 1/N$ (He et al., 2022). In MOFS-BDE, the basic crossover rate is set as 0.4, the scale factor is set as $F = 0.5 * rand$, the one-bit purifying search is conducted every 5 iterations, and the turbulence coefficient is set as $\sigma = 0.01$ (Zhang et al., 2020). In IDMS-IPM, the initial step size is set as $\alpha^0 = 2$, the parameters to control the change of step size parameter are set as $\beta_1 = \beta_2 = 0.95$ and $\gamma = 1$, and the mutation rate is set as $p_m = 1/N$ (Li et al., 2019). In NSGAII-IPM, NSGAII/SDR, MOEA/D, and SPEA2, the crossover and mutation rates are set as $p_c = 0.9$ and $p_m = 1/N$ as suggested in Li et al. (2016). Moreover, the number of neighbors in MOEA/D is set as $T = 10$ (Zhang & Li, 2007). For all these algorithms, the population/swarm size is set as 100, and the stopping criterion is set as 5,000 objective function evaluations. The number of function evaluations to stop the FS methods is smaller than that in previous KPF selection studies (Li et al., 2020a; He et al., 2022), where 10,000 function evaluations were used as the stopping criterion. The reason for this setting is to test if the proposed NSGAII-MIIP algorithm can quickly obtain good KPF selection results with relatively limited computational resources, which is our aim in designing the MI-guided improvement strategy for NSGAII-MIIP.

Performance metrics to evaluate the FS performance include GM, the F1-score, the area under the curve (AUC), and the number of selected KPFs (#KPFs). GM, the F1-score, and AUC are common classification performance measures for unbalanced data that evaluate the KPFs' predictive performance for product quality. They are the higher the better. The F1-score is defined as the harmonic mean of precision and recall. AUC measures the area under the ROC (receiver operating characteristic) curve, which draws the true positive rate against the false positive rate on different classification thresholds. An FS method with good classification performance while obtaining a small #KPFs reveals that it is effective in eliminating irrelevant and redundant PFs to maintain real KPFs with good quality prediction performance.

6. KPF selection results

6.1. Performance comparison on the original CMP datasets

Table 2 shows the performance metric results obtained by NSGAII-MIIP and benchmark algorithms on the four original CMP datasets, i.e., ADPN, LATEX, PAPER, and SPIRA. In the table, the mean and standard deviation (std.) (mean $\pm$ std.) of the performance metrics (on the test set) over the 30 experimental runs are shown. The Wilcoxon signed-rank test (Wilcoxon, 1945) is used to compare NSGAII-MIIP with benchmark algorithms. The sign $\Uparrow\Uparrow$ ($\Downarrow\Downarrow$) denotes that NSGAII-MIIP obtains a significantly better (worse) result at a significance level of 0.05, and $\uparrow$ ($\downarrow$) is used to denote the significant test results at a significance level of 0.1. The best mean result on a dataset is highlighted in bold for each performance metric.

In terms of classification performance, NSGAII-MIIP generally obtains better or similar results on GM, F1-score, and AUC compared to benchmark algorithms. On ADPN, NSGAII-MIIP obtains slightly lower

Table 2: Comparisons of KPF selection performance between NSGAII-MIIP and benchmark algorithms on the four original CMP datasets.

Dataset	Metric	NSGAII-MIIP	GADMS-IPM	MOPSO-LS	MOFS-BDE	NSGAII-IPM	IDMS-IPM	NSGAII/SDR	MOEA/D	SPEA2
ADPN	GM (%)	76.48 ± 23.46	76.41 ± 18.10	73.92 ± 23.15	$\mathbf{77.54 \pm 14.49}$	69.79 ± 24.48	$69.46 \pm 26.34 \uparrow$	69.52 ± 29.60	73.23 ± 30.54	$68.17 \pm 29.94 \uparrow\uparrow$
	F1-score (%)	69.11 ± 23.16	67.79 ± 17.45	66.44 ± 22.83	$\mathbf{69.97 \pm 15.94}$	62.51 ± 23.97	61.79 ± 26.05	62.00 ± 28.15	67.00 ± 28.97	$60.89 \pm 28.66 \uparrow$
	AUC (%)	83.44 ± 26.21	82.83 ± 20.40	80.44 ± 22.13	85.08 ± 14.25	$77.76 \pm 22.67 \uparrow$	81.78 ± 18.13	84.00 ± 15.37	$\mathbf{88.83 \pm 11.90}$	82.61 ± 22.71
	#KPFs	$\mathbf{2.4 \pm 0.5}$	$3.1 \pm 1.3 \uparrow\uparrow$	2.6 ± 0.7	$3.0 \pm 0.5 \uparrow\uparrow$	$6.6 \pm 1.5 \uparrow\uparrow$	$25.6 \pm 5.3 \uparrow\uparrow$	$2.8 \pm 1.1 \uparrow$	$5.2 \pm 1.7 \uparrow\uparrow$	$7.0 \pm 2.1 \uparrow\uparrow$
LATEX	GM (%)	$\mathbf{77.47 \pm 10.46}$	75.59 ± 10.46	77.20 ± 9.68	73.76 ± 15.35	$72.70 \pm 11.25 \uparrow$	76.96 ± 8.88	74.81 ± 11.18	75.99 ± 9.14	76.73 ± 10.03
	F1-score (%)	$\mathbf{68.75 \pm 13.17}$	66.30 ± 13.32	67.82 ± 12.17	64.83 ± 18.99	$63.05 \pm 14.07 \uparrow$	68.07 ± 11.68	65.46 ± 14.50	66.55 ± 11.09	67.39 ± 12.59
	AUC (%)	$\mathbf{88.02 \pm 6.76}$	$83.63 \pm 6.85 \uparrow\uparrow$	$85.31 \pm 7.06 \uparrow\uparrow$	86.52 ± 7.34	$82.44 \pm 7.20 \uparrow\uparrow$	$83.65 \pm 6.82 \uparrow\uparrow$	$84.24 \pm 7.14 \uparrow\uparrow$	$85.40 \pm 6.39 \uparrow$	$84.42 \pm 6.77 \uparrow\uparrow$
	#KPFs	$\mathbf{3.8 \pm 0.9}$	$6.7 \pm 2.2 \uparrow\uparrow$	4.2 ± 1.3	$6.3 \pm 1.4 \uparrow\uparrow$	$13.5 \pm 2.8 \uparrow\uparrow$	$38.0 \pm 3.3 \uparrow\uparrow$	$5.5 \pm 3.9 \uparrow\uparrow$	$7.9 \pm 2.7 \uparrow\uparrow$	$13.8 \pm 3.0 \uparrow\uparrow$
PAPER	GM (%)	87.91 ± 7.14	87.73 ± 8.68	88.49 ± 7.99	$73.52 \pm 25.57 \uparrow\uparrow$	$76.71 \pm 18.64 \uparrow\uparrow$	86.92 ± 8.37	$\mathbf{88.66 \pm 7.46}$	87.37 ± 7.84	86.91 ± 9.66
	F1-score (%)	55.87 ± 12.47	56.52 ± 13.10	56.54 ± 12.74	51.89 ± 22.09	51.00 ± 17.33	$\mathbf{57.67 \pm 14.72}$	55.03 ± 12.56	56.56 ± 13.26	54.47 ± 12.38
	AUC (%)	$\mathbf{92.67 \pm 4.98}$	92.06 ± 5.53	91.61 ± 6.70	$89.90 \pm 7.15 \uparrow\uparrow$	$87.76 \pm 10.38 \uparrow\uparrow$	$90.28 \pm 9.05 \uparrow\uparrow$	92.25 ± 5.64	91.44 ± 7.08	91.55 ± 6.27
	#KPFs	3.1 ± 0.5	3.0 ± 0.7	$2.8 \pm 0.7 \downarrow$	$4.7 \pm 0.8 \uparrow\uparrow$	$4.7 \pm 1.2 \uparrow\uparrow$	$6.2 \pm 2.6 \uparrow\uparrow$	$\mathbf{2.7 \pm 0.7} \downarrow\downarrow$	$3.6 \pm 0.8 \uparrow\uparrow$	3.4 ± 1.3
SPIRA	GM (%)	$\mathbf{81.26 \pm 8.84}$	$73.16 \pm 12.61 \uparrow\uparrow$	$74.79 \pm 11.42 \uparrow\uparrow$	$73.43 \pm 10.83 \uparrow\uparrow$	$68.42 \pm 18.06 \uparrow\uparrow$	$69.24 \pm 19.84 \uparrow\uparrow$	$71.45 \pm 16.48 \uparrow\uparrow$	$70.34 \pm 9.44 \uparrow\uparrow$	$73.83 \pm 12.30 \uparrow\uparrow$
	F1-score (%)	$\mathbf{76.78 \pm 10.36}$	$66.11 \pm 15.61 \uparrow\uparrow$	$68.92 \pm 14.39 \uparrow\uparrow$	$66.95 \pm 13.26 \uparrow\uparrow$	$61.57 \pm 19.34 \uparrow\uparrow$	$62.34 \pm 21.99 \uparrow\uparrow$	$65.44 \pm 16.95 \uparrow\uparrow$	$62.87 \pm 11.76 \uparrow\uparrow$	$66.96 \pm 14.90 \uparrow\uparrow$
	AUC (%)	$\mathbf{86.79 \pm 9.51}$	$83.20 \pm 9.59 \uparrow$	$82.85 \pm 10.77 \uparrow\uparrow$	$79.79 \pm 9.52 \uparrow\uparrow$	$81.76 \pm 11.33 \uparrow\uparrow$	83.78 ± 12.50	$82.68 \pm 10.37 \uparrow\uparrow$	$81.75 \pm 10.67 \uparrow\uparrow$	85.17 ± 9.67
	#KPFs	$\mathbf{3.1 \pm 0.8}$	$4.0 \pm 1.0 \uparrow\uparrow$	3.5 ± 1.4	3.6 ± 1.3	$7.3 \pm 1.8 \uparrow\uparrow$	$25.9 \pm 2.8 \uparrow\uparrow$	4.0 ± 3.2	$5.5 \pm 2.0 \uparrow\uparrow$	$8.2 \pm 2.4 \uparrow\uparrow$

(mean) GM, F1-score, and AUC values than MOFS-BDE, and a lower AUC value than NSGAII/SDR and MOEA/D. In other cases, NSGAII-MIIP obtains higher GM, F1-score, and AUC values than the benchmark algorithms. On LATEX, NSGAII-MIIP obtains higher GM, F1-score, and AUC values than all benchmark algorithms, and the AUC value of NSGAII-MIIP is significantly higher than the benchmark algorithms except for MOFS-BDE. On PAPER, NSGAII-MIIP obtains significantly higher GM and AUC values than MOFS-BDE and NSGAII-IPM, obtains a significantly higher AUC value than IDMS-IPM, and obtains similar results in other cases. On SPIRA, NSGAII-MIIP performs much more effectively than the benchmark algorithms as NSGAII-MIIP obtains significantly higher GM, F1-score, and AUC values in almost all cases.

NSGAII-MIIP also selects fewer KPFs than benchmark algorithms in most cases. Specifically, on ADPN and LATEX, NSGAII-MIIP obtains significantly lower #KPFs values than benchmark algorithms except for MOPSO-LS. On PAPER, NSGAII-MIIP obtains a significantly lower #KPFs value than MOFS-BDE, NSGAII-IPM, IDMS-IPM, and MOEA/D. Although NSGAII-MIIP obtains a significantly higher #KPFs value than MOPSO-LS and NSGAII/SDR on PAPER, the difference of #KPFs values between NSGAII-MIIP and the two algorithms is not big. On SPIRA, NSGAII-MIIP obtains a significantly lower #KPFs value than GADMS-IPM, NSGAII-IPM, MOEA/D, and SPEA2.

Overall, the results in Table 2 show that NSGAII-MIIP generally obtains better or similar classification performance while selecting a smaller number of KPFs compared to the benchmark algorithms. This indicates that NSGAII-MIIP performs more effectively in selecting the KPFs.

6.2. Performance comparison on the synthetic datasets

In this section, we evaluate the performance of NSGAII-MIIP on the four synthetic datasets, i.e., ADPN-F, LATEX-F, PAPER-F, and SPIRA-F. The performance metric results of NSGAII-MIIP and benchmark algorithms on these datasets are shown in Table 3, where the notations are the same as that in Table 2.

In terms of classification performance, NSGAII-MIIP obtains higher GM, F1-score, and AUC values than the benchmark algorithms in most cases. On ADPN-F, NSGAII-MIIP obtains higher GM, F1-score,

Table 3: Comparison of KPF selection performance between NSGAII-MIIP and benchmark algorithms on the four synthetic datasets.

Dataset	Metric	NSGAII-MIIP	GADMS-IPM	MOPSO-LS	MOFS-BDE	NSGAII-IPM	IDMS-IPM	NSGAII/SDR	MOEA/D	SPEA2
ADPN-F	GM (%)	$\mathbf{77.95 \pm 11.38}$	72.71 ± 19.92	68.77 ± 29.23	70.13 ± 25.52	$66.36 \pm 30.07 \uparrow$	$68.12 \pm 25.95 \uparrow$	$64.41 \pm 30.54 \uparrow\uparrow$	70.66 ± 21.86	70.25 ± 27.02
	F1-score (%)	$\mathbf{70.33 \pm 12.86}$	$63.13 \pm 20.58 \uparrow$	61.11 ± 27.40	63.89 ± 25.46	$59.44 \pm 28.55 \uparrow$	$59.59 \pm 24.97 \uparrow\uparrow$	$56.63 \pm 27.84 \uparrow\uparrow$	$61.89 \pm 21.02 \uparrow$	62.68 ± 25.99
	AUC (%)	$\mathbf{85.67 \pm 16.84}$	82.56 ± 23.48	81.67 ± 23.02	82.19 ± 23.72	79.42 ± 25.48	83.31 ± 16.53	$77.22 \pm 29.57 \uparrow\uparrow$	78.94 ± 26.74	81.50 ± 26.07
	#KPFs	$\mathbf{3.0 \pm 0.7}$	$10.6 \pm 3.2 \uparrow\uparrow$	3.3 ± 1.4	$7.4 \pm 1.7 \uparrow\uparrow$	$16.2 \pm 2.4 \uparrow\uparrow$	$53.6 \pm 5.4 \uparrow\uparrow$	$5.2 \pm 2.5 \uparrow\uparrow$	$8.3 \pm 2.3 \uparrow\uparrow$	$17.9 \pm 2.6 \uparrow\uparrow$
LATEX-F	GM (%)	76.67 ± 10.93	$\mathbf{78.30 \pm 7.89}$	$72.97 \pm 11.27 \uparrow\uparrow$	$71.88 \pm 13.28 \uparrow\uparrow$	$71.25 \pm 10.20 \uparrow\uparrow$	$70.70 \pm 13.12 \uparrow\uparrow$	76.30 ± 10.76	75.05 ± 9.44	77.90 ± 10.89
	F1-score (%)	67.64 ± 13.24	$\mathbf{69.21 \pm 9.59}$	$63.33 \pm 13.66 \uparrow\uparrow$	$62.96 \pm 16.70 \uparrow$	$61.81 \pm 12.96 \uparrow\uparrow$	$61.19 \pm 15.53 \uparrow\uparrow$	66.79 ± 13.45	65.58 ± 12.01	68.69 ± 13.52
	AUC (%)	$\mathbf{86.29 \pm 5.52}$	84.32 ± 7.31	$82.33 \pm 8.30 \uparrow\uparrow$	83.81 ± 10.92	$81.95 \pm 7.15 \uparrow\uparrow$	$82.80 \pm 7.58 \uparrow\uparrow$	84.78 ± 7.82	84.35 ± 6.05	$84.56 \pm 6.84 \uparrow$
	#KPFs	8.0 ± 4.1	$19.9 \pm 3.1 \uparrow\uparrow$	$\mathbf{7.2 \pm 3.6}$	$17.6 \pm 2.4 \uparrow\uparrow$	$30.0 \pm 3.7 \uparrow\uparrow$	$67.2 \pm 3.6 \uparrow\uparrow$	$20.5 \pm 5.3 \uparrow\uparrow$	$13.7 \pm 3.8 \uparrow\uparrow$	$29.6 \pm 5.6 \uparrow\uparrow$
PAPER-F	GM (%)	$\mathbf{88.65 \pm 7.70}$	87.32 ± 7.96	87.47 ± 9.87	83.86 ± 11.00	$78.28 \pm 19.55 \uparrow\uparrow$	$82.34 \pm 11.26 \uparrow\uparrow$	86.28 ± 8.24	$83.50 \pm 11.84 \uparrow$	86.03 ± 9.94
	F1-score (%)	57.67 ± 15.18	55.28 ± 13.59	55.83 ± 13.34	57.43 ± 10.74	56.45 ± 18.64	$\mathbf{59.38 \pm 16.42}$	52.98 ± 11.13	51.75 ± 13.49	58.96 ± 13.63
	AUC (%)	92.38 ± 5.44	91.51 ± 5.63	91.24 ± 8.11	91.82 ± 5.40	91.73 ± 6.08	$\mathbf{92.50 \pm 5.77}$	92.33 ± 5.82	91.21 ± 6.88	91.95 ± 6.65
	#KPFs	3.5 ± 0.8	3.5 ± 1.2	3.7 ± 0.9	$5.3 \pm 1.5 \uparrow\uparrow$	$8.3 \pm 1.9 \uparrow\uparrow$	$19.5 \pm 4.4 \uparrow\uparrow$	$\mathbf{2.9 \pm 1.3} \downarrow$	$4.9 \pm 1.3 \uparrow\uparrow$	$6.5 \pm 2.1 \uparrow\uparrow$
SPIRA-F	GM (%)	$\mathbf{78.71 \pm 7.42}$	$74.62 \pm 7.73 \uparrow\uparrow$	$67.45 \pm 15.06 \uparrow\uparrow$	$74.64 \pm 10.83 \uparrow\uparrow$	$71.21 \pm 11.75 \uparrow\uparrow$	$69.53 \pm 14.79 \uparrow\uparrow$	$72.93 \pm 10.85 \uparrow\uparrow$	$71.33 \pm 13.26 \uparrow\uparrow$	$73.50 \pm 12.27 \uparrow\uparrow$
	F1-score (%)	$\mathbf{73.64 \pm 8.76}$	$67.92 \pm 9.66 \uparrow\uparrow$	$59.14 \pm 18.17 \uparrow\uparrow$	$68.21 \pm 13.04 \uparrow\uparrow$	$63.82 \pm 14.34 \uparrow\uparrow$	$61.98 \pm 17.90 \uparrow\uparrow$	$66.24 \pm 12.72 \uparrow\uparrow$	$64.18 \pm 16.33 \uparrow\uparrow$	$66.44 \pm 15.18 \uparrow\uparrow$
	AUC (%)	$\mathbf{85.30 \pm 10.57}$	$81.11 \pm 8.93 \uparrow\uparrow$	$79.40 \pm 11.51 \uparrow\uparrow$	$81.69 \pm 11.15 \uparrow$	$81.21 \pm 10.29 \uparrow\uparrow$	82.59 ± 11.70	84.50 ± 7.98	82.88 ± 12.07	82.80 ± 9.14
	#KPFs	$\mathbf{3.9 \pm 1.2}$	$10.5 \pm 2.9 \uparrow\uparrow$	$5.3 \pm 1.9 \uparrow\uparrow$	$8.8 \pm 2.1 \uparrow\uparrow$	$17.4 \pm 2.7 \uparrow\uparrow$	$50.0 \pm 6.4 \uparrow\uparrow$	$8.8 \pm 3.8 \uparrow\uparrow$	$10.1 \pm 3.3 \uparrow\uparrow$	$18.7 \pm 3.9 \uparrow\uparrow$

and AUC values than all benchmark algorithms. The significance test results indicate that the GM, F1-score, or AUC value of NSGAII-MIIP is significantly higher than that of the benchmark algorithms except for MOPSO-LS, MOFS-BDE, and SPEA2. On LATEX-F, NSGAII-MIIP obtains a significantly higher GM, F1-score, or AUC value than benchmark algorithms except for GADMS-IPM, NSGAII/SDR, and MOEA/D. NSGAII-MIIP obtains slightly lower GM and F1-score values than GADMS-IPM and SPEA2, but the significance test results are not significant. On PAPER-F, NSGAII-MIIP obtains the highest GM value, significantly higher than NSGAII-IPM, IDMS-IPM, and MOEA/D. Meanwhile, the F1-score and AUC values of NSGAII-MIIP are slightly lower than IDMS-IPM. On SPIRA-F, NSGAII-MIIP obtains significantly higher GM, F1-score, and AUC values than benchmark algorithms in almost all cases.

In terms of #KPFs, NSGAII-MIIP obtains a significantly lower #KPFs value than the benchmark algorithms on the four synthetic datasets in most cases. In a few cases, NSGAII-MIIP obtains similar or slightly higher #KPFs values. Specifically, compared with GADMS-IPM, NSGAII-MIIP obtains the same mean #KPFs value on PAPER-F. Compared with MOPSO-LS, NSGAII-MIIP obtains a slightly higher #KPFs value on LATEX-F and similar #KPFs values on ADPN-F and PAPER-F. Compared with NSGASII/SDR, NSGAII-MIIP obtains a significantly higher #KPFs value on PAPER-F. Although NSGAII-MIIP does not significantly reduce #KPFs compared with the above benchmark algorithms in these cases, it obtains better classification performance. This indicates that NSGAII-MIIP is more effective in selecting informative PFs than the benchmark algorithms.

Moreover, the performance results of these algorithms in Table 3 deteriorate more or less compared to those in Table 2. Generally, these algorithms select more KPFs and obtain worse classification performance on the four synthetic datasets, denoting that the added noisy features have affected the KPF selection performance. However, compared with the benchmark algorithms, NSGAII-MIIP is less influenced by the noisy features. The results of #KPFs on ADPN-F, PAPER-F, and SPIRA-F are very close to that of ADPN, PAPER, and SPIRA. The GM, F1-score, and AUC results on ADPN-F, LATEX-F, and PAPER-F are similar to those of ADPN, LATEX, and PAPER. In comparison, the #KPFs values of the benchmark algorithms except for MOPSO-LS are substantially increased and the classification performance results of

the benchmark algorithms are generally much decreased on the synthetic datasets. These results show that NSGAII-MIIP outperforms the benchmark algorithms on these high-dimensional synthetic CMP datasets.

It is also worth noting that the improvement degrees of NSGAII-MIIP compared to benchmark algorithms on different datasets are different. For example, NGAII-MIIP obtains much better results on SPIRA-F and SPIRA, while obtaining slightly better or similar results on PAPER and PAPER-F. This can be explained with two possible reasons. First, the KPF selection task on PAPER and PAPER-F is much easier than that on SPIRA and SPIRA-F as PAPER has substantially fewer PFs than SPIRA. Therefore, the computational resources are more adequate for benchmark algorithms to solve the KPF selection model on PAPER. Second, different performances on different datasets may be because of different intrinsic properties of different datasets. One objective of the feature importance measure in the improvement phase of NSGAII-MIIP is to address the feature redundancy problem. If the redundancy level among features in a dataset is less significant, the performance of the improvement phase would be less significant as well.

6.3. Computation time

Fig. 1 shows the average computation time over the 30 runs consumed by each algorithm on the eight datasets. NSGAII-MIIP and MOPSO-LS are the two most time-efficient algorithms since they generally consume less time than the remaining algorithms on these datasets. On the datasets except for ADPN and ADPN-F, we can find wide gaps between the time bars of NSGAII-MIIP/MOPSO-LS and those of other algorithms. The reason is that both NSGAII-MIIP and MOPSO-LS adopt an improvement step to accelerate feature reduction during the iteration process. Since the time of function evaluations in these wrapper-based FS methods is determined by the size of an evaluated feature subset, accelerating feature reduction reduces the evaluation time. It is worth noting that, although MOFS-BDE also adopts an improvement step to improve the search performance, this step updates only one solution each time and executes every five iterations. This limits MOFS-BDE's ability to accelerate feature reduction on these datasets. Compared with MOPSO-LS, NSGAII-MIIP consumes less time on five datasets and more time on three datasets.

7. Further analysis on the performance of NSGAII-MIIP

NSGAII-MIIP and benchmark FS methods apply a two-phase strategy to select the final KPFs. In the first phase, an MOO algorithm is applied to obtain a non-dominated set. In the second phase, the IPM is used to select the final solution (i.e., KPF set) from the non-dominated set. The search results of the MOO algorithms in the first phase have a significant impact on the final KPF selection performance. Therefore, in this section, we further evaluate the performance of NSGAII-MIIP in the first phase. First, we evaluate the overall KPF selection performance (on the test set) of non-dominated sets obtained by NSGAII-MIIP and benchmark algorithms. Then, we evaluate the search performance and convergence properties of NSGAII-MIIP and benchmark algorithms based on the two objective function values.

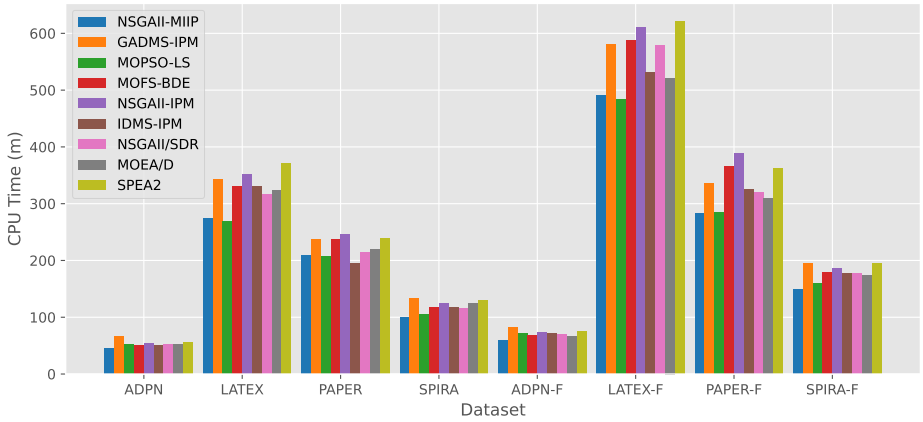

Figure 1: Computation time of the FS methods.

7.1. Performance measures

Three performance metrics, hypervolume (HV) (Yuan et al., 2016), inverted generational distance (IGD), and set coverage (Zhang & Li, 2007) are used to evaluate the performance of non-dominated solutions (i.e., the search results of the MOO algorithms) obtained by the FS methods obtained in the first phase. HV is a Pareto-compliant metric evaluating the goodness of a non-dominated set by estimating the hypervolume it dominates in the objective space. The reference point to calculate HV is set as $1.1\mathbf{z}^{nad}$ as suggested in Yuan et al. (2016), where $\mathbf{z}^{nad}$ is defined as the point with the worst objective function values of all non-dominated solutions obtained by the compared algorithms. IGD measures the average distance value between each Pareto-optimal point and the closest point in a non-dominated set to evaluate the goodness of the non-dominated set. SC measures the percentage of solutions in a non-dominated set $\mathcal{Y}$ covered by (dominated by or equal to) one or more solutions in another non-dominated set $\mathcal{X}$. In the experiments, we set $\mathcal{Y}$ as the Pareto-optimal set to evaluate the effectiveness of $\mathcal{X}$. Both HV and SC are the higher the better, and IGD is the lower the better.

We can obtain 30 values of HV and SC based on the search results from the 30 experimental runs for each algorithm. It is worth noting that, we obtain the non-dominated set from all the solutions found by the compared algorithms for each experimental run. The obtained non-dominated set is set as the estimated Pareto-optimal set for IGD and SC calculations. Moreover, the objective function values are normalized with the max-min normalization method before calculating performance metric values. The maximum and minimum of each objective function used in the normalization method are set as the maximum and minimum objective function values of all non-dominated solutions obtained by the compared algorithms.

7.2. Comparison of overall KPF selection performance of non-dominated sets found by the multi-objective FS methods

In this section, the KPF selection performances of the non-dominated solutions found by NSGAII-MIIP and benchmark algorithms are compared. The above-mentioned performance metrics to evaluate MOO algorithms are based on the objective functions in the KPF selection model. The two objective functions in the proposed KPF selection model are the "1-GM" and "#KPFs", which are evaluated on the training set. However, since we need to properly evaluate the KPF selection performance, the classification performance on the test set should be used. Therefore, we further define the objective functions on the test set to evaluate the overall KPF selection performance. Specifically, a pair of the objective functions is defined as $\mathbf{F}_{obj} = (f_1 = 1 - classification\ performance, f_2 = \#KPFs)^T$. The AUC, GM, and F1-score are used as the classification performance measures respectively, which yield three pairs of the objective functions. Based on the three measures, we then obtain the HV, IGD, and SC results by comparing between NSGAII-MIIP and benchmark algorithms. Due to the page limit, we show the results on the AUC measure in Table 4 and show the results on GM and the F1-score in the supplementary material of this paper.

Table 4 compares the overall KPF selection performance of the obtained non-dominated sets between NSGAII-MIIP and benchmark algorithms. The mean and std. (mean ± std.) results of HV, IGD, and SC over the 30 runs are shown, and the signs ⇈ (↑) and ⇊ (↓) show the results of the Wilcoxon signed-rank test, which is the same as that in the previous section. Generally, NSGAII-MIIP obtains the best results on HV, IGD, and SC in 19 of 24 cases. Specifically, NSGAII-MIIP obtains significantly better HV, IGD, and SC values than the benchmark algorithms except for MOPSO-LS and MOFS-BDE in almost all cases according to the significance test results. Compared to MOPSO-LS, NSGAII-MIIP obtains better results on HV, IGD, and SC in 20 of 24 cases, and the results are significant in 9 cases. Compared to MOFS-BDE, NSGAII-MIIP obtains better results on HV, IGD, and SC in 20 of 24 cases, and the results are significant in 17 cases. NSGAII-MIIP only obtains a significantly worse IGD value than MOFS-BDE on ADPN. To sum up, the overall KPF selection performance of the non-dominated solutions found by NSGAII-MIIP is much better than these benchmark MOO algorithms.

7.3. Comparison of search performance

In this section, the search performance of NSGAII-MIIP is compared to benchmark MOO algorithms. Objective function values of non-dominated solutions of these algorithms are used for the comparison. MOFS-BDE and NSGAII-IPM adopted a different KPF selection model where accuracy is used instead of GM. Therefore, we further collect experimental results by applying them to the same KPF selection model as NSGAII-MIIP to facilitate the comparison. The HV, IGD, and SC metrics are used as the performance metrics. The Wilcoxon signed-rank test is used to verify if the results between NSGAII-MIIP and benchmark algorithms are significantly different.

Table 4: Comparison of overall KPF selection performance of obtained non-dominated sets between NSGAII-MIIP and benchmark algorithms using the AUC measure.

Metric	Dataset	NSGAII-MIIP	GADMS-IPM	MOPSO-LS	MOFS-BDE	NSGAII-IPM	IDMS-IPM	NSGA-II/SDR	MOEA/D	SPEA2
HV	ADPN	**1.196 ± 0.010**	1.177 ± 0.027 ††	1.184 ± 0.049 ††	1.193 ± 0.015	0.999 ± 0.192 ††	0.216 ± 0.126 ††	1.164 ± 0.048 ††	1.129 ± 0.051 ††	1.043 ± 0.067 ††
	LATEX	1.108 ± 0.048	1.028 ± 0.096 ††	1.106 ± 0.065	**1.109 ± 0.061**	0.894 ± 0.106 ††	0.307 ± 0.114 ††	1.054 ± 0.125 ††	1.024 ± 0.094 ††	0.917 ± 0.077 ††
	PAPER	**1.175 ± 0.041**	1.145 ± 0.074 ††	1.152 ± 0.076 ††	0.900 ± 0.103 ††	0.934 ± 0.144 ††	0.832 ± 0.245 ††	1.127 ± 0.086 ††	1.128 ± 0.104 ††	1.113 ± 0.076 ††
	SPIRA	**1.164 ± 0.053**	1.126 ± 0.077 ††	1.148 ± 0.063	1.115 ± 0.112 ††	0.998 ± 0.089 ††	0.255 ± 0.117 ††	1.072 ± 0.125 ††	1.060 ± 0.091 ††	1.031 ± 0.081 ††
	ADPN-F	**1.198 ± 0.009**	1.065 ± 0.057 ††	1.196 ± 0.014	1.137 ± 0.025 ††	0.934 ± 0.056 ††	0.195 ± 0.069 ††	1.120 ± 0.107 ††	1.136 ± 0.025 ††	0.893 ± 0.165 ††
	LATEX-F	1.110 ± 0.050	0.942 ± 0.065 ††	**1.110 ± 0.059**	0.999 ± 0.048 ††	0.806 ± 0.052 ††	0.283 ± 0.068 ††	0.931 ± 0.054 ††	1.051 ± 0.057 ††	0.821 ± 0.074 ††
	PAPER-F	**1.097 ± 0.056**	1.054 ± 0.088 ††	1.066 ± 0.097 ††	1.096 ± 0.081	0.877 ± 0.107 ††	0.390 ± 0.160 ††	1.062 ± 0.111 ↑	1.011 ± 0.142 ††	0.938 ± 0.116 ††
	SPIRA-F	**1.176 ± 0.044**	1.015 ± 0.077 ††	1.146 ± 0.082 ↑	1.079 ± 0.084 ††	0.862 ± 0.106 ††	0.231 ± 0.109 ††	1.050 ± 0.096 ††	1.029 ± 0.133 ††	0.873 ± 0.085 ††
IGD	ADPN	0.040 ± 0.036	0.068 ± 0.058 ↑	0.056 ± 0.064	**0.024 ± 0.024** ↓	0.206 ± 0.151 ††	0.906 ± 0.112 ††	0.067 ± 0.066 ↑	0.100 ± 0.071 ††	0.181 ± 0.075 ††
	LATEX	**0.091 ± 0.038**	0.171 ± 0.066 ††	0.092 ± 0.050	0.140 ± 0.063 ††	0.274 ± 0.049 ††	0.762 ± 0.095 ††	0.142 ± 0.091 ††	0.157 ± 0.054 ††	0.250 ± 0.061 ††
	PAPER	**0.051 ± 0.057**	0.069 ± 0.073 ††	0.075 ± 0.080 ††	0.274 ± 0.108 ††	0.273 ± 0.144 ††	0.312 ± 0.212 ††	0.088 ± 0.073 ††	0.090 ± 0.110 ††	0.111 ± 0.077 ††
	SPIRA	**0.063 ± 0.041**	0.109 ± 0.083 ††	0.072 ± 0.058	0.116 ± 0.092 ††	0.184 ± 0.065 ††	0.835 ± 0.124 ††	0.132 ± 0.080 ††	0.159 ± 0.096 ††	0.170 ± 0.086 ††
	ADPN-F	**0.026 ± 0.051**	0.193 ± 0.096 ††	0.035 ± 0.052	0.138 ± 0.081 ††	0.297 ± 0.086 ††	0.942 ± 0.072 ††	0.134 ± 0.089 ††	0.125 ± 0.083 ††	0.332 ± 0.140 ††
	LATEX-F	**0.072 ± 0.031**	0.185 ± 0.045 ††	0.077 ± 0.036	0.148 ± 0.035 ††	0.289 ± 0.044 ††	0.765 ± 0.074 ††	0.192 ± 0.049 ††	0.106 ± 0.032 ††	0.283 ± 0.063 ††
	PAPER-F	**0.096 ± 0.037**	0.126 ± 0.067 ††	0.129 ± 0.088 ††	0.096 ± 0.043	0.225 ± 0.073 ††	0.632 ± 0.170 ††	0.115 ± 0.070	0.166 ± 0.110 ††	0.206 ± 0.094 ††
	SPIRA-F	**0.056 ± 0.041**	0.179 ± 0.071 ††	0.094 ± 0.099 ↑	0.146 ± 0.070 ††	0.299 ± 0.075 ††	0.883 ± 0.097 ††	0.149 ± 0.080 ††	0.192 ± 0.123 ††	0.293 ± 0.063 ††
SC	ADPN	0.300 ± 0.447	0.167 ± 0.303	0.317 ± 0.425	**0.450 ± 0.497**	0.000 ± 0.000 ††	0.000 ± 0.000 ††	0.250 ± 0.410	0.033 ± 0.183 ††	0.033 ± 0.183 ††
	LATEX	0.092 ± 0.154	0.050 ± 0.132	0.149 ± 0.199	0.082 ± 0.140	0.008 ± 0.046 ††	0.006 ± 0.030 ††	**0.172 ± 0.244**	0.008 ± 0.046 ††	0.007 ± 0.037 ††
	PAPER	**0.700 ± 0.337**	0.417 ± 0.475 ††	0.550 ± 0.422 ††	0.000 ± 0.000 ††	0.067 ± 0.217 ††	0.033 ± 0.127 ††	0.417 ± 0.437 ††	0.483 ± 0.464 ††	0.217 ± 0.364 ††
	SPIRA	**0.317 ± 0.352**	0.217 ± 0.364	0.290 ± 0.384	0.123 ± 0.214 ††	0.007 ± 0.037 ††	0.000 ± 0.000 ††	0.097 ± 0.190 ††	0.090 ± 0.231 ††	0.037 ± 0.119 ††
	ADPN-F	**0.600 ± 0.443**	0.000 ± 0.000 ††	0.533 ± 0.414	0.000 ± 0.000 ††	0.000 ± 0.000 ††	0.000 ± 0.000 ††	0.100 ± 0.242 ††	0.017 ± 0.091 ††	0.000 ± 0.000 ††
	LATEX-F	**0.144 ± 0.214**	0.007 ± 0.037 ††	0.143 ± 0.195	0.040 ± 0.074 ††	0.000 ± 0.000 ††	0.000 ± 0.000 ††	0.010 ± 0.036 ††	0.012 ± 0.047 ††	0.005 ± 0.026 ††
	PAPER-F	**0.397 ± 0.121**	0.189 ± 0.213 ††	0.261 ± 0.234 ††	0.242 ± 0.246 ††	0.022 ± 0.085 ††	0.000 ± 0.000 ††	0.319 ± 0.215 ↑	0.136 ± 0.204 ††	0.028 ± 0.086 ††
	SPIRA-F	**0.300 ± 0.385**	0.000 ± 0.000 ††	0.300 ± 0.428	0.017 ± 0.091 ††	0.000 ± 0.000 ††	0.000 ± 0.000 ††	0.033 ± 0.127 ††	0.017 ± 0.091 ††	0.000 ± 0.000 ††

Table 5: Comparison of search performance between NSGAII-MIIP and benchmark algorithms.

Metric	Dataset	NSGAII-MIIP	GADMS-IPM	MOPSO-LS	MOFS-BDE	NSGAII-IPM	IDMS-IPM	NSGA-II/SDR	MOEA/D	SPEA2
HV	ADPN	**1.165 ± 0.024**	1.142 ± 0.037 ††	1.148 ± 0.027 ††	1.122 ± 0.043 ††	0.946 ± 0.071 ††	0.463 ± 0.127 ††	1.092 ± 0.039 ††	1.111 ± 0.040 ††	1.047 ± 0.063 ††
	LATEX	**1.155 ± 0.032**	1.100 ± 0.041 ††	1.107 ± 0.042 ††	1.078 ± 0.040 ††	0.920 ± 0.067 ††	0.382 ± 0.077 ††	1.066 ± 0.070 ††	1.060 ± 0.062 ††	0.942 ± 0.059 ††
	PAPER	**1.127 ± 0.030**	1.107 ± 0.041 ††	1.097 ± 0.046 ††	1.122 ± 0.038	1.078 ± 0.041 ††	0.932 ± 0.144 ††	1.095 ± 0.055 ††	1.088 ± 0.044 ††	1.088 ± 0.053 ††
	SPIRA	**1.144 ± 0.025**	1.126 ± 0.040 ††	1.112 ± 0.052 ††	1.116 ± 0.036 ††	0.947 ± 0.067 ††	0.420 ± 0.067 ††	1.073 ± 0.070 ††	1.055 ± 0.060 ††	1.018 ± 0.067 ††
	ADPN-F	**1.158 ± 0.026**	1.056 ± 0.041 ††	1.123 ± 0.045 ††	1.052 ± 0.031 ††	0.906 ± 0.057 ††	0.357 ± 0.072 ††	1.068 ± 0.062 ††	1.099 ± 0.029 ††	0.948 ± 0.033 ††
	LATEX-F	**1.142 ± 0.037**	0.964 ± 0.057 ††	1.089 ± 0.041 ††	0.954 ± 0.036 ††	0.807 ± 0.048 ††	0.310 ± 0.070 ††	0.932 ± 0.047 ††	1.069 ± 0.044 ††	0.842 ± 0.058 ††
	PAPER-F	**1.141 ± 0.024**	1.121 ± 0.034 ††	1.111 ± 0.029 ††	1.126 ± 0.025 ††	0.977 ± 0.071 ††	0.553 ± 0.141 ††	1.098 ± 0.039 ††	1.082 ± 0.044 ††	1.048 ± 0.047 ††
	SPIRA-F	**1.129 ± 0.035**	1.018 ± 0.058 ††	1.047 ± 0.075 ††	1.008 ± 0.045 ††	0.830 ± 0.046 ††	0.322 ± 0.111 ††	0.993 ± 0.074 ††	1.013 ± 0.076 ††	0.864 ± 0.071 ††
IGD	ADPN	**0.025 ± 0.021**	0.061 ± 0.030 ††	0.050 ± 0.024 ††	0.074 ± 0.034 ††	0.186 ± 0.054 ††	0.588 ± 0.112 ††	0.087 ± 0.041 ††	0.084 ± 0.026 ††	0.118 ± 0.045 ††
	LATEX	**0.027 ± 0.021**	0.064 ± 0.024 ††	0.061 ± 0.026 ††	0.073 ± 0.019 ††	0.161 ± 0.042 ††	0.602 ± 0.063 ††	0.080 ± 0.037 ††	0.096 ± 0.033 ††	0.150 ± 0.036 ††
	PAPER	0.036 ± 0.021	0.050 ± 0.021 ††	0.069 ± 0.032 ††	**0.032 ± 0.012**	0.066 ± 0.023 ††	0.132 ± 0.068 ††	0.097 ± 0.041 ††	0.060 ± 0.023 ††	0.065 ± 0.020 ††
	SPIRA	**0.024 ± 0.014**	0.042 ± 0.024 ††	0.048 ± 0.028 ††	0.048 ± 0.021 ††	0.141 ± 0.043 ††	0.548 ± 0.078 ††	0.078 ± 0.039 ††	0.091 ± 0.032 ††	0.106 ± 0.042 ††
	ADPN-F	**0.034 ± 0.021**	0.124 ± 0.029 ††	0.068 ± 0.029 ††	0.129 ± 0.026 ††	0.230 ± 0.043 ††	0.686 ± 0.061 ††	0.105 ± 0.036 ††	0.092 ± 0.028 ††	0.195 ± 0.030 ††
	LATEX-F	**0.043 ± 0.028**	0.142 ± 0.045 ††	0.079 ± 0.026 ††	0.154 ± 0.037 ††	0.258 ± 0.046 ††	0.694 ± 0.067 ††	0.160 ± 0.039 ††	0.089 ± 0.029 ††	0.232 ± 0.054 ††
	PAPER-F	0.046 ± 0.047	0.054 ± 0.028 ↑	0.061 ± 0.024 ††	**0.044 ± 0.011**	0.111 ± 0.032 ††	0.394 ± 0.118 ††	0.083 ± 0.040 ††	0.083 ± 0.030 ††	0.089 ± 0.028 ††
	SPIRA-F	**0.032 ± 0.016**	0.107 ± 0.029 ††	0.085 ± 0.035 ††	0.119 ± 0.026 ††	0.226 ± 0.038 ††	0.667 ± 0.114 ††	0.115 ± 0.041 ††	0.118 ± 0.044 ††	0.200 ± 0.048 ††
SC	ADPN	**0.404 ± 0.243**	0.171 ± 0.191 ††	0.217 ± 0.155 ††	0.042 ± 0.095 ††	0.000 ± 0.000 ††	0.000 ± 0.000 ††	0.181 ± 0.171 ††	0.036 ± 0.076 ††	0.013 ± 0.051 ††
	LATEX	**0.420 ± 0.187**	0.053 ± 0.119 ††	0.134 ± 0.092 ††	0.016 ± 0.044 ††	0.007 ± 0.039 ††	0.000 ± 0.000 ††	0.024 ± 0.065 ††	0.011 ± 0.037 ††	0.003 ± 0.014 ††
	PAPER	**0.371 ± 0.181**	0.228 ± 0.163 ††	0.194 ± 0.101 ††	0.305 ± 0.149 ↑	0.093 ± 0.100 ††	0.022 ± 0.080 ††	0.226 ± 0.123 ††	0.120 ± 0.100 ††	0.109 ± 0.109 ††
	SPIRA	**0.411 ± 0.152**	0.160 ± 0.160 ††	0.232 ± 0.144 ††	0.077 ± 0.116 ††	0.000 ± 0.000 ††	0.000 ± 0.000 ††	0.136 ± 0.150 ††	0.033 ± 0.092 ††	0.004 ± 0.023 ††
	ADPN-F	**0.333 ± 0.198**	0.013 ± 0.050 ††	0.155 ± 0.180 ††	0.006 ± 0.030 ††	0.000 ± 0.000 ††	0.000 ± 0.000 ††	0.054 ± 0.155 ††	0.005 ± 0.026 ††	0.004 ± 0.020 ††
	LATEX-F	**0.274 ± 0.312**	0.000 ± 0.000 ††	0.041 ± 0.093 ††	0.015 ± 0.042 ††	0.000 ± 0.000 ††	0.000 ± 0.000 ††	0.000 ± 0.000 ††	0.015 ± 0.066 ††	0.000 ± 0.000 ††
	PAPER-F	**0.264 ± 0.189**	0.171 ± 0.198 ††	0.133 ± 0.158 ††	0.120 ± 0.151 ††	0.004 ± 0.020 ††	0.000 ± 0.000 ††	0.155 ± 0.109 ††	0.050 ± 0.075 ††	0.031 ± 0.068 ††
	SPIRA-F	**0.393 ± 0.217**	0.013 ± 0.042 ††	0.049 ± 0.064 ††	0.028 ± 0.058 ††	0.000 ± 0.000 ††	0.000 ± 0.000 ††	0.026 ± 0.097 ††	0.005 ± 0.028 ††	0.000 ± 0.000 ††

Table 5 shows the HV, IGD, and SC results obtained by NSGAII-MIIP and the benchmark algorithms. The significance test results indicate that NSGAII-MIIP obtains significantly better HV, IGD, and SC values than benchmark algorithms in almost all cases. The only exception is that NSGAII-MIIP obtains an HV value similar to MOFS-BDE on PAPER, and obtains slightly higher IGD values than MOFS-BDE on PAPER and PAPER-F. A possible reason for the exception is that the PAPER dataset has relatively fewer PFs than other datasets. Therefore, the search space of PAPER is much smaller than other datasets and thus requires fewer computational resources. This allows MOFS-BDE to obtain good search results with current computational resources. Overall, the results in Table 5 demonstrate that NSGAII-MIIP has superior search results than these benchmark MOO algorithms.

To reveal the convergence behavior of NSGAII-MIIP, we further draw the convergence curves of NSGAII-MIIP and benchmark MOO algorithms. The non-dominated solutions at each iteration are recorded to evaluate the convergence performance of the algorithms. In this paper, the convergence distance (CD)

metric (Li et al., 2019) is adopted to evaluate the convergence performance. The CD value estimates the similarity between a non-dominated set and the Pareto-optimal set. It is calculated as the average of the IGD and the generational distance (GD). As 30 runs are conducted, the average CD values over the 30 runs are used to draw the convergence curve of each algorithm.

Fig. 2 shows the convergence curves of the algorithms. NSGAII-MIIP shows similar converging behaviors on the eight datasets. It obtains lower convergence curves than benchmark algorithms except for MOPSO-LS in the whole evolutionary process. NSGAII-MIIP obtains higher convergence curves than MOPSO-LS in the early evolutionary phase and gradually obtains lower curves than MOPSO-LS during the iteration process. MOPSO-LS starts with a lower CD value than NSGAII-MIIP. This is because MOPSO-LS uses a different threshold value (i.e., 0.6) to decode the real-coded particles, where the expected number of selected features in the initialized particles of MOPSO-LS is around $40\%N$ instead of $50\%N$ in other algorithms. The convergence curves of NSGAII-MIIP and MOEA/D are close in the early evolutionary phase. However, the convergence curves of MOEA/D gradually get higher than NSGAII-MIIP, showing that it suffers from the premature convergence problem. It is worth noting that the curves of NSGA-II/SDR start with a higher point than other algorithms. This is because NSGA-II/SDR uses the strengthened dominance relation instead of the Pareto dominance concept in other algorithms to rank solutions, which makes the non-dominated solutions of NSGA-II/SDR from the initial population different from other algorithms and thus yields a different CD value. The convergence curves of IDMS-IPM are significantly higher than other algorithms. The reason is that the direct search mechanism of IDMS-IPM makes it converge slowly.

7.4. Discussion

The experimental results presented in Sections 6 and 7 demonstrate that NSGAII-MIIP has competitive KPF selection performance. First, NSGAII-MIIP (with IPM) can select a smaller number of KPFs while maintaining similar or higher classification performance on both original and synthetic CMP datasets compared to the benchmark algorithms. Second, the candidate KPF sets (non-dominated solutions) found by NSGAII-MIIP in the first phase generally have better quality prediction performance than benchmark algorithms, indicating the MOO algorithm NSGAII-MIIP (without IPM) has better KPF selection performance. Third, NSGAII-MIIP has better search performance than benchmark MOO algorithms as it can quickly increase the overall fitness values of the non-dominated solutions during the iteration process. Finally, NSGAII-MIIP is efficient, requiring less computation time than the benchmark algorithms. The effectiveness and efficiency of NSGAII-MIIP can be explained by the following reasons:

1. The KPF selection model used in NSGAII-MIIP is effective for unbalanced CMP data. It addresses the data imbalance problem by using GM to evaluate the quality prediction performance of different PF subsets. A high GM value requires good performance in classifying both positive and negative instances. In comparison, accuracy is primarily impacted by negative (majority class) instances of

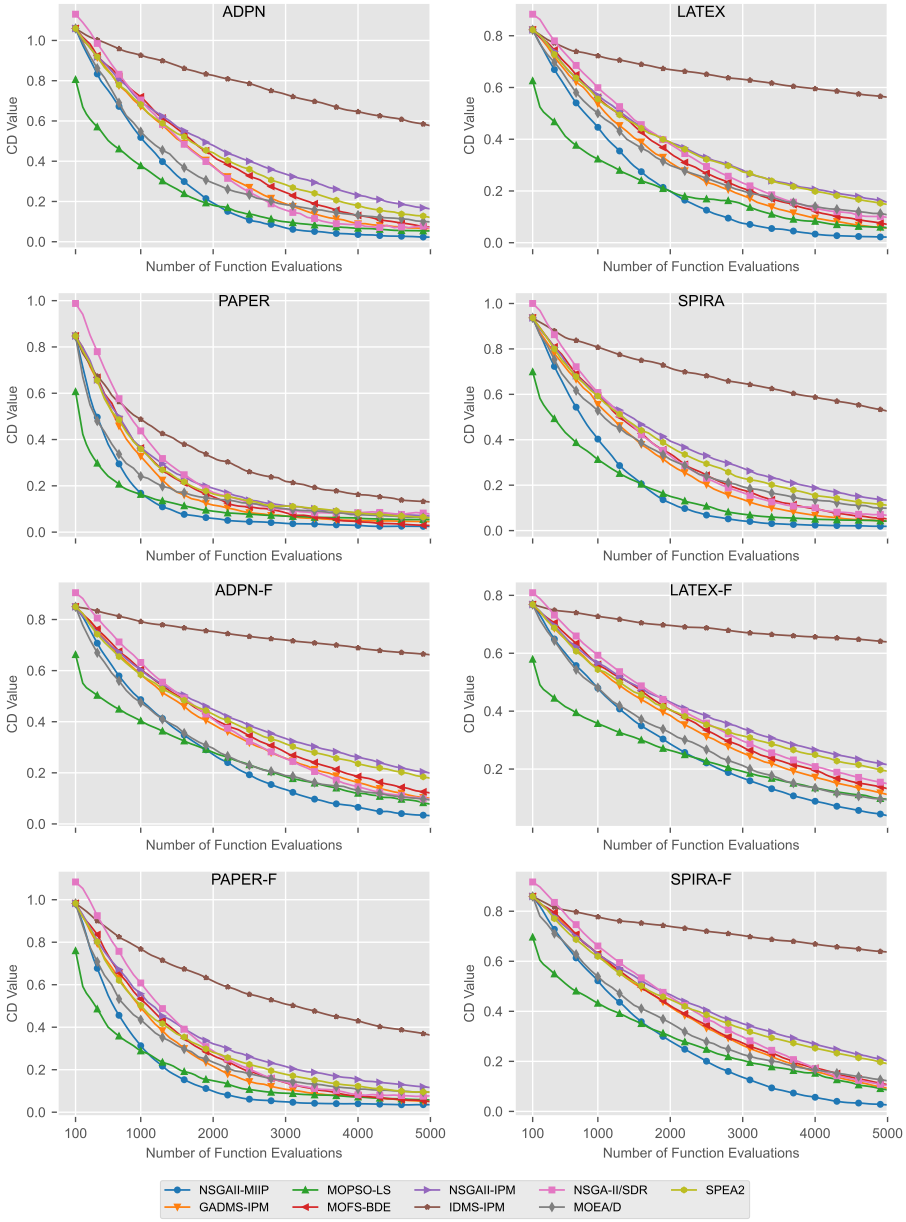

Figure 2: Convergence curves of NSGAII-MIIP and benchmark algorithms.

unbalanced data. The advantage of GM over accuracy in forming the KPF selection model is demonstrated in Table 2. On LATEX, PAPER, and SPIRA datasets, NSGAII-IPM that applies accuracy in the KPF selection model obtains substantially lower GM, F1-score, and AUC values than NSGAII-MIIP and other algorithms that apply GM in the model.

2. The improvement phase improves the KPF selection performance of NSGAII-MIIP. The proposed MI-guided improvement strategy uses a feature importance measure considering both feature relevance

and redundancy to guide the improvement operations. Therefore, NSGAII-MIIP can effectively search for candidate KPF sets by improving their relevance degree to product quality and reducing the interior feature redundancy. Such a strategy improves both the search performance of NSGAII-MIIP and the quality prediction performance (on the test set) of the obtained non-dominated solutions. In Section 6 of the supplementary material, we have conducted an ablation study to justify the MI-guided improvement strategy. Two variants of NSGAII-MIIP called NSGAII-MIIP-N and NSGAII-MIIP-R are established. NSGAII-MIIP-N does not use an improvement phase to purify the non-dominated solutions during the iteration process. NSGAII-MIIP-R replaces the MI-guided improvement strategy in NSGAII-MIIP with the random search-based improvement strategy in MOPSO-LS (He et al., 2022). The experimental results reveal that NSGAII-MIIP obtains better search performance and KPF/feature selection results than the two variants in most cases.

3. The improvement phase improves the time efficiency of NSGAII-MIIP. The results in Section 6.3 denote that NSGAII-MIIP is more time-efficient than benchmark algorithms, including the NSGA-II variants NSGAII-IPM and NSGA-II/SDR. The first reason for the efficiency of NSGAII-MIIP is that the improvement-phase-embedded ranking approach is efficient as we explained in Section 4.5. The second reason is that the MI-guided improvement strategy improves the convergence speed of NSGAII-MIIP and makes it quickly reduces the number of features selected by solutions in the population. Therefore, the time of function evaluations during the iteration process can be saved.

8. Conclusions

In this paper, we propose a novel MOEA called NSGAII-MIIP to select KPFs in CMPs. NSGAII-MIIP aims to optimize a bi-objective KPF selection (FS) model that maximizes the GM metric and minimizes the number of selected features. To improve the FS performance, NSGAII-MIIP uses an improvement phase to purify the non-dominated solutions during the iteration process. In this improvement phase, an MI-guided improvement strategy that adopts an importance measure is used. This measure considers both feature relevance and redundancy to evaluate the feature importance so that the most promising/unpromising features in terms of a non-dominated solution can be selected to take the improvement operations. Moreover, the improvement phase is seamlessly embedded in the solution ranking process of NSGAII-MIIP. Thus, NSGAII-MIIP can efficiently rank the solutions from both the improvement phase and genetic operations without consuming additional ranking time. We have verified the performance of NSGAII-MIIP on four CMP datasets and four synthetic datasets. NSGAII-MIIP demonstrates superior KPF selection performance than benchmark multi-objective FS methods, particularly on synthetic datasets with a higher dimensionality. Moreover, NSGAII-MIIP exhibits better search performance than several typical MOO algorithms, showing the competitiveness of NSGAII-MIIP on high-dimensional KPF selection problems.

In practice, the quality of a product can be measured by several variables. Building a KPF selection method for CMPs with multiple response variables is worth studying. Moreover, the difficulty of a KPF selection problem increases dramatically with the increase of data dimensionality. It is worth further investigating effective surrogate strategy or cooperative coevolutionary strategy to improve the FS performance of an MOEA for high-dimensional CMP data. First, a surrogate fitness function can be built with filter measures or historical optimization data of an MOEA to reduce the fitness evaluation time of high-dimensional FS problems. Second, for high-dimensional CMP data, the original large feature space can be divided into several small feature sub-spaces, which substantially decreases the search space of the FS problem. Based on this, a cooperative coevolution strategy can be designed to apply several MOEA instances to solve FS problems in different feature sub-spaces. The solutions of different sub-spaces can then integrate into the solutions of the original feature space.

Conflict of interest: None

Acknowledgments

The authors would like to thank the editor and anonymous referees for the constructive comments and suggestions. This work was supported by the National Natural Science Foundation of China (NSFC) [grant numbers 72101182, 72231005, 72261147706]; the State Administration of Science, Technology and Industry for National Defense of China [grant number JSZL2021204B001]; and the Humanities and Social Sciences Youth Fund of Ministry of Education of China [grant number 19YJC630221].

References

Ahadzadeh, B., Abdar, M., Safara, F., Khosravi, A., Menhaj, M. B., & Suganthan, P. N. (2023). SFE: A simple, fast, and efficient feature selection algorithm for high-dimensional data. *IEEE Transactions on Evolutionary Computation*, *27*(6), 1896–1911.

Amaldi, E., & Kann, V. (1998). On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, *209*(1), 237 – 260.

Anzanello, M. J., Albin, S. L., & Chaovalitwongse, W. A. (2009). Selecting the best variables for classifying production batches into two quality levels. *Chemometrics and Intelligent Laboratory Systems*, *97*(2), 111 – 117.

Anzanello, M. J., Albin, S. L., & Chaovalitwongse, W. A. (2012). Multicriteria variable selection for classification of production batches. *European Journal of Operational Research*, *218*(1), 97 – 105.

Arthur, D., & Vassilvitskii, S. (2007). k-means++: the advantages of careful seeding. In N. Bansal, K. Pruhs, & C. Stein (Eds.), *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007* (pp. 1027–1035). SIAM.

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, *5*(4), 537–550.

Chien, C.-F., Hung, W.-T., Pan, C.-W., & Van Nguyen, T. H. (2022). Decision-based virtual metrology for advanced process control to empower smart production and an empirical study for semiconductor manufacturing. *Computers & Industrial Engineering*, *169*, 108245.

Cosson, R., Santana, R., Derbel, B., & Liefooghe, A. (2024). On bi-objective combinatorial optimization with heterogeneous objectives. *European Journal of Operational Research*, *319*(1), 89–101.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197.

Deng, Y., Du, S., Wang, D., Shao, Y., & Huang, D. (2023). A calibration-based hybrid transfer learning framework for rul prediction of rolling bearing across different machines. *IEEE Transactions on Instrumentation and Measurement*, *72*, 1–15. doi:10.1109/TIM.2023.3260283.

Gauchi, J.-P., & Chagnon, P. (2001). Comparison of selection methods of explanatory variables in PLS regression with application to manufacturing process data. *Chemometrics and Intelligent Laboratory Systems*, *58*(2), 171 – 193.

Guo, W., & Banerjee, A. G. (2017). Identification of key features using topological data analysis for accurate prediction of manufacturing system outputs. *Journal of Manufacturing Systems*, *43*, 225–234.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*, 1157–1182.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, *11*(1), 10–18.

Hancer, E. (2022). A multi-objective artificial bee colony algorithm for cost-sensitive subset selection. *Neural Computing and Applications*, *34*(20), 17523–17537.

Hancer, E. (2024). An improved evolutionary wrapper-filter feature selection approach with a new initialisation scheme. *Machine Learning*, *113*(8), 4977–5000.

Hancer, E., Xue, B., & Zhang, M. (2023). An evolutionary filter approach to feature selection in classification for both single- and multi-objective scenarios. *Knowledge-Based Systems*, *280*, 111008.

He, Z., Hu, H., Zhang, M., Zhang, Y., & Li, A.-D. (2022). A decomposition-based multi-objective particle swarm optimization algorithm with a local search strategy for key quality characteristic identification in production processes. *Computers & Industrial Engineering*, *172*, 108617.

Jiao, R., Xue, B., & Zhang, M. (2024). Solving multiobjective feature selection problems in classification via problem reformulation and duplication handling. *IEEE Transactions on Evolutionary Computation*, *28*(4), 846–860.

John, G. H., & Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence* UAI'95 (pp. 338–345). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*(1), 273 – 324.

Kozodoi, N., Lessmann, S., Papakonstantinou, K., Gatsoulis, Y., & Baesens, B. (2019). A multi-objective approach for profit-driven feature selection in credit scoring. *Decision Support Systems*, *120*, 106 – 117.

Kwak, N., & Choi, C.-H. (2002). Input feature selection for classification problems. *IEEE Transactions on Neural Networks*, *13*(1), 143–159.

Li, A.-D., He, Z., Wang, Q., & Zhang, Y. (2019). Key quality characteristics selection for imbalanced production data using a two-phase bi-objective feature selection method. *European Journal of Operational Research*, *274*(3), 978 – 989.

Li, A.-D., He, Z., & Zhang, Y. (2016). Bi-objective variable selection for key quality characteristics selection based on a modified NSGA-II and the ideal point method. *Computers in Industry*, *82*, 95 – 103.

Li, A.-D., Xue, B., & Zhang, M. (2020a). Multi-objective feature selection using hybridization of a genetic algorithm and direct multisearch for key quality characteristic selection. *Information Sciences*, *523*, 245 – 265.

Li, A.-D., Xue, B., & Zhang, M. (2021). Improved binary particle swarm optimization for feature selection with new initialization and search space reduction strategies. *Applied Soft Computing*, *106*, 107302.

Li, A.-D., Xue, B., & Zhang, M. (2023). Multi-objective particle swarm optimization for key quality feature selection in complex manufacturing processes. *Information Sciences*, *641*, 119062.

Li, W., Xiang, D., Tsung, F., & Pu, X. (2020b). A diagnostic procedure for high-dimensional data streams via missed discovery rate control. *Technometrics*, *62*(1), 84–100.

Liu, S., Wang, H., Peng, W., & Yao, W. (2022). A surrogate-assisted evolutionary feature selection algorithm with parallel random grouping for high-dimensional classification. *IEEE Transactions on Evolutionary Computation*, *26*(5), 1087–1101.

Mistry, K., Zhang, L., Neoh, S. C., Lim, C. P., & Fielding, B. (2017). A micro-GA embedded PSO feature selection approach to intelligent facial emotion recognition. *IEEE Transactions on Cybernetics*, *47*(6), 1496–1509.

Moradi, P., & Gholampour, M. (2016). A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. *Applied Soft Computing*, *43*, 117 – 130.

Nguyen, B. H., Xue, B., & Zhang, M. (2024). A constrained competitive swarm optimizer with an svm-based surrogate model for feature selection. *IEEE Transactions on Evolutionary Computation*, *28*(1), 2–16.

Nguyen, H. B., Xue, B., Liu, I., Andreae, P., & Zhang, M. (2016). New mechanism for archive maintenance in pso-based multi-objective feature selection. *Soft Computing*, *20*(10), 3927–3946.

Oh, I., Lee, J., & Moon, B. R. (2004). Hybrid genetic algorithms for feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(11), 1424–1437.

Oztekin, A., Al-Ebbini, L., Sevkli, Z., & Delen, D. (2018). A decision analytic approach to predicting quality of life for lung transplant recipients: A hybrid genetic algorithms-based methodology. *European Journal of Operational Research*, *266*(2), 639–651.

Pandiyan, V., Caesarendra, W., Tjahjowidodo, T., & Tan, H. H. (2018). In-process tool condition monitoring in compliant abrasive belt grinding process using support vector machine and genetic algorithm. *Journal of Manufacturing Processes*, *31*, 199–213.

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(8), 1226–1238.

Sahinkoc, H. M., & Ümit Bilge (2022). A reference set based many-objective co-evolutionary algorithm with an application to the knapsack problem. *European Journal of Operational Research*, *300*(2), 405–417.

Shi, J. (2023). In-process quality improvement: Concepts, methodologies, and applications. *IISE Transactions*, *55*(1), 2–21.

Simumba, N., Okami, S., Kodaka, A., & Kohtake, N. (2022). Multiple objective metaheuristics for feature selection based on stakeholder requirements in credit scoring. *Decision Support Systems*, *155*, 113714.

Song, X., Zhang, Y., Gong, D., & Sun, X. (2021). Feature selection using bare-bones particle swarm optimization with mutual information. *Pattern Recognition*, *112*, 107804.

Tian, Y., Cheng, R., Zhang, X., Su, Y., & Jin, Y. (2019). A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization. *IEEE Transactions on Evolutionary Computation*, *23*(2), 331–345.

Tizhoosh, H. (2005). Opposition-based learning: A new scheme for machine intelligence. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)* (pp. 695–701). volume 1.

Tran, B., Xue, B., & Zhang, M. (2018). A new representation in pso for discretization-based feature selection. *IEEE Transactions on Cybernetics*, *48*(6), 1733–1746.

Wang, P., Xue, B., Liang, J., & Zhang, M. (2023a). Feature clustering-assisted feature selection with differential evolution. *Pattern Recognition*, *140*, 109523.

Wang, Z., Gao, S., Zhou, M., Sato, S., Cheng, J., & Wang, J. (2023b). Information-theory-based nondominated sorting ant colony optimization for multiobjective feature selection in classification. *IEEE Transactions on Cybernetics*, *53*(8), 5276–5289.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, *1*(6), 80 – 83.

Wold, S., Sjöström, M., & Eriksson, L. (2001). PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, *58*(2), 109 – 130.

Xue, B., Zhang, M., & Browne, W. N. (2013). Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE Transactions on Cybernetics*, *43*(6), 1656–1671.

Xue, B., Zhang, M., Browne, W. N., & Yao, X. (2016). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, *20*(4), 606–626.

Xue, Y., Cai, X., & Neri, F. (2022). A multi-objective evolutionary algorithm with interval based initialization and self-adaptive crossover operator for large-scale feature selection in classification. *Applied Soft Computing*, *127*, 109420.

Yu, L., & Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, *5*(Oct.), 1205–1224.

Yuan, Y., Xu, H., Wang, B., & Yao, X. (2016). A new dominance relation-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, *20*(1), 16–37.

Zhang, D., Liu, Z., Jia, W., Liu, H., & Tan, J. (2022a). Path enhanced bidirectional graph attention network for quality prediction in multistage manufacturing process. *IEEE Transactions on Industrial Informatics*, *18*(2), 1018–1027.

Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, *11*(6), 712–731.

Zhang, Y., wei Gong, D., zhi Gao, X., Tian, T., & yan Sun, X. (2020). Binary differential evolution with self-learning for multi-objective feature selection. *Information Sciences*, *507*, 67–85.

Zhang, Y., Wang, Y.-H., Gong, D.-W., & Sun, X.-Y. (2022b). Clustering-guided particle swarm feature selection algorithm for high-dimensional imbalanced data with missing values. *IEEE Transactions on Evolutionary Computation*, *26*(4), 616–630.

Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength pareto evolutionary algorithm. In *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems. Proceedings of the EUROGEN'2001. Athens. Greece, September 19-21* (pp. 95–100).