Multiobjective feature selection for key quality characteristic identification in production processes using a nondominated-sorting-based whale optimization algorithm

Article *in* Computers & Industrial Engineering · September 2020

DOI: 10.1016/j.cie.2020.106852

An-Da Li
Tianjin University of Commerce

Multiobjective feature selection for key quality characteristic identification in production processes using a nondominated-sorting-based whale optimization algorithm

An-Da Li[a,*], Zhen He[b]

[a]School of Management, Tianjin University of Commerce, Tianjin 300134, China
[b]College of Management and Economics, Tianjin University, Tianjin 300072, China

Abstract

Identifying key quality characteristics (QCs) in production processes is essential for product quality control and improvement. This paper proposes a multiobjective wrapper-based feature selection (FS) method for key QC (KQC) identification on unbalanced production data using a novel modified nondominated-sorting-based whale optimization algorithm (MNSWOA) and the ideal point method (IPM). In the proposed approach, the FS problem is defined as maximizing the geometric mean (GM) measure and minimizing the feature (QC) subset size. To solve the defined FS problem, MNSWOA is adopted first to find a set of candidate solutions (feature subsets), and then IPM is adopted to select the final solution. In MNSWOA, a modified fast nondominated sorting approach is proposed to adapt the single objective whale optimization algorithm to the multiobjective scenario. Moreover, a uniform reference solution selection strategy and the mutation operations are embedded in MNSWOA to improve its search performance. Experimental results on four unbalanced production datasets show that the proposed FS method performs effectively and efficiently for KQC identification. Further comparisons show that MNSWOA obtains better search performance than benchmark multiobjective optimization methods, including a modified NSGA-II, SPEA2, MOEA/D, NSPSO and CMDPSO.

Keywords: feature selection, whale optimization algorithm, multiobjective optimization, classification, unbalanced data, quality improvement

1. Introduction

Modern production processes generally contain a large number of quality characteristics (QCs), including process parameters, assembly parameters and part parameters (Lee and Thornton, 1996). In practice, not all these QCs can be perfectly designed or controlled in the early production stages of products. Therefore,

☆https://doi.org/10.1016/j.cie.2020.106852

*Corresponding author.
Email addresses: adli@tjcu.edu.cn (An-Da Li), zhhe@tju.edu.cn (Zhen He)

selecting the key QCs (KQCs) that significantly affect the final product quality is an essential task for continuous product quality control and improvement (Peres and Fogliatto, 2018; Shang et al., 2014). With the rapid improvement of data acquisition technologies, high dimensional production data collected from the production lines appear more frequently in modern manufacturing industries, which lays foundation on the data-driven KQC identification approaches (Li et al., 2020b; Yang et al., 2019). Nevertheless, the high data dimensionality proposes significant challenges for traditional statistical methods, e.g., ANOVA or multiple linear regression methods (Pierre and Tuv, 2011).

Feature (variable) selection (FS) (Guyon and Elisseeff, 2003; Xue et al., 2016; Manochandar and Punniyamoorthy, 2018; Li and Jiang, 2019) in machine learning and data mining contexts aims to select the key input variables related to the output variable, i.e., the class label in a classification task, for high dimensional data. A number of studies (Anzanello et al., 2009, 2012; Li et al., 2016, 2019, 2020a) have applied FS methods to KQC identification problems in recent years. In these studies, the QCs are treated as features and the quality level (e.g., conforming or nonconforming) of products is treated as the class label.

From the perspective of optimization, FS can be defined as a multiobjective problem of maximizing/minimizing the classification accuracy/error and minimizing the feature subset size (Xue et al., 2013). Some studies (Eroglu and Kilic, 2017; Mistry et al., 2017) adopted single objective evolutionary computation (EC) techniques to solve this problem. However, before applying the single objective EC techniques, the two original optimization objectives should be integrated into one single objective, which is generally not an easy task as much domain knowledge is required. To narrow this gap, multiobjective FS methods (Xue et al., 2013; Li et al., 2016; Nguyen et al., 2016; Rosales-Prez et al., 2017) that adopt multiobjective EC techniques as the optimizer for the multiobjective FS problem have been proposed. As a nature of the multiobjective approaches, the multiobjective FS methods find a set of nondominated solutions (feature subsets) approximating the Pareto front. To reduce the number of solutions, multiobjective decision methods, e.g., the ideal point method (IPM) (Li et al., 2016, 2019), have been applied to further select the best solution from the nondominated solutions. The limitation of the multiobjective FS problem defined above is that the adopted feature importance measure, i.e., accuracy (or error), is a biased classification performance measure on the unbalanced data, which can lead to biased FS results. Therefore, other metrics (e.g., true positive rate (TPR) and true negative rate (TNR) (Tan et al., 2014), geometric mean (GM) of TPR and TNR (Li et al., 2019)) have been adopted in literature instead of accuracy to form FS selection problems.

Although EC-based FS methods have been widely studied, improving the optimization performance of an EC technique is always an essential task, since the search space of FS problems expands dramatically with the increase of data dimensionality (i.e., the number of features). Whale optimization algorithm (WOA) (Mirjalili and Lewis, 2016; Elaziz and Mirjalili, 2019) is a recently proposed EC technique inspired by the hunting behavior of humpbacks. It has been successfully applied to many optimization problems including FS (Mafarja and Mirjalili, 2017, 2018). WOA adopts three strategies (i.e., *encircling prey*, *spiral* *position updating* and *search for prey*) with adequate exploration and exploitation capabilities to update solutions during the optimization process. It has the advantages of fast convergence speed, a small number of parameters and easy implementation. Thus, for the FS problems that have a large solution space, WOA bears the capability of quickly reducing the irrelevant or redundant features to find a desirable feature subset. However, most existing WOA-based FS methods are still based on the single objective optimization scheme, which motivates us to develop a multiobjective WOA for FS.

In this paper, we develop a FS method (named MNSWOA-IPM) based on a multiobjective WOA for KQC identification in production processes. In this method, FS (KQC identification) is modeled as a multiobjective problem of maximizing GM and minimizing the feature subset size as suggested by Li et al. (2019). The GM measure is used instead of accuracy for feature importance evaluation since the production data are unbalanced, i.e., the number of instances (products) belong to different classes (product quality levels) differs substantially. To solve the defined multiobjective FS problem, a modified nondominated-sorting-based WOA (MNSWOA) is developed to obtain a set of nondominated solutions (i.e., feature/QC subsets), from which IPM is used to select the final solution (KQC set). The main contribution of this paper is that several strategies are adopted in MNSWOA which adapts WOA to an effective multiobjective optimization method. First, a modified fast nondominated sorting approach is proposed for MNSWOA to rank solutions in the multiobjective scenario. Compared with the traditional fast nondominated sorting approach, this sorting approach enhances the swarm diversity by modifying the ranks of duplicate solutions in the swarm. Second, as WOA needs a reference solution to update other solutions in the swarm at each iteration, we propose a uniform selection strategy that uniformly selects the reference solution for MNSWOA in the multiobjective scenario. Finally, mutation operations are conducted on the solutions in the swarm to improve the global search ability of MNSWOA. The experimental results on four unbalanced production datasets show that the proposed MNSWOA-IPM method obtains competitive KQC identification results. Further comparisons show that MNSWOA obtains better search performance than several representative multiobjective EC techniques for the defined FS problem.

The organization for the following parts of this paper is as follows. Section 2 provides the literature review on FS methods. Section 3 briefly describes the standard single objective WOA, the multiobjective decision method IPM and the definition of the KQC identification problem. Section 4 proposes the FS method MNSWOA-IPM. Section 5 explains the experimental design. Section 6 presents the experimental results and discussions. Section 7 further compares the search performance between MNSWOA and existing multiobjective EC techniques. Conclusions and future research interests are introduced in Section 8.

2. Literature review

FS can be divided into filter and wrapper approaches according to the feature importance evaluation strategies adopted (Xue et al., 2016). Filters evaluate features using the measures based on distance (Robnik-Šikonja and Kononenko, 2003), information theory (Peng et al., 2005; Yan and Jia, 2019; Yu and Liu, 2004), rough set theory (Zouache and Abdelaziz, 2018), etc. Wrappers evaluate features according to their classification performance on the instances, which requires a learning algorithm. Wrappers cost more computational time than filters as building a learning model is generally more time consuming than building a filter measure. Meanwhile, wrappers can generally produce better classification results than filters since they use a more straightforward way (i.e., classification performance) to evaluate the feature importance.

From the perspective of optimization, wrapper-based FS methods can be categorized into single objective and multiobjective approaches. The single objective approach usually defines FS as a single objective problem of maximizing the classification performance (e.g., accuracy) of feature subsets. Sequential forward selection (SFS) and sequential backward selection (SBS) (Kohavi and John, 1997) are two typical heuristic search strategies for the single objective FS problem. As FS has shown to be an NP-hard problem (Amaldi and Kann, 1998), EC techniques, e.g., genetic algorithms (GAs) (Oh et al., 2004; Min et al., 2006) and particle swarm optimization (PSO) (Xue et al., 2014b; Zhang et al., 2014), have also been widely used as the optimizer due to their good global search ability. In comparison, the multiobjective approach usually defines FS as a multiobjective problem of maximizing the classification performance and minimizing the feature subset size. The multiobjective approach has a better capability to obtain a concise learning model than the single objective one since reducing the number of selected features is explicitly defined as one optimization objective.

The multiobjective EC techniques have been increasingly applied to FS in recent years. The recently proposed multiobjective EC-based FS methods in literature are summarize in Table 1, which can help us gain a better understanding of the commonalities and differences among these methods. For balanced data, most studies have used accuracy/error to measure the classification performance and defined the two FS objectives as maximizing/minimizing the accuracy/error and minimizing the feature subset size. Various multiobjective EC techniques have been applied to optimize the two objectives. Nondominated sorting genetic algorithm II (NSGA-II) (Deb et al., 2002) is one of the most popular multiobjective evolutionary algorithms (MOEAs), which adopts a fast nondominated sorting approach to rank solutions and a crowding distance calculation method to maintain the population diversity. It has been widely used as the optimizer for FS applications including radial basis function neural networks (Guilln et al., 2009), hybrid fault diagnosis of gearbox (Li et al., 2011) and KQC identification (Li et al., 2016). In particular, Li et al. (2016) proposed a FS method for KQC identification using NSGA-II and IPM. The IPM is used to select the most desirable solution from the nondominated solutions found by NSGA-II. Besides NSGA-II, Rosales-Prez et al. (2017) proposed a novel

MOEA named evolutionary multiobjective model and instance selection (EMOMIS) to simultaneously select features and instances for support vector machines (SVMs). The multiobjective PSO (MOPSO) methods (Xue et al., 2013; Nguyen et al., 2016; Amoozegar and Minaei-Bidgoli, 2018) have also been widely used for FS. Xue et al. (2013) proposed two MOPSO-based FS methods called CMDPSOFS and NSPSOFS, based on the two PSO optimizers, CMDPSO and NSPSO. The results have shown that CMDPSOFS obtains better FS results than the three typical MOEAs, including NSGA-II, SPEA2 and PEAS. Nguyen et al. (2016) proposed a MOPSO method named ISRPSO for FS, where several local search operators were used to improve the search performance. The results have shown that ISRPSO obtains better FS results than CMDPSOFS, whereas it takes more computational time since the local search process needs additional time. Amoozegar and Minaei-Bidgoli (2018) proposed a MOPSO method named HMPSOFS for FS, which adopts a refining process to perform the local search to update the nondominated solutions. Moreover, FS methods based on the multiobjective differential evolution (DE) (Xue et al., 2014a) and multiobjective artificial bee colony (ABC) (Hancer et al., 2018; Zhang et al., 2019) have been proposed in recent years.

For unbalanced data, measuring the classification performance using accuracy (or error) may lead to biased FS results, since the majority class instances have a much higher impact on the accuracy (error) rate than the minority class instances. However, the classification accuracy (error) on the minority class is as critical as, or even more critical than that on the majority class (Bhowan et al., 2013). For example, in a quality control scenario, whether the defects (minority class instances) can be accurately detected is more critical than the detection of conforming products (majority class instances).

To address the data imbalance problem, alternative classification measures are used instead of accuracy to build multiobjective EC-based FS methods. The main consideration using these measures is giving the minority class instances a higher weight while measuring the classification performance. Pacheco et al. (2013); Huang et al. (2010) and Tan et al. (2014) adopted TPR and TNR (Type I and II errors) to form the multiobjective FS problems on unbalanced data, where NSGA-II (Pacheco et al., 2013; Huang et al., 2010) and the modified micro-genetic algorithm (MmGA) (Tan et al., 2014) are used as the optimizers. Ekbal and Saha (2012) proposed a NSGA-II based FS method using precision and recall measures. de la Hoz et al. (2014) and Zhu et al. (2017) adopted the Jaccard's coefficients on different classes to build the multiobjective FS problems.

Adopting the classification performance measures such as TPR & TNR, recall & precision and Jaccard's coefficients can substantially improve the FS performance on unbalanced data. However, compared with accuracy, adopting these measures actually increases the number of objectives to be optimized. As shown by Hughes (2005) and Ishibuchi et al. (2008), the performance of a multiobjective EC technique decreases as the number of objectives increases. To reduce the number of objectives as well as addressing the unbalanced data, Kozodoi et al. (2019); Li et al. (2019, 2020a) adopted integrated classification performance measures in building multiobjective FS methods. Specifically, Kozodoi et al. (2019) proposed a profit-driven FS

Table 1: Summary of multiobjective EC techniques for FS.

Literature	Optimization approach	Optimization objectives	Considering data imbalance	Application
Guilln et al. (2009)	NSGA-II	Error rate$^-$, Feature subset size$^-$	No	Radial basis function neural networks
Li et al. (2011)	NSGA-II	Error rate$^-$, Feature subset size$^-$	No	Hybrid fault diagnosis of gearbox
Li et al. (2016)	Modified NSGA-II (MNSGAII)	Accuracy$^+$, Feature subset size$^-$	No	KQC identification
Rosales-Prez et al. (2017)	EMOMIS	Accuracy$^+$, Reduction rate$^+$	No	Feature and instance selection for SVM
Xue et al. (2013)	CMDPSOFS, NSPSOFS	Error rate$^-$, Feature subset size$^-$	No	General classification problems
Nguyen et al. (2016)	ISRPSO	Error rate$^-$, Feature subset size$^-$	No	General classification problems
Amoozegar and Minaei-Bidgoli (2018)	HMPSOFS	Error rate$^-$, Feature subset size$^-$	No	General classification problems
Xue et al. (2014a)	Multiobjective DE (DEMOFS)	Error rate$^-$, Feature subset size$^-$	No	General classification problems
Hancer et al. (2018)	Bin-MOABC, Num-MOABC	Error rate$^-$, Feature subset size$^-$	No	General classification problems
Zhang et al. (2019)	Two-archive multiobjective ABC (TMABC-FS)	Error rate$^-$, Cost of features$^-$	No	General classification problems
Pacheco et al. (2013)	NSGA-II	Type I and II errors$^-$	Yes	Discriminant analysis in two-class classification
Huang et al. (2010)	NSGA-II	Accuracy$^+$, TPR$^+$, TNR$^+$	Yes	Customer churn prediction in telecommunications
Tan et al. (2014)	Modified micro-genetic algorithm (MmGA)	TPR$^+$, TNR$^+$, Feature subset size$^-$	Yes	Feature selection for neural network models
Ekbal and Saha (2012)	NSGA-II	Precision$^+$, Recall$^+$	Yes	Named entity recognition
de la Hoz et al. (2014)	NSGA-II	Jaccards coefficients$^+$	Yes	Network anomaly detection
Zhu et al. (2017)	Improved NSGA-III (I-NSGA-III)	Jaccards coefficients$^+$	Yes	Intrusion detection
Kozodoi et al. (2019)	NSGA-II	Expected maximum profit$^+$, Feature subset size$^-$	Yes	Credit scoring
Li et al. (2019, 2020a)	IDMS, GADMS	GM$^+$, Feature subset size$^-$	Yes	KQC identification

$^+$ denotes to be maximized and $^-$ denotes to be minimized.

method using NSGA-II for credit scoring. In this method, the confusion matrix with costs is used for calculating the profits of feature subsets. Since the classification accuracy of the minority class instances is given a higher weight when calculating the profits, the data imbalance problem is handled. Li et al. (2019) proposed an improved direct multisearch (IDMS) based FS method for KQC identification. Similarly, Li et al. (2020a) proposed a FS method based on a hybrid optimization method named GADMS that combines a GA and direct multisearch for KQC identification. In these two methods, the GM of TPR and TNR is used to measure the classification performance of feature subsets. Since a low value of either TPR or TNR significantly decreases the GM value, GM is an effective classification performance measure for unbalanced data.

3. Preliminaries

This section briefly introduces the single objective WOA, IPM and the KQC identification problem, which are the basic concepts of the proposed FS (KQC identification) method in this paper.

3.1. Whale optimization algorithm (WOA)

WOA (Mirjalili and Lewis, 2016) is a swarm-based optimization method inspired by the hunting behavior of humpback whales, i.e., a swarm of humpbacks changes their positions (solutions) hunting for preys. To be specific, WOA adopts three solution updating strategies, i.e., *encircling prey*, *spiral position updating* and *search for prey*, mimicking the hunting behavior of whales. Given the swarm size N and the solution dimensionality D, the position of whale i at the tth iteration can be denoted by $\mathbf{X}_i^t = (x_{i,1}^t, x_{i,2}^t, ..., x_{i,D}^t)$, $i = 1, ..., N$. Then, the three solution updating strategies can be depicted as follows.

3.1.1. Encircling prey

The encircling prey strategy updates whale i's ($i = 1, ..., N$) position from $\mathbf{X}_i^t$ to $\bar{\mathbf{X}}_i^t$ as

$$\bar{\mathbf{X}}_i^t = \mathbf{X}^{*t} - A_i \cdot |C_i \cdot \mathbf{X}^{*t} - \mathbf{X}_i^t|, \tag{1}$$

where $\mathbf{X}^{*t}$ (called reference solution in this paper) is the best position obtained at the tth iteration, $A_i \cdot |C_i \cdot \mathbf{X}^{*t} - \mathbf{X}_i^t|$ decides the step length of the encircling prey, and $|\bullet|$ means calculating the absolute value for each element in the vector $\bullet$. A_i and C_i are two random values obtained as

$$A_i = 2a_t \cdot r_i - a_t, \tag{2}$$

$$C_i = 2 \cdot r_i', \tag{3}$$

where r_i and r_i' denote two random values in $[0, 1]$ from the uniform distribution, and $a_t = 2 - 2t/T_{max}$ is a parameter linearly decreasing from 2 to 0 during the iterations, which means that whales gradually narrow the encircling scope.

3.1.2. Spiral position updating

The spiral position updating strategy updates whale i's ($i = 1, ..., N$) position from $\mathbf{X}_i^t$ to $\bar{\mathbf{X}}_i^t$ as

$$\bar{\mathbf{X}}_i^t = \mathbf{D}' \cdot e^{bl_i} \cdot \cos(2\pi l_i) + \mathbf{X}^{*t}, \tag{4}$$

where $\mathbf{D}' = |\mathbf{X}^{*t} - \mathbf{X}_i^t|$ is the distance vector between position $\mathbf{X}_i^t$ and the reference position $\mathbf{X}^{*t}$, l_i is a random value in $[-1, 1]$, and b is a user-defined parameter whose default value is 1.

3.1.3. Search for prey

The search for prey strategy is adopted in WOA to enhance the global search ability. It updates whale i's ($i = 1, ..., N$) position from $\mathbf{X}_i^t$ to $\bar{\mathbf{X}}_i^t$ as

$$\bar{\mathbf{X}}_i^t = \mathbf{X}_{rand}^t - A_i \cdot |C_i \cdot \mathbf{X}_{rand}^t - \mathbf{X}_i^t|, \tag{5}$$

where $\mathbf{X}_{rand}^t$ is a position (solution) randomly selected from the swarm, and A_i and C_i are obtained by Eqs. (2) and (3). It can be seen that the only difference between the search for prey strategy and the encircling prey strategy is that different reference solutions (i.e., $\mathbf{X}^{*t}$ and $\mathbf{X}_{rand}^t$) are used to update the whale's position.

3.1.4. Overall WOA procedure

WOA integrates the three solution updating strategies mentioned above to evolve new solutions. At each iteration, one of the three strategies is selected to update the position (solution) for a whale. Considering whale i, a random number $p_i \in [0, 1]$ is first generated to decide whether the spiral position updating strategy is conducted ($p_i \geq 0.5$) or not ($p_i < 0.5$). If $p_i < 0.5$, then the absolute value $|A_i|$ is used to decide the whether the encircling prey strategy ($|A_i| < 1$) or the search for prey strategy ($|A_i| \geq 1$) is chosen to update the position.

3.2. Ideal point method (IPM)

IPM is a widely used multiobjective decision method that selects one best compromise solution from a set of candidate solutions (Freimer and Yu, 1976). In IPM, an ideal point is first defined in the objective space, and then the solution (in the candidate set) with the minimum distance to the ideal point is selected as the best compromise solution. For the FS problem, Li et al. (2016) applied IPM to select the final feature subset from a set of nondominated solutions (candidate feature subsets) found by NSGA-II.

Let Ω be the set of candidate solutions (feature subsets) found by a multiobjective approach, $x \in \Omega$ be a solution in the set, and $f_i(x), i = 1, ..., M$ be the objective function values (M denotes the number of objectives). Then, the selection process (Li et al., 2016) for the best compromise solution (final feature subset) using IPM can be briefly described as 3 steps. First, the ith objective function value $f_i(x)$ for each solution $x \in \Omega$ is normalized as

$$f_i^N(x) = (f_i(x) - \overline{f_i(x)})/\sigma(f_i(x)), i = 1, ..., M \tag{6}$$

where $\overline{f_i(x)}$ and $\sigma(f_i(x)))$ are the mean and standard deviation of the ith objective function value for solutions in Ω. Second, the ideal point is defined as (f_1^*,..., f_M^*), where

$$f_i^* = \min_{x \in \Omega} f_i^N(x), i = 1, ..., M. \tag{7}$$

Finally, the best compromise solution x^* with the minimum Euclidean distance to the ideal point is obtained as

$$x^* = \arg\min_{x \in \Omega} \sqrt{\sum_{i=1}^{M} (f_i^N(x) - f_i^*)^2}. \tag{8}$$

3.3. The KQC identification problem

Let a dataset collected from production processes be $\Theta^{M \times (D+1)}$, which contains M instances (products), a set $\mathbb{F} = \{F_1, F_2, ..., F_D\}$ of features (QCs), and one class label (the quality level indicator) $C \in \{-1, 1\}$, where $C = -1$ indicates the majority class (e.g., the normal quality) and $C = 1$ indicates the minority class (e.g., the premium quality). Then, KQC identification can be defined as a FS problem that selects a set $\mathbb{F}_s \subseteq \mathbb{F}$ of features with a powerful predictive ability of the product quality while reducing as many as possible of irrelevant or redundant features .

In this paper, we model the FS task as a multiobjective optimization problem that maximizes the classification performance and minimizes the feature subset size based on the wrapper framework. GM is an appropriate measure to evaluate the classification performance on the unbalanced production data because either a low value of TPR or TNR significantly decreases the GM value. As suggested by Li et al. (2019), in this paper, we use GM to measure the classification performance and define the FS (KQC identification) problem as

$$\begin{array}{ll} \min & \{f_1(\mathbb{F}_s) = 1 - \mathrm{GM}(\mathbb{F}_s); f_2(\mathbb{F}_s) = |\mathbb{F}_s|\} \\ \text{s.t.} & \mathbb{F}_s \subseteq \mathbb{F} \end{array}, \tag{9}$$

where GM($\mathbb{F}_s$) denotes the GM value obtained by $\mathbb{F}_s$ and $|\mathbb{F}_s|$ denotes the size of $\mathbb{F}_s$, i.e., the number of features in $\mathbb{F}_s$. To evaluate the GM value of a feature subset, we use inner 5-fold cross validation (CV) on the training set as suggested by Kohavi and John (1997).

4. Proposed FS approach

In this section, a two-phase optimization method named MNSWOA-IPM is proposed to solve the FS problem defined in Eq. (9). In MNSWOA-IPM, MNSWOA is first proposed to find a set of nondominated solutions and then IPM (see Section 3.2) is used to select the most desirable solution (final feature subset) from the nondominated solutions. The details of MNSWOA are given as follows.

Algorithm 1: Procedure of MNSWOA.

Input: Training set with a set $\mathbb{F}$ of features, swarm size N, maximum number of iterations T_{max} ;
Output: A set Ω of nondominated solutions ;

/* Initialize iteration counter t and swarm $\mathbb{S}^t$, where $\mathtt{X}_i^t$ $(i = 1, ..., N)$ denotes whale i's position (solution). */

1 $t \leftarrow 0$, $\mathbb{S}^t \leftarrow \{\mathbf{X}_1^t, \mathbf{X}_2^t, ..., \mathbf{X}_N^t\}$;
2 Evaluate the objective function values of each solution in $\mathbb{S}^t$;
3 Sort solutions in $\mathbb{S}^t$ with the *modified fast nondominated sorting* (see Section 4.3) approach ;
4 **while** $t < T_{max}$ **do**
5     $\mathbb{P} \leftarrow \mathbb{S}^t$;
6   **for each** $\boldsymbol{X}_i^t \in \mathbb{S}^t$ **do**
7         $A_i \leftarrow$ Using Eq. (2) ;
8         $p_i \leftarrow rand()$; /* $rand()$ denotes a random value in $[0, 1]$ */
9   Select a reference solution $\mathbf{X}^{*t}$ from $\mathbb{P}$ with the *uniform selection* (see Section 4.4) strategy ;
10   **if** $p_i < 0.5$ **then**
11   **if** $|A_i| < 1$ **then**
12   Update $\mathbf{X}_i^t$ using Eq. (1) ; /* Encircling prey */
13   **else**
14   Update $\mathbf{X}_i^t$ using Eq. (5) ; /* Search for prey */
15   **end**
16   **else**
17   Update $\mathbf{X}_i^t$ using Eq. (4) ; /* Spiral position updating */
18   **end**
19   **end**
20   Conduct *mutation operations* (see Section 4.5) on solutions in $\mathbb{S}^t$;
21   Evaluate the objective function values of each solution in $\mathbb{S}^t$;
22     $\mathbb{S}^c \leftarrow \mathbb{P} \cup \mathbb{S}^t$;
23   Sort solutions in $\mathbb{S}^c$ with the *modified fast nondominated sorting* (see Section 4.3) approach;
24     $\mathbb{S}^{t+1} \leftarrow$ The first N solutions in the sorted $\mathbb{S}^c$;
25     $t \leftarrow t + 1$;
26 **end**
27 **return** $\Omega \leftarrow$ nondominated solutions in $\mathbb{S}^t$;

4.1. Overall procedure of MNSWOA

The procedure of MNSWOA is shown in Algorithm 1, which includes several key points. First, to address the multiobjective FS problem, the modified fast nondominated sorting approach for improving the swarm diversity is proposed to rank solutions (lines 3 and 23). Second, the solution updating strategies (lines 10 to 18) in WOA are adopted as the main mechanism to update solutions during the evolutionary process of MNSWOA. Third, different from the single objective problems, multiobjective ones generally have a set of best (nondominated) solutions. Therefore, the uniform selection strategy (line 9) is proposed to select the reference solution $\mathbf{X}^{*t}$ for the encircling prey and spiral position updating strategies. Fourth, to further enhance the global search ability, mutation operations (line 20) are conducted on the solutions in the swarm. Finally, MNSWOA adopts an elitism strategy similar to NSGA-II (Deb et al., 2002), where the best N solutions from the union (i.e., $\mathbb{P} \cup \mathbb{S}^t$) of parent and updated solutions are kept for the following iteration (line 24). In the following parts, the key elements of MNSWOA are described in more detail.

4.2. Solution encoding and initialization

In this paper, each solution (a whale's position) in the swarm is encoded as a real-valued vector $\mathbf{X} = (x_1, x_2, ..., x_D)$, where D is the number of original features (QCs), and $x_j \in [-1, 1]$ ($j = 1, 2, ..., D$) denotes whether the jth feature is selected by $\mathbf{X}$. Specifically, $x_j > 0$ denotes the jth feature is selected and $x_j \leq 0$ denotes the jth feature is eliminated. Note that if x_j is updated to be a value larger than 1 we amend it to 1, else if it is updated to be a value smaller than -1 we amend it to -1. To initialize the swarm, the random initialization method is used, where each element x_j in a solution $\mathbf{X}$ is set as a random value in $[-1, 1]$.

4.3. Modified fast nondominated sorting approach

The fast nondominated sorting approach proposed by Deb et al. (2002) is one of the most commonly used solution ranking method in multiobjective EC techniques. This approach iteratively selects and eliminates nondominated solutions from the population (swarm) to rank solutions. Finally, each solution $\mathbf{X}_i$ is assigned to a nondomination rank $Rank(\mathbf{X}_i) \geq 1$ which denotes the nondominated front the solution lies on. A smaller rank value denotes a better fitness of the solution. However, the traditional fast nondominated sorting approach is not able to identify and reduce the duplicate solutions in the population (Li et al., 2016). This can lead to the result that a large number of high quality but redundant solutions exist in the swarm during the evolutionary process, which reduces the diversity of population. To solve this problem, a modified sorting approach (Li et al., 2016) is proposed to detect and reduce the duplicate solutions in the population. However, this sorting approach may not work well on real-coded solutions for detecting the duplicate solutions. For example, the real-coded solutions $\mathbf{X}_a = (-0.5, 0.5, 1, -0.2)$ and $\mathbf{X}_b = (-0.2, 0.5, 0.5, -0.2)$ actually refer to the same feature subset. Nevertheless, the sorting approach proposed by Li et al. (2016) cannot identify these two repeated solutions, since in this approach two solutions are considered to be duplicate only when the two solutions are exactly the same (i.e., $\mathbf{X}_a = \mathbf{X}_b$ in the given example). Therefore, the sorting approach (Li et al., 2016) should be further modified for MNSWOA which adopts real-coded solutions.

In this paper, we propose a modified fast nondominated sorting approach for MNSWOA. The procedure of the proposed sorting approach is shown in Algorithm 2. First, we use the traditional fast nondominated sorting approach (Deb et al., 2002) to assign a nondomination rank to each solution in the combined swarm $\mathbb{S}^c$. The solutions are sorted according to the assigned ranks in ascending order. Second, we propose a process to increase (modify) the nondomination ranks of the duplicate solutions in the combined swarm, and then resort the solutions based on the modified ranks (see lines 4 to 11). Since a lower nondomination rank denotes a better performance of a solution, this process lowers the fitnesses of the duplicate solutions. As shown in Algorithm 1 (line 24), the first (best) N solutions in the combined swarm $\mathbb{S}^c$ are always kept

Algorithm 2: Procedure of the modified fast nondominated sorting approach.

Input: Swarm $\mathbb{S}^c = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_{N^c}\}$, the swarm size N ;
Output: A set $\mathbb{S}_{sorted}$ of sorted solutions ;

1 $\mathbb{S}_{sorted} \leftarrow \emptyset$, $\mathbb{S}_r \leftarrow \emptyset$;
2 $\mathbb{S}' \leftarrow$ Sort solutions in $\mathbb{S}^c$ with the traditional fast nondominated sorting approach ;
3 $r_{max} \leftarrow$ Maximum nondomination rank of solutions in $\mathbb{S}'$;
4 **for** $i \leftarrow 1$ *to* N^c **do**
5   **if** $Decode(\boldsymbol{X}_i) \in Decode(\mathbb{S}_{sorted})$ **then**
        /* If feature subset $Decode(\mathtt{X}_i)$ is redundant with any feature subset in $Decode(\mathbb{S}_{sorted})$, modify $\mathtt{X}_i$'s rank $Rank(\mathtt{X}_i)$. ($Decode(\mathtt{X}_i)$ denotes the feature subset decoded from solution $\mathtt{X}_i$, and $Decode(\mathbb{S}_{sorted})$ denotes the set of feature subsets decoded from $\mathbb{S}_{sorted}$.) */
6         $Rank(\mathbf{X}_i) \leftarrow Rank(\mathbf{X}_i) + r_{max}$;
7         $\mathbb{S}_r \leftarrow \mathbb{S}_r \cup \{\mathbf{X}_i\}$;
8   **else**
9         $\mathbb{S}_{sorted} \leftarrow \mathbb{S}_{sorted} \cup \{\mathbf{X}_i\}$;
10   **end**
11 **end**
12 $\mathbb{S}_{sorted} \leftarrow \mathbb{S}_{sorted} \cup \mathbb{S}_r$;
13 $r_n \leftarrow$ The nondomination rank of the Nth solution in $\mathbb{S}_{sorted}$;
14 $\mathbb{S}_{sorted} \leftarrow$ Resort solutions in $\mathbb{S}_{sorted}$ whose nondomination ranks equal r_n in terms of the objective function f_2 in ascending order ;
15 **return** $\mathbb{S}_{sorted}$;

for the next iteration, this nondomination rank increasing process lowers the probabilities of the duplicate solutions to be added to the next iteration. Thus, the duplicate solutions in the swarm of the next iteration are reduced compared with the situation that does not adopt the nondomination rank increasing process, which improves the swarm diversity. Finally, to distinguish the goodness of either two solutions, a criterion should be further defined to compare the solutions on the same nondominated front. In this paper, we use the objective function f_2 (feature subset size) as the criterion. In other words, for solutions on the same nondominated front we prefer the solutions with fewer selected features to be added to the next iteration to improve the feature reduction capability of the algorithm. As shown in Algorithm 2 (line 14), solutions on the r_nth nondominated front is sorted according to objective function f_2.

4.4. Uniform selection of the reference solution

The encircling prey and spiral position updating strategies of MNSWOA require a reference solution $\mathbf{X}^{*t}$ to update solutions. The reference solution is the best solution obtained so far in the single objective scenario. In the multiobjective scenario, each nondominated solution can be a candidate of the reference solution. Therefore, it is required to propose a proper reference solution selection strategy for encircling prey and spiral position updating strategies in the MNSWOA method.

Essentially, the encircling prey and spiral position updating strategies can be seen as searching in the space around the selected reference solution. Therefore, to guarantee stable and effective optimization performance, it is preferred that all nondominated solutions are uniformly selected as the reference. To achieve this goal, we propose the uniform selection strategy. This strategy adopts a parameter η for each solution in the swarm that records the number of times the solution being selected as the reference. The nondominated solution (whose nondomination rank equals 1) with the minimum η value is always selected as the reference solution (if more than one solution matches the condition, randomly select one from the matched solutions). The parameter η for a solution is added by 1 every time it is selected as the reference.

4.5. Mutation operations

In WOA, the search for prey strategy is designed to enhance the global search ability, where a solution is updated by exchanging information with another solution randomly selected in the swarm. However, as one nature of the swarm-based EC techniques, all solutions in the swarm tend to be similar during the evolutionary process, which limits the global search ability of the search for prey strategy. In this paper, to further improve the global search ability of MNSWOA, the mutation operations are conducted on the solutions in the swarm after the three WOA-based solution updating strategies. Let $\mathbf{X} = (x_1, x_2, ..., x_D)$ be a solution and j_m be a random integer between 1 and D. Then $\mathbf{X}$ mutates to $\mathbf{X}' = (x'_1, x'_2, ..., x'_D)$ where each element x_j $(j = 1, 2, ..., D)$ is

$$x'_j = \begin{cases} -x_j, & j = j_m \\ x_j, & j \neq j_m \end{cases} . \tag{10}$$

According to the encoding strategy of MNSWOA, a mutation operation changes the state that the j_mth feature is selected or not. To balance the global and local search abilities, we set the mutation rate $p_m = 0.5$, meaning that each solution in the swarm has a probability of 0.5 to mutate.

4.6. Time complexity analysis

The time complexity of an EC technique is composed of two parts, the objective function evaluation process and the evolutionary process. In most real world optimization applications, the time complexity of the objective function evaluation process is much lower than that of the evolutionary process, and thus can be neglected when analyzing the time complexity. However, evaluating the objective functions is very time consuming in an EC-based wrapper method, since a learning algorithm is involved in the evaluation process of objective functions. Moreover, the evaluation time varies on different solutions, since it is affected by the data dimensionality indicated by the solution (feature subset). Because the time complexity of the objective function evaluation is problem-related and is hard to be measured in a wrapper-based FS method, in the following paragraph we will analyze the time complexity of MNSWOA without considering the objective function evaluation process.

According to Algorithm 1, the time complexity of MNSWOA at each iteration is governed by the modified fast nondominated sorting approach. The time complexity of this sorting approach is $O(N^2) + O(ND)$ (see Appendix A), which shows that the complexity is decided by the swarm size N and the number of original features D. Compared with NSGA-II, which uses the traditional fast nondominated sorting approach, MNSWOA has a similar time complexity if $D \leq N$, and a higher time complexity if $D > N$. This means that our method is not as time efficient as NSGA-II if the number of features is much larger than the swarm size. However, since the number of features of the production datasets in the experiments is smaller than or similar to the swarm size, our method and NSGA-II have similar efficiency in this paper.

As mentioned above, evaluating the objective functions in a wrapper-based FS approach is time consuming. Thus, in this paper, we further adopt a caching strategy to reduce the computational time of the objective function evaluation process in MNSWOA. Specifically, a cache is used to store the evaluated solutions and their objective function values. During the evolutionary process, the objective function values of a solution are directly obtained from the cache if the solution has already been evaluated, or the common wrapper-based objective function evaluation process is conducted.

5. Experimental design

This section presents the details of the experimental design including datasets, benchmark methods, experimental configuration and performance metrics.

5.1. Datasets

Four datasets, i.e., LATEX, ADPN, SPIRA and PAPER, collected from production processes are used in the experiments. LATEX, ADPN and SPIRA were first used by Gauchi and Chagnon (2001) in FS for regression. LATEX was collected from emulsion polymerisation batch operations of latex production, ADPN was collected from the adiponitrile production process, and SPIRA was collected from the fermentation process of an antibiotic production. In these datasets, production process parameters, such as temperatures, concentrations and pressures are the QCs. PAPER was first used by Wold et al. (2001) for validating the partial least squares regression tools. It was collected from the process of newspapers and magazines recycling, and the QCs include concentration and temperature measures. Anzanello et al. (2009) divided the instances in the four datasets into two classes, the premium quality (the minority class) and the regular quality (the majority class), in terms of the threshold of the quality indication variable y provided by Gauchi and Chagnon (2001) and Wold et al. (2001). So, the datasets are adapted to the classification problems addressed in this paper. Table 2 shows the details of the four datasets.

Table 2: Details of the production datasets.

Dataset	Number of instances	Number of minority class instances	Number of majority class instances	Number of features (QCs)
LATEX	262	78	184	117
ADPN	71	20	51	100
SPIRA	145	50	95	96
PAPER	384	33	351	54

5.2. Benchmark methods

Two conventional FS methods and four recently proposed EC-based FS methods are used as the benchmark methods. The two conventional methods are SFS and SBS (Kohavi and John, 1997), which use greedy search strategies to optimize the FS problem. The four EC-based methods are WOA-CM (Mafarja and Mirjalili, 2018), NSGAII-IPM (Li et al., 2016), NSPSOFS (Xue et al., 2013) and CMDPSOFS (Xue et al., 2013). WOA-CM (Mafarja and Mirjalili, 2018) is a single objective WOA method proposed for FS. It embeds the genetic operations in the evolutionary process to improve the search performance. Moreover, it adopts a fitness function combining the accuracy with the percentage of selected features. NSGAII-IPM is a multiobjective FS method recently proposed for KQC identification. It adopts a modified NSGA-II (MNSGAII) with a modified fast nondominated sorting approach to improve the search performance, and adopts a multiobjective decision method, i.e., IPM, to select the best solution from the set of nondominated solutions found by MNSGAII. NSPSOFS and CMDPSOFS are two recently proposed MOPSO-based FS methods (Xue et al., 2013). NSPSOFS introduces the nondominated sorting strategy to PSO and CMDPSOFS applies the strategies of crowding, mutation and dominance to PSO to address the multiobjective FS problems. As NSPSOFS and CMDPSOFS do not use a multiobjective decision method to further reduce the number of solutions, IPM is adopted to further select the final solution from the set of returned nondominated solutions, similar to that in MNSWOA-IPM and NSGAII-IPM. The multiobjective FS problem in NSGAII-IPM, NSPSOFS and CMDPSOFS is defined as maximizing the accuracy and minimizing the feature subset size.

5.3. Parameter settings

In proposed MNSWOA-IPM, we set the swarm size $N = 100$ and the maximum number of iterations $T_{max} = 100$. The user-defined parameter b in spiral position updating of MNSWOA-IPM is set as 1 as suggested by Mirjalili and Lewis (2016). For a fair comparison, in the benchmark EC approaches (i.e., WOA-CM, NSGAII-IPM, NSPSOFS and CMDPSOFS), the settings of the swarm (or population) size N and the maximum number of iterations T_{max} are the same as MNSWOA-IPM, i.e., $N = 100$ and $T_{max} = 100$. In WOA-CM, the crossover and mutation rates are set as 1 and 0.9 as suggested by Mafarja and Mirjalili (2018). In NSGAII-IPM, the crossover and mutation rates are $p_c = 0.9$ and $p_m = 1/D$ (D is the number of original features) as suggested by Li et al. (2016). For NSPSOFS and CMDPSOFS, the parameter settings except for the swarm size are the same as that used by Xue et al. (2013). Specifically, in NSPSOFS, the maximum velocity $v_{max} = 0.6$, the constants of acceleration $c_1 = c_2 = 1.49618$, and the inertia weight $w = 0.7298$. In CMDPSOFS, $v_{max} = 0.6$, c_1 and c_2 are random values between 1.5 and 2.0, w is a random value between 0.1 and 0.5, and the mutation rate is $1/D$.

5.4. Experimental configuration

To evaluate the performance of the FS methods for KQC identification, 10-fold CV is used to conduct the experiments. In a 10-fold CV process, the original dataset is divided into 10 folds and 10 validation runs are conducted. For each run, one fold (10%) is used as the test set and the remaining 9 folds (90%) are combined as the training set. The training set is used to build a learning model (i.e., classifier) with the selected features of the FS method. The classification performance of the learnt model on the test set is then used to evaluate the FS effectiveness. Moreover, for all the stochastic methods (the methods except SFS and SBS), we repeat the 10-fold CV 3 times with different running seeds, which yields $3 \times 10 = 30$ runs of experiments for each method on each dataset. The Wilcoxon signed-rank test (Wilcoxon, 1945) is used to compare the results of the 30 experimental runs between proposed MNSWOA-IPM and the benchmark methods. Note that, the training set is also used by the FS method to obtain the final feature subset. It is mainly used by the FS method for evaluating the classification performance (goodness) of a solution (feature subset) during the optimization process. Specifically, inner 5-fold CV is used based on the training set to estimate the classification performance (i.e., objective function f_1 in the proposed method). In an inner 5-fold CV process, the training set is divided into 5 folds to generate 5 pairs of the inner training and test sets. For each pair, the inner training set is used to build an inner learning model and the model's classification performance on the inner test set is obtained. Finally, the average classification performance over the 5 inner test sets is calculated to evaluate the goodness of a given feature subset. The aforementioned procedure to validate a FS method using 10-fold CV is shown in Figure 1, which also shows the relation between the 10-fold CV and inner 5-fold CV. For a more detailed description of the inner 5-fold CV in a wrapper-based FS method, please see Kohavi and John (1997).

All the experiments are run on a PC with 3.60GHz CPU and 16GB RAM. The naive Bayesian (NB) classifier (John and Langley, 1995) is adopted as the learning algorithm, since it is a simple and high performance classifier. In the experiments, the NB classifier is directly invoked from the Waikato Environment for Knowledge Analysis (WEKA) (Hall et al., 2009). The imbalance ratio (the ratio of majority class instances to minority class instances) on the PAPER dataset is much higher than that on the other datasets, which has a negative effect that the trained classifier is unreliable itself (Bhowan et al., 2013). To combat this negative effect, we use the WEKA configuration that assigns a higher weight (equal to the imbalance ratio) to the training instances of the minority class when training the NB classifier. MNSWOA-IPM, WOA-CM,

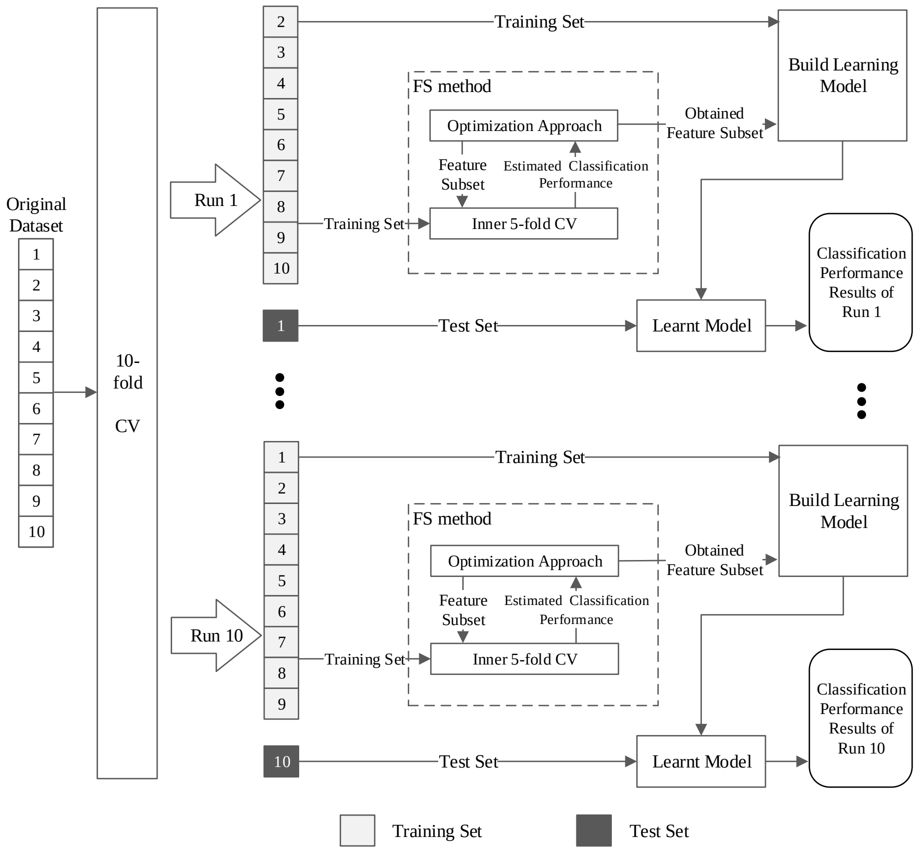

Figure 1: Procedure to validate a FS method using 10-fold CV.

NSGAII-IPM, NSPSOFS and CMDPSOFS are implemented in MATLAB 2016b, and SFS and SBS are employed in WEKA with default settings.

5.5. Performance metrics

The performance metrics to evaluate the FS methods are "accuracy", "TPR & TNR", "AUC" (area under a receiver operating characteristic curve) (Fawcett, 2006) , "the number of selected features" and "computational time". Accuracy is the most commonly used performance metric for classification. In this paper, TPR & TNR and AUC are used in addition to accuracy to comprehensively measure the classification performance of the selected features of each method on the unbalanced production datasets. The number of selected features can measure the dimensionality reduction effectiveness of the methods. The computational time measures the efficiency of each method.

Since binary classification problems are addressed in this paper, we let TP, TN, FP and FN be the numbers of true positive, true negative, false positive and false negative instances, respectively. Then, accuracy is obtained as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{11}$$

In terms of TPR & TNR, TPR measures the percentage of correctly classified positive instances (the minority class), and TNR measures the percentage of correctly classified negative instances (the majority class). These two metrics can well address the data imbalance problem, since the classification performance on both the majority and minority classes are evaluated by TNR and TPR. The two metrics are defined as

$$TPR = \frac{TP}{TP + FN}, \tag{12}$$

$$TNR = \frac{TN}{TN + FP}. \tag{13}$$

A receiver operating characteristic (ROC) curve is a two dimensional graph plotting the TPR against the false positive rate (FPR) for different decision thresholds of a classifier (Fawcett, 2006). The higher the curve is, the better the classification performance is. Since the ROC curve is not sensitive to the change of class distribution, it is a good classification performance indicator for unbalanced data. AUC is a scalar value that denotes the area under an ROC curve. It simplifies the comparison of ROC curves. The value of AUC is in $[0.5, 1]$, and a larger value denotes better classification performance. In this paper, the calculation method proposed by Fawcett (2006) is used to obtain the AUC value.

6. Results and discussion

Tables 3 to 6 show the FS (KQC identification) results on the accuracy, TPR & TNR, and AUC metrics, which are the larger the better. Tables 7 and 8 show the results on the number of selected features and computational time metrics, which are the smaller the better. In each of these tables, the mean and standard deviation of the performance metric values over the 30 runs are shown for each method. The Wilcoxon signed-rank test is used to compare the proposed method with each benchmark method, where "↑↑" or "↓↓" denotes the proposed MNSWOA-IPM obtains better or worse results at a significance level of $\alpha = 0.05$ and "↑" or "↓" denotes MNSWOA-IPM obtains better or worse results at a significance level of $\alpha = 0.1$. The average results on the performance metrics over the four datasets are listed on the "AVERAGE" row for each method.

6.1. Comparisons on obtained accuracy rates

According to Table 3, MNSWOA-IPM generally obtains similar or higher accuracy rates than the benchmark methods on the first three datasets, i.e., LATEX, ADPN and SPIRA. On LATEX, MNSWOA-IPM obtains a significantly higher accuracy rate than SFS and SBS at the significance level of 0.05, and obtains a significantly higher accuracy rate than WOA-CM at the significance level of 0.1. On ADPN, MNSWOA-IPM obtains a significantly higher accuracy rate than SFS, SBS, WOA-CM and NSPSOFS at the significance level of 0.05. On SPIRA, MNSWOA-IPM obtains a significantly higher accuracy rate than the benchmark methods except SFS at the significance level of 0.05. On the last dataset PAPER, MNSWOA-IPM obtains a slightly lower accuracy rate than other methods, and the statistical significance test results denote that MNSWOA-IPM obtains a significantly lower accuracy rate than two methods, i.e., SBS and WOA-CM. To sum up, the proposed MNSWOA-IPM can obtain comparable or even better accuracy rates than benchmark methods in most cases.

Table 3: The accuracy rates (%) obtained by each method.

Dataset	MNSWOA-IPM	SFS	SBS	WOA-CM	NSGAII-IPM	NSPSOFS	CMDPSOFS
LATEX	81.14 ± 7.23	78.62 ± 6.72 ↑↑	76.75 ± 8.72 ↑↑	77.38 ± 8.83 ↑	79.27 ± 7.26	78.91 ± 7.96	77.78 ± 7.72
ADPN	82.72 ± 7.90	77.32 ± 13.22 ↑↑	75.71 ± 14.36 ↑↑	77.80 ± 8.97 ↑↑	78.65 ± 14.74	75.42 ± 12.62 ↑↑	80.17 ± 14.65
SPIRA	81.22 ± 8.59	82.81 ± 5.38	73.38 ± 11.53 ↑↑	72.86 ± 11.80 ↑↑	74.93 ± 8.14 ↑↑	74.73 ± 11.46 ↑↑	77.49 ± 8.81 ↑↑
PAPER	87.94 ± 4.61	88.00 ± 4.13	90.07 ± 4.72 ↓	89.73 ± 4.67 ↓	88.76 ± 4.73	88.43 ± 4.69	88.59 ± 4.65
AVERAGE	83.25	81.69	78.98	79.44	80.40	79.37	81.01

6.2. Comparisons on obtained TPR and TNR values

According to Table 4, the proposed MNSWOA-IPM obtains significantly better TPR results in most cases. On LATEX and ADPN, MNSWOA-IPM obtains significantly higher TPR values than five benchmark methods at the significance level of 0.05 and obtains significantly higher TPR values than the other one benchmark method at the significance level of 0.1. On SPIRA and PAPER, MNSWOA-IPM obtains significantly higher TPR values than the five benchmark methods, i.e., SBS, WOA-CM, NSGAII-IPM, NSPSOFS and CMDPSOFS, at the significance level of 0.05. According to Table 5, MNSWOA-IPM obtains similar TNR results to the benchmark methods in most cases. In a few cases, MNSWOA-IPM obtains significantly better or worse TNR results than the benchmark methods. Specifically, MNSWOA-IPM obtains a significantly higher TNR value than SBS, WOA-CM and NSGAII-IPM on SPIRA, obtains a significantly lower TNR value than SFS on LATEX, and obtains a significantly lower TNR value than benchmark methods except SFS on PAPER. Comprehensively analyzing the TPR and TNR results in Tables 4 and 5, MNSWOA-IPM substantially increases the TPR values with a slight decrease of TNR values compared with the benchmark methods. Moreover, according to the average results over the datasets, MNSWOA-IPM obtains a substantially higher average TPR value than the benchmark methods. In comparison, MNSWOA-IPM obtains a slightly lower average TNR value (84.84%) than the highest TNR value (87.04%) obtained by CMDPSOFS.

6.3. Comparisons on obtained AUC values

According to Table 6, MNSWOA-IPM obtains the highest AUC value on LATEX (88.68%), ADPN (87.50%) and SPIRA (84.71%), and obtains a similar AUC value (93.03%) to the highest one (93.21%) obtained by SFS on PAPER. Moreover, MNSWOA-IPM obtains the highest average AUC value (88.48%)

Table 4: The TPR values (%) obtained by each method.

Dataset	MNSWOA-IPM	SFS	SBS	WOA-CM	NSGAII-IPM	NSPSOFS	CMDPSOFS
LATEX	74.05 ± 20.17	49.82 ± 17.79 ↑↑	63.93 ± 15.23 ↑↑	64.64 ± 18.52 ↑↑	64.17 ± 22.11 ↑	59.11 ± 20.17 ↑↑	57.83 ± 22.80 ↑↑
ADPN	83.33 ± 23.57	75.00 ± 25.00 ↑	65.00 ± 32.02 ↑↑	68.33 ± 32.87 ↑↑	69.36 ± 31.16 ↑↑	55.83 ± 39.10 ↑↑	64.03 ± 36.59 ↑↑
SPIRA	74.00 ± 13.81	74.00 ± 15.62	62.00 ± 24.41 ↑↑	60.67 ± 21.59 ↑↑	64.39 ± 19.20 ↑↑	57.00 ± 25.19 ↑↑	63.17 ± 18.73 ↑↑
PAPER	90.56 ± 15.02	88.33 ± 14.53	79.17 ± 23.35 ↑↑	72.78 ± 23.27 ↑↑	65.42 ± 25.85 ↑↑	69.03 ± 24.48 ↑↑	71.13 ± 23.89 ↑↑
AVERAGE	80.48	71.79	67.52	66.61	65.83	60.24	64.04

Table 5: The TNR values (%) obtained by each method.

Dataset	MNSWOA-IPM	SFS	SBS	WOA-CM	NSGAII-IPM	NSPSOFS	CMDPSOFS
LATEX	84.16 ± 8.78	90.70 ± 5.63 ↓↓	82.14 ± 9.36	82.68 ± 9.58	85.46 ± 8.38	87.16 ± 8.28	86.14 ± 7.42
ADPN	82.33 ± 13.03	78.00 ± 20.88	80.00 ± 12.65	81.33 ± 11.47	82.26 ± 15.60	83.00 ± 16.70	86.54 ± 15.42
SPIRA	85.11 ± 13.92	87.56 ± 8.90	79.22 ± 11.32 ↑↑	79.19 ± 12.33 ↑↑	80.45 ± 13.85 ↑	84.11 ± 10.37	85.11 ± 12.12
PAPER	87.74 ± 5.07	88.04 ± 4.90	91.16 ± 5.64 ↓↓	91.45 ± 5.38 ↓↓	91.00 ± 4.47 ↓↓	90.30 ± 5.24 ↓↓	90.36 ± 5.28 ↓↓
AVERAGE	84.84	86.07	83.13	83.66	84.79	86.15	87.04

over the datasets, while the average AUC values obtained by other methods are all below 85%. According to the statistical significance test results, MNSWOA-IPM obtains significantly higher AUC values than benchmark methods in 11/24 cases at the significance level of 0.05, and obtains significantly higher AUC values than the benchmark methods in 5/24 cases at the significance level of 0.1. To sum up, the proposed method obtains a high level of AUC results, which shows that it obtains good classification performance on the unbalanced production datasets.

Table 6: The AUC (%) values obtained by each method.

Dataset	MNSWOA-IPM	SFS	SBS	WOA-CM	NSGAII-IPM	NSPSOFS	CMDPSOFS
LATEX	88.68 ± 5.24	82.58 ± 12.66	81.44 ± 5.94 ↑↑	82.27 ± 6.64 ↑↑	86.58 ± 6.24	81.89 ± 9.93 ↑↑	84.08 ± 7.25 ↑↑
ADPN	87.50 ± 13.71	80.17 ± 16.51 ↑	77.00 ± 23.22 ↑↑	83.83 ± 15.21	82.64 ± 22.26	76.17 ± 21.84 ↑↑	80.14 ± 25.89 ↑
SPIRA	84.71 ± 8.81	82.13 ± 11.34 ↑	82.29 ± 14.08	80.84 ± 11.95 ↑	78.77 ± 9.47 ↑↑	80.89 ± 11.11 ↑	82.12 ± 9.58 ↑↑
PAPER	93.03 ± 8.49	93.21 ± 5.32	91.81 ± 6.73	90.76 ± 8.57	87.07 ± 10.33 ↑↑	90.46 ± 7.61 ↑↑	89.68 ± 8.38 ↑↑
AVERAGE	88.48	84.52	83.14	84.43	83.77	82.35	84.01

6.4. Comparisons on the number of selected features

According to Table 7, MNSWOA-IPM selects the fewest features (i.e., KQCs) on all the datasets. On LATEX, ADPN and PAPER, the statistical significance test results show that MNSWOA-IPM selects significantly fewer features than all the benchmark methods at the significance level of 0.05. On SPIRA, MNSWOA-IPM selects significantly fewer features than three benchmark methods (SBS, WOA-CM and NSPSOFS) at the significance level of 0.05, and selects significantly fewer features than NSGAII-IPM at the significance level of 0.1. SFS, NSGAII-IPM, NSPSOFS and CMDPSOFS select slightly more features than MNSWOA-IPM on the four datasets, and the results of "number of selected features" of these methods are similar. SBS and WOA-CM obtain the worst "number of selected features" results since they select many more features than other methods. The above results imply that the proposed MNSWOA-IPM is very effective in reducing the number of selected features.

Table 7: The number of selected features obtained by each method.

Dataset	MNSWOA-IPM	SFS	SBS	WOA-CM	NSGAII-IPM	NSPSOFS	CMDPSOFS
LATEX	3.80 ± 0.65	7.70 ± 2.90 ↑↑	93.80 ± 15.85 ↑↑	59.17 ± 9.83 ↑↑	6.20 ± 2.26 ↑↑	5.97 ± 3.48 ↑↑	7.30 ± 4.20 ↑↑
ADPN	2.20 ± 0.40	5.30 ± 1.27 ↑↑	25.80 ± 9.87 ↑↑	39.17 ± 7.63 ↑↑	3.10 ± 0.75 ↑↑	5.67 ± 3.40 ↑↑	5.03 ± 2.81 ↑↑
SPIRA	3.40 ± 0.71	3.50 ± 1.02	51.70 ± 16.51 ↑↑	42.40 ± 5.25 ↑↑	3.97 ± 1.14 ↑	4.80 ± 1.68 ↑↑	4.30 ± 2.62
PAPER	3.13 ± 0.67	4.40 ± 2.15 ↑↑	38.90 ± 8.79 ↑↑	26.63 ± 3.96 ↑↑	4.93 ± 0.77 ↑↑	7.87 ± 3.11 ↑↑	5.30 ± 1.37 ↑↑
AVERAGE	3.13	5.22	52.55	41.84	4.55	6.08	5.48

6.5. Computational time

According to Table 8, MNSWOA-IPM takes less computational time than SBS and takes more computational time than SFS on the four datasets. SFS is time efficient due to its greedy search strategy needs only a small number of function evaluations. Additionally, the objective function evaluation process in SFS is time efficient, since SFS searches for the best solution from an empty feature subset, which makes the sizes of feature subsets to be evaluated in SFS are generally small. Compared with the EC-based benchmark methods (WOA-CM, NSGAII-IPM, NSPSOFS and CMDPSOFS), MNSWOA-IPM takes significantly less computational time on all the datasets. This denotes that the caching strategy used in MNSWOA-IPM is effective.

Table 8: Computational time (CPU time in minutes) taken by each method.

Dataset	MNSWOA-IPM	SFS	SBS	WOA-CM	NSGAII-IPM	NSPSOFS	CMDPSOFS
LATEX	6.93 ± 0.84	1.47 ± 0.75 ↓↓	40.01 ± 18.02 ↑↑	25.64 ± 3.68 ↑↑	10.26 ± 0.60 ↑↑	10.08 ± 1.90 ↑↑	10.00 ± 1.74 ↑↑
ADPN	1.14 ± 0.13	0.21 ± 0.05 ↓↓	7.47 ± 0.94 ↑↑	2.97 ± 0.35 ↑↑	1.61 ± 0.06 ↑↑	1.70 ± 0.20 ↑↑	1.57 ± 0.19 ↑↑
SPIRA	2.85 ± 0.40	0.23 ± 0.07 ↓↓	17.07 ± 2.40 ↑↑	9.21 ± 0.77 ↑↑	4.10 ± 0.19 ↑↑	3.84 ± 0.59 ↑↑	3.67 ± 0.63 ↑↑
PAPER	4.99 ± 0.81	0.26 ± 0.16 ↓↓	6.17 ± 2.31 ↑↑	15.95 ± 1.57 ↑↑	7.27 ± 0.28 ↑↑	8.91 ± 1.25 ↑↑	8.29 ± 0.72 ↑↑
AVERAGE	3.98	0.54	17.68	13.44	5.81	6.13	5.88

6.6. Discussion

Comprehensively analyzing the results in Tables 3 to 8, we can reach the following conclusions:

First, the proposed MNSWOA-IPM performs more effectively than benchmark methods in selecting KQCs on the unbalanced production data. Specifically, MNSWOA-IPM generally obtains high values of accuracy and AUC, while obtaining substantially higher TPR values with similar or slightly lower TNR values than benchmark methods. This denotes that MNSWOA-IPM can obtain good classification performance on both the majority and minority classes. The reason that leads to the results is that the GM measure used in MNSWOA-IPM can measure the classification ability of feature subsets more precisely than the accuracy measure used in benchmark methods on the unbalanced production data.

Second, the multiobjective EC approaches have a better feature reduction capability than the single objective EC approach, i.e., WOA-CM. The multiobjective approaches, including NSGAII-IPM, NSPSOFS and CMDPSOFS, adopt the same optimization objectives (maximizing the accuracy and minimizing the feature subset size) as that of WOA-CM. Compared with WOA-CM, these multiobjective approaches select much fewer features while obtaining similar classification performance. This is due to the fact that the multiobjective approaches independently treat “the feature subset size” as an optimization objective in addition to the classification performance measure, which promotes the algorithm to select only a few features. In comparison, “the percentage of selected features” in WOA-CM gains a much lower weight than “the classification accuracy” to form the integrated objective function, which limits its ability to reduce irrelevant or redundant features. Moreover, from the perspective of optimization, the swarm (or population) diversity in the multiobjective approaches can be well maintained. This is because that the swarm of these multiobjective approaches during the evolutionary process (even in the late evolutionary phases) is composed of solutions with different feature subset sizes as the feature subset size is defined as one separate optimization objective. This is a clear advantage compared with single objective approaches for FS.

Third, the multiobjective EC approaches are much more time efficient than the single objective EC approach, i.e., WOA-CM. As stated above, the multiobjective approaches have a better feature reduction capability than WOA-CM. This implies that the average feature subset size of the multiobjective approaches for objective function evaluation during the whole evolutionary process is smaller than that of WOA-CM. Since the size of a feature subset (solution) indicates the dimensionality of the data used to build the evaluator (the classifier) in the wrapper framework, it is obvious that the multiobjective approaches with a better feature reduction capability take less time in evaluating the objective functions than WOA-CM.

Finally, it should be noted that the multiobjective approaches actually adopt a two-phase FS scheme to obtain the final feature subset (KQC set). In the first phase, a multiobjective optimization method is used to find a set of solutions on the nondominated front by simultaneously optimizing the two FS objectives, the classification performance and the feature subset size. In the second phase, the multiobjective decision method (IPM) is used to selected the most desirable feature subset from the found solutions in the first phase. It is worth noting that, it is flexible to perform the second phase FS task. The decision maker may use other multiobjective decision methods such as TOPSIS instead of IPM to select a desirable final feature subset. Moreover, the decision maker’s preference can be considered while selecting the final feature subset. The tradeoffs between the two FS objectives can be considered by the decision maker based on the domain knowledge for selecting the final feature subset. Comparing the performance of different multiobjective decision methods and examining the FS performance considering different types of decision makers’ preferences are worth studying in the future.

6.7. Comparisons on the identified KQCs

The above experimental results based on 10-fold CV have shown that the proposed MNSWOA-IPM method is an effective and efficient FS method for KQC identification. From a practical point of view, the practitioners are concerned about which QCs (feature) are identified as the KQCs. In this regard, we further compare the identified KQCs of the FS methods on the four datasets. To obtain the KQCs, the four original datasets (LATEX, ADPN, SPIRA and PAPER) are input to the 7 FS methods (including

MNSWOA-IPM and the benchmark methods). The identified KQCs of each FS method are shown in Table 9, where each identified KQC is denoted by its index in the full QC set. According to the table, MNSWOA-IPM generally selects a few KQCs on the four datasets; SFS, NSGAII-IPM, NSPSOFS, and CMDPSOFS generally select similar or slightly more KQCs than MNSWOA-IPM on the four datasets; SBS and WOA-CM select substantially more KQCs than other methods. These results are consistent with the experimental results in Section 6.4, and show that MNSWOA-IPM is effective in reducing the number of selected features (QCs). Specifically, on LATEX, MNSWOA-IPM identifies QCs 27, 53, 57, 96 and 97 as the KQCs. These QCs are also identified by SBS and WOA-CM. However, SBS and WOA-CM also select a large number of other QCs. SFS and NSGAII-IPM also identify QCs 53, 96 and 97 as the KQCs, while they do not identify QC 27 as the KQC. On ADPN, MNSWOA-IPM identifies QCs 8, 24 and 68 as the KQCs. WOA-CM identifies the three QCs as well, while it identifies many more QCs as KQCs than MNSWOA-IPM. Except for MNSWOA-IPM and WOA-CM, QCs 24 and 68 are also identified by SFS and NSPSOFS, and QC 8 is only identified by SBS. On SPIRA, MNSWOA-IPM identifies QCs 71, 77 and 78 as the KQCs. Except for MNSWOA-IPM, QC 71 is also identified all the other methods, QC 77 is also identified by SBS, WOA-CM and NSGAII-IPM, and QC 78 is also identified by SFS, SBS and CMDPSOFS. On PAPER, MNSWOA-IPM identifies QCs 25, 33 and 45 as the KQCs. Except for MNSWOA-IPM, QC 33 is also identified by SBS, WOA-CM and CMDPSOFS, and QC 45 is also identified by the benchmark methods except SFS. However, no methods except MNSWOA-IPM can identify QC 25 as the KQC. A possible reason is that the highly unbalanced PAPER dataset strongly affects the KQC identification performance of the benchmark methods. In these methods, some informative features (QCs) are not able to be identified because the data imbalance problem is not considered during the FS process.

7. Further analysis

The results in Section 6 have shown that the proposed MNSWOA-IPM is effective for FS (KQC identification) on unbalanced production data. One main reason why MNSWOA-IPM obtains better performance than benchmark methods is that it considers the imbalance problem of production data while constructing the FS problem. Except for the established FS problem, the optimization method plays an important role in FS as well. Specifically, the performance of MNSWOA in the first phase of MNSWOA-IPM determines whether high quality candidate solutions can be found for further selection by IPM. In this section, we further compare MNSWOA with several existing multiobjective optimization methods, by applying these methods to the same FS problem (defined in Eq. (9)) as optimized by MNSWOA. The fitness values (i.e., objective functions f_1 and f_2) are used to compare the search performance of these methods.

Table 9: The KQCs identified by each method.

Method	LATEX	ADPN	SPIRA	PAPER
MNSWOA-IPM	27, 53, 57, 96, 97	8, 24, 68	71, 77, 78	25, 33, 45
SFS	37, 47, 53, 55, 96, 97, 115	24, 25, 45, 68, 94, 96	23, 71, 78	34, 47
SBS	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 20, 21, 22, 23, 24, 26, 27, 29, 30, 31, 32, 33, 34, 39, 40, 42, 43, 44, 45, 46, 47, 51, 52, 53, 54, 56, 57, 58, 60, 61, 62, 64, 65, 66, 68, 69, 70, 72, 73, 74, 76, 77, 78, 79, 80, 83, 84, 85, 86, 87, 95, 96, 97, 98, 103, 109, 117	1, 2, 7, 8, 13, 14, 15, 18, 21, 23, 24, 29, 33, 35, 36, 43, 45, 46	1, 4, 5, 6, 8, 9, 10, 11, 13, 14, 15, 16, 22, 25, 26, 27, 28, 32, 35, 36, 39, 40, 41, 42, 43, 48, 49, 50, 51, 56, 58, 60, 61, 63, 67, 70, 71, 75, 77, 78, 80, 83, 84, 86, 89, 90, 92, 93, 96	2, 3, 5, 6, 7, 9, 10, 13, 20, 21, 22, 23, 24, 27, 29, 30, 31, 33, 34, 35, 36, 37, 38, 39, 40, 41, 43, 45, 46, 48, 49, 50, 51, 52, 53
WOA-CM	2, 3, 7, 10, 13, 15, 23, 24, 25, 26, 27, 38, 40, 45, 46, 51, 53, 55, 57, 59, 62, 63, 64, 66, 69, 71, 72, 73, 75, 76, 77, 80, 81, 83, 84, 85, 87, 91, 92, 93, 94, 95, 96, 97, 98, 100, 102, 104, 114	1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 14, 15, 19, 24, 25, 28, 29, 30, 32, 35, 36, 38, 39, 40, 43, 46, 48, 51, 55, 58, 60, 62, 64, 68, 69, 71, 72, 74, 76, 78, 82, 86, 87, 88, 89, 90, 93, 95, 96, 97, 98, 99, 100	1, 6, 14, 27, 28, 30, 31, 32, 35, 37, 40, 42, 46, 55, 56, 59, 61, 67, 69, 71, 77, 79, 81, 83, 84, 86, 90, 96	4, 5, 6, 7, 11, 13, 15, 18, 20, 23, 24, 26, 28, 29, 30, 32, 33, 34, 36, 37, 38, 40, 41, 44, 45, 46, 48, 53
NSGAII-IPM	23, 51, 53, 69, 78, 84, 96, 97	36, 71	26, 32, 71, 77, 96	5, 7, 23, 37, 45, 47, 53
NSPSOFS	23, 27, 50, 51, 116	24, 68, 71, 79, 80, 92, 96	32, 52, 71	4, 7, 23, 28, 29, 32, 37, 40, 41, 45, 53
CMDPSOFS	23, 27, 51, 74, 81, 96, 102	25, 68	63, 71, 78, 95	7, 18, 33, 34, 37, 42, 45

7.1. Experimental design

Several benchmark multiobjective optimization methods are used to evaluate the search performance of MNSWOA. First, the three multiobjective optimization methods, MNSGAII (Li et al., 2016), NSPSO (Xue et al., 2013) and CMDPSO (Xue et al., 2013), for establishing the benchmark FS methods in Section 5 are used. Second, two widely used MOEAs, i.e., SPEA2 (Zitzler et al., 2001) and MOEA/D (Zhang and Li, 2007) with the Tchebycheff approach, are adopted as the benchmark methods as well. Finally, to test whether the unique components in MNSWOA are effective, we design three MNSWOA variants named MNSWOA-NS, MNSWOA-UNI and MNSWOA-MU as the benchmark methods. Compared with MNSWOA, the differences in the three variants are: MNSWOA-NS adopts the standard fast nondominated sorting approach and the crowding distance calculation method proposed by Deb et al. (2002) to sort solutions; MNSWOA-UNI randomly selects the reference solution instead of the uniform selection strategy; MNSWOA-MU does not conduct the mutation operations. All the multiobjective optimization methods are used on the same FS problem in Eq. (9), and the experimental configurations are the same as those in Section 5.4, i.e., 3 repetitions of 10-fold CV (30 experimental runs in total) are conducted. The parameter settings for MNSGAII, NSPSOFS and CMDPSOFS are the same as those in Section 5.3. The three MNSWOA variants adopt the same parameters as those used in MNSWOA. In SPEA2 and MOEA/D, the size of population, the maximum number of iterations, the crossover rate and the mutation rate are 100, 100, 0.9 and $1/D$ (D is the number of original features), which are the same as those used in MNSGAII. The number of neighbors in MOEA/D is set as 10 as suggested by Zhang and Li (2007). Similar to the comparisons in Section 6, "↑↑" or "↑" ("↓↓" or "↓") denotes MNSWOA obtains significantly better (worse) results than the compared method at a significance level of 0.05 or 0.1 using the Wilcoxon signed-rank test.

7.2. Quality indicators

We conduct two sets of comparisons to validate the MNSWOA method. First, three widely used quality indicators are adopted to compare the final search results of the multiobjective optimization methods. They are the inverted generational distance (IGD) (Deb and Jain, 2014; Yuan et al., 2016), the hypervolume (HV) measure (Yuan et al., 2016; Zitzler and Thiele, 1999) and the set coverage (SC) measure (Zhang and Li, 2007; Zitzler and Thiele, 1999). Second, the convergence distance (CD) indicator (Li et al., 2019) that measures the quality of solutions at each iteration is adopted to draw the convergence curves of the multiobjective optimization methods. A brief introduction of these indicators is presented as follows.

The IGD indicator is defined as follows. For a set $\mathbb{P}$ of solutions, its IGD value $IGD(\mathbb{P})$ is the average of distances between Pareto optimal solutions and their nearest solution in $\mathbb{P}$. Let $\mathbb{P}^*$ be the Pareto set. Then, $IGD(\mathbb{P})$ is obtained as

$$IGD(\mathbb{P}) = \frac{1}{|\mathbb{P}^*|} \sum_{p^* \in \mathbb{P}^*} \min_{p \in \mathbb{P}} D(p, p^*), \tag{14}$$

where $|\ |$ denotes the set size and $D(p, p^*)$ denotes the Euclidean distance between the two solutions p and p^* in the objective space. The smaller the $IGD(\mathbb{P})$ is, the higher the quality of $\mathbb{P}$ is.

The HV indicator is defined as follows. For a set $\mathbb{P}$ of solutions, $HV(\mathbb{P})$ denotes the hypervolume dominated by $\mathbb{P}$. Let $\mathbf{r} = (r_1, ..., r_M)$ be the reference point in the objective space. Then, $HV(\mathbb{P})$ is obtained as

$$HV(\mathbb{P}) = volume \left(\bigcup_{p \in \mathbb{P}} \prod_{m=1}^{M} |r_m - f_{p,m}| \right), \tag{15}$$

where M denotes the number of objectives, $f_{p,m}$ denotes the mth objective function value of solution p. The larger the $HV(\mathbb{P})$ is, the better the $\mathbb{P}$ is.

The SC indicator is defined as follows. Let $\mathbb{P}$ and $\mathbb{Q}$ be two sets of solutions. $SC(\mathbb{P}, \mathbb{Q})$ indicates the percentage of solutions in $\mathbb{Q}$ that are covered by (dominated by or equal to) solutions in $\mathbb{P}$. It is obtained as

$$SC(\mathbb{P}, \mathbb{Q}) = \frac{|\{q \in \mathbb{Q} | \exists p \in \mathbb{P} : p \preceq q\}|}{|\mathbb{Q}|}, \tag{16}$$

where $p \preceq q$ denotes p dominates or equals q in the objective space. A larger $SC(\mathbb{P}, \mathbb{Q})$ denotes a higher quality of $\mathbb{P}$.

Since the ranges of different objective functions can be very different, it is required to solve the incommensurability problem. In this paper, the objective functions f_1 and f_2 are normalized to f_1^N and f_2^N with the min-max normalization method, where the maximum or minimum function value is obtained from the union of returned solutions by all the compared methods. The normalized function values f_1^N and f_2^N are adopted instead of f_1 and f_2 when calculating the IGD, HV and SC values for each method. The IGD indicator requires a known true Pareto set, which is unknown for most real world FS problems, including the KQC identification problem addressed in this paper. Therefore, the best solutions of the 30 experimental runs obtained by all the multiobjective optimization methods are combined to form the Pareto set. For the HV indicator, the reference point is defined as $\mathbf{r} = (1.1, 1.1)$ as suggested by Yuan et al. (2016). For the SC indicator, we let $\mathbb{Q}$ be the Pareto set. Thus, the SC value indicates the percentage of Pareto solutions covered by $\mathbb{P}$.

The CD indicator (Li et al., 2019) measures the "distance" between the Pareto solutions and the solutions found by an optimization method at each iteration. Let $\mathbb{P}^*$ be the Pareto set and $\mathbb{P}^t$ be the set of found solutions at the tth iteration. Then, the CD value for $\mathbb{P}^t$ is obtained by

$$CD(\mathbb{P}^t) = 0.5 * \left(\frac{1}{|\mathbb{P}^*|} \sum_{p^* \in \mathbb{P}^*} \min_{p \in \mathbb{P}^t} D(p, p^*) + \frac{1}{|\mathbb{P}^t|} \sum_{p \in \mathbb{P}^t} \min_{p^* \in \mathbb{P}^*} D(p^*, p) \right), \tag{17}$$

where $D(p, p^*)$ denotes the Euclidean distance between the two solutions p and p^*. According to Eq. (17), the first part inside the big parentheses is actually the definition of IGD. The second part is the average of distances between solutions in $\mathbb{P}^t$ and the nearest Pareto solution, which is the definition of the generational distance (GD) indicator. Therefore, the CD indicator can been as an integration of IGD and GD. To solve the incommensurability problem when calculating the CD indicator, the two objective functions are normalized with the min-max normalization method as well. Since the CD value of the found solutions at each iteration can be calculated by Eq. (17), the convergence curve can be drawn for each method on each dataset.

7.3. Comparisons of search results of the multiobjective optimization methods

Table 10 shows the results on the quality indicators obtained by MNSWOA and the benchmark methods (except for the MNSWOA variants). In the table, the means and standard deviations of the IGD, HV and SC values over the 30 runs are shown for each method and the "AVERAGE" row shows the average results over the four datasets. According to the IGD results, MNSWOA obtains lower IGD values than the benchmark methods. The statistical significance test results show that MNSWOA can obtain significantly better IGD results in most cases. Specifically, at the significance level of 0.05, MNSWOA performs significantly better

Table 10: Comparisons of search results between MNSWOA and benchmark methods.

Indicator	Dataset	MNSWOA	MNSGAII	SPEA2	MOEA/D	NSPSO	CMDPSO
IGD	LATEX	0.019 ± 0.018	$0.046 \pm 0.018 \uparrow\uparrow$	$0.042 \pm 0.018 \uparrow\uparrow$	$0.076 \pm 0.048 \uparrow\uparrow$	$0.081 \pm 0.037 \uparrow\uparrow$	$0.086 \pm 0.056 \uparrow\uparrow$
	ADPN	0.025 ± 0.029	$0.049 \pm 0.034 \uparrow\uparrow$	$0.036 \pm 0.024 \uparrow\uparrow$	$0.051 \pm 0.031 \uparrow\uparrow$	$0.062 \pm 0.018 \uparrow\uparrow$	$0.082 \pm 0.050 \uparrow\uparrow$
	SPIRA	0.032 ± 0.023	0.037 ± 0.014	$0.040 \pm 0.018 \uparrow\uparrow$	$0.090 \pm 0.035 \uparrow\uparrow$	$0.080 \pm 0.049 \uparrow\uparrow$	$0.084 \pm 0.046 \uparrow\uparrow$
	PAPER	0.024 ± 0.027	0.018 ± 0.015	0.026 ± 0.017	$0.080 \pm 0.048 \uparrow\uparrow$	$0.042 \pm 0.017 \uparrow\uparrow$	$0.033 \pm 0.018 \uparrow\uparrow$
	AVERAGE	0.025	0.038	0.036	0.074	0.066	0.071
HV	LATEX	1.168 ± 0.018	$1.083 \pm 0.062 \uparrow\uparrow$	$1.086 \pm 0.063 \uparrow\uparrow$	$1.112 \pm 0.032 \uparrow\uparrow$	$1.059 \pm 0.070 \uparrow\uparrow$	$1.010 \pm 0.120 \uparrow\uparrow$
	ADPN	1.172 ± 0.032	$1.130 \pm 0.051 \uparrow\uparrow$	$1.163 \pm 0.034 \uparrow\uparrow$	$1.136 \pm 0.040 \uparrow\uparrow$	$1.105 \pm 0.069 \uparrow\uparrow$	$1.062 \pm 0.106 \uparrow\uparrow$
	SPIRA	1.120 ± 0.042	$1.089 \pm 0.042 \uparrow\uparrow$	$1.100 \pm 0.038 \uparrow\uparrow$	$1.050 \pm 0.059 \uparrow\uparrow$	$1.002 \pm 0.113 \uparrow\uparrow$	$0.989 \pm 0.110 \uparrow\uparrow$
	PAPER	1.119 ± 0.039	1.119 ± 0.038	$1.112 \pm 0.044 \uparrow\uparrow$	$1.018 \pm 0.073 \uparrow\uparrow$	$1.017 \pm 0.081 \uparrow\uparrow$	$1.070 \pm 0.060 \uparrow\uparrow$
	AVERAGE	1.145	1.106	1.115	1.079	1.046	1.033
SC	LATEX	0.406 ± 0.179	$0.057 \pm 0.132 \uparrow\uparrow$	$0.049 \pm 0.114 \uparrow\uparrow$	$0.145 \pm 0.106 \uparrow\uparrow$	$0.030 \pm 0.063 \uparrow\uparrow$	$0.025 \pm 0.060 \uparrow\uparrow$
	ADPN	0.450 ± 0.166	$0.126 \pm 0.132 \uparrow\uparrow$	$0.183 \pm 0.193 \uparrow\uparrow$	$0.156 \pm 0.112 \uparrow\uparrow$	$0.039 \pm 0.070 \uparrow\uparrow$	$0.038 \pm 0.088 \uparrow\uparrow$
	SPIRA	0.404 ± 0.147	$0.096 \pm 0.138 \uparrow\uparrow$	$0.131 \pm 0.138 \uparrow\uparrow$	$0.135 \pm 0.124 \uparrow\uparrow$	$0.035 \pm 0.069 \uparrow\uparrow$	$0.029 \pm 0.080 \uparrow\uparrow$
	PAPER	0.415 ± 0.158	$0.351 \pm 0.178 \uparrow$	$0.317 \pm 0.166 \uparrow$	$0.187 \pm 0.104 \uparrow\uparrow$	$0.051 \pm 0.073 \uparrow\uparrow$	$0.117 \pm 0.113 \uparrow\uparrow$
	AVERAGE	0.419	0.157	0.170	0.156	0.039	0.052

than all the benchmark methods on LATEX and ADPN, performs significantly better than 4 of the 5 benchmark methods on SPIRA, and performs significantly better than 3 of the 5 benchmark methods on PAPER. According to HV results, MNSWOA performs significantly better than the benchmark methods in almost all cases. The statistical significance test results show that MNSWOA obtains significantly better (higher) HV results in 19 of the 20 cases compared with the benchmark methods and only obtains a similar HV value to that of MNSGAII on PAPER. According to SC results, MNSWOA performs significantly better than the benchmark methods at the significance level of 0.05 in almost all cases. The only exception is that MNSWOA obtains a significantly better SC value than MNSGAII and SPEA2 at the significance level of 0.1 on PAPER. To sum up, the results on the IGD, HV, and SC indicators show a better search ability of MNSWOA than the benchmark multiobjective optimization methods.

Table 11 shows the results on the quality indicators of MNSWOA and its variants (i.e., MNSWOA-NS, MNSWOA-UNI and MNSWOA-MU). MNSWOA obtains significantly better IGD, HV and SC results than MNSWOA-NS and MNSWOA-MU at the significance level of 0.05 in almost all cases. The only exception is that MNSWOA obtains a significantly better IGD value than MNSWOA-MU at the significance level of 0.1 on PAPER. This implies that the modified nondominated sorting approach and the mutation operations are effective for improving the search performance of MNSWOA. Comparing MNSWOA with MNSWOA-UNI, MNSWOA obtains significantly better IGD and HV results on LATEX and SPIRA, and obtains significantly better SC results on SPIRA and PAPER. This finding shows that at least one quality indicator indicates that MNSWOA obtains significantly better search performance than MNSWWOA-UNI on the datasets except ADPN. According to above results, we can conclude that MNSWOA obtains better search performance than its variants. Moreover, the modified nondominated sorting approach, the uniform selection strategy and the mutation operations proposed for MNSWOA are all effective for improving its search performance.

Table 11: Comparisons of search results between MNSWOA and its variants.

Indicator	Dataset	MNSWOA	MNSWOA-NS	MNSWOA-UNI	MNSWOA-MU
IGD	LATEX	0.019 ± 0.018	$0.082 \pm 0.039 \uparrow\uparrow$	$0.031 \pm 0.027 \uparrow\uparrow$	$0.059 \pm 0.042 \uparrow\uparrow$
	ADPN	0.025 ± 0.029	$0.050 \pm 0.040 \uparrow\uparrow$	0.029 ± 0.032	$0.035 \pm 0.022 \uparrow\uparrow$
	SPIRA	0.032 ± 0.023	$0.101 \pm 0.043 \uparrow\uparrow$	$0.045 \pm 0.033 \uparrow$	$0.068 \pm 0.046 \uparrow\uparrow$
	PAPER	0.024 ± 0.027	$0.068 \pm 0.046 \uparrow\uparrow$	0.026 ± 0.025	$0.033 \pm 0.023 \uparrow$
	AVERAGE	0.025	0.076	0.032	0.049
HV	LATEX	1.168 ± 0.018	$1.105 \pm 0.037 \uparrow\uparrow$	$1.162 \pm 0.023 \uparrow$	$1.117 \pm 0.035 \uparrow\uparrow$
	ADPN	1.172 ± 0.032	$1.144 \pm 0.041 \uparrow\uparrow$	1.171 ± 0.034	$1.154 \pm 0.041 \uparrow\uparrow$
	SPIRA	1.120 ± 0.042	$1.042 \pm 0.069 \uparrow\uparrow$	$1.108 \pm 0.039 \uparrow\uparrow$	$1.071 \pm 0.053 \uparrow\uparrow$
	PAPER	1.119 ± 0.039	$1.017 \pm 0.082 \uparrow\uparrow$	1.115 ± 0.035	$1.062 \pm 0.070 \uparrow\uparrow$
	AVERAGE	1.145	1.077	1.139	1.101
SC	LATEX	0.406 ± 0.179	$0.257 \pm 0.065 \uparrow\uparrow$	0.369 ± 0.162	$0.172 \pm 0.071 \uparrow\uparrow$
	ADPN	0.450 ± 0.166	$0.283 \pm 0.128 \uparrow\uparrow$	0.392 ± 0.188	$0.184 \pm 0.131 \uparrow\uparrow$
	SPIRA	0.404 ± 0.147	$0.264 \pm 0.099 \uparrow\uparrow$	$0.349 \pm 0.111 \uparrow$	$0.132 \pm 0.095 \uparrow\uparrow$
	PAPER	0.415 ± 0.158	$0.170 \pm 0.090 \uparrow\uparrow$	$0.344 \pm 0.145 \uparrow\uparrow$	$0.139 \pm 0.077 \uparrow\uparrow$
	AVERAGE	0.419	0.243	0.363	0.157

7.4. Convergence property analysis

The above results illustrate that the proposed MNSWOA method shows a good search ability in solving the FS problem addressed in this paper. In this section, we further analyze the convergence property of MNSWOA. The analysis is composed of two parts. First, the convergence curves of MNSWOA are compared with those of the benchmark optimization methods, i.e., MNSGAII, SPEA2, MOEA/D, NSPSO and CMDPSO. Second, the convergence curves of MNSWOA are compared with those of the MNSWOA variants, i.e., MNSWOA-NS, MNSWOA-UNI and MNSWOA-MU.

Comparisons of convergence curves obtained by MNSWOA and the benchmark methods are shown in Figure 2, where x-axis is the number of iterations and y-axis is the distance between the Pareto solutions and the solutions obtained at each iteration measured by the CD indicator in Eq. (17). Several results can be found in Figure 2. First, MNSWOA shows a better convergence property than other methods. On each dataset, the convergence curve of MNSWOA is much lower than those of other methods, showing that MNSWOA can converge fast and obtain better final convergence results. Second, the PSO-based multiobjective approaches, i.e., NSPSO and CMDPSO, converge faster than the two multiobjective GAs, MNSGAII and SPEA2. Meanwhile, the two PSO methods also suffer the premature convergence problem, since in the late phase of the evolutionary process their convergence curves become higher than MNSGAII and SPEA2. This shows that the two PSO methods can converge faster but have worse global search performance than the two GAs. Third, the convergence performance of MOEA/D is between the two PSO methods (NSPSO and CMDPSO) and the two GAs (MNSGAII and SPEA2). In the early phase of the evolutionary process, MOEA/D has a slightly lower convergence speed than NSPSO and CMDPSO. In the late phase of the evolutionary process, MOEA/D can obtain similar convergence curves to those of MNSGAII and SPEA2. To sum up, the proposed MNSWOA obtains both a fast convergence speed and a good final convergence result, showing that it is a very desirable multiobjective optimization method for the

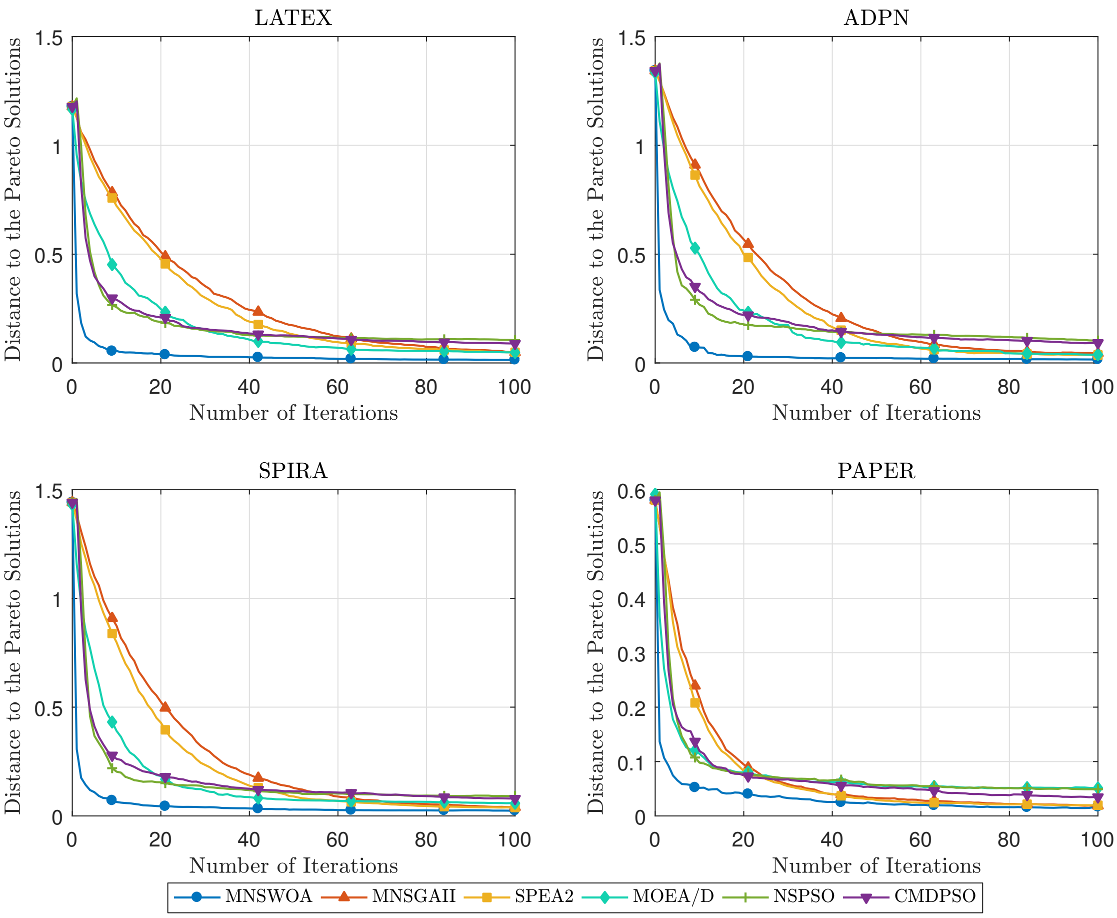

Figure 2: Convergence curves obtained by MNSWOA and benchmark multiobjective optimization methods.

FS problem addressed in this paper.

Comparisons of the convergence curves obtained by MNSWOA and its variants are shown in Figure 3, where the convergence curves at iterations 20 to 100 are magnified to facilitate the comparison. First, it is found that MNSWOA and all the variants can quickly reduce the distance values measured by the CD indicator on each dataset, which means that all these methods can quickly converge. Second, according to the enlarged parts of the convergence curves, MNSWOA obtains lower curves than the variants. This denotes that all the new components used in MNSWOA are effective to improve the convergence performance. Finally, the convergence curves of MNSWOA-NS and MNSWOA-MU are much higher than those of MNSWOA. Meanwhile, the convergence curves of MNSWOA-UNI are slightly higher than those of MNSWOA. This means that the modified fast nondominated sorting approach and the mutation operations are more effective than the uniform selection strategy in improving the search performance. To sum up, the comparison results in Figure 3 denote that the modified fast nondominated sorting approach, the uniform

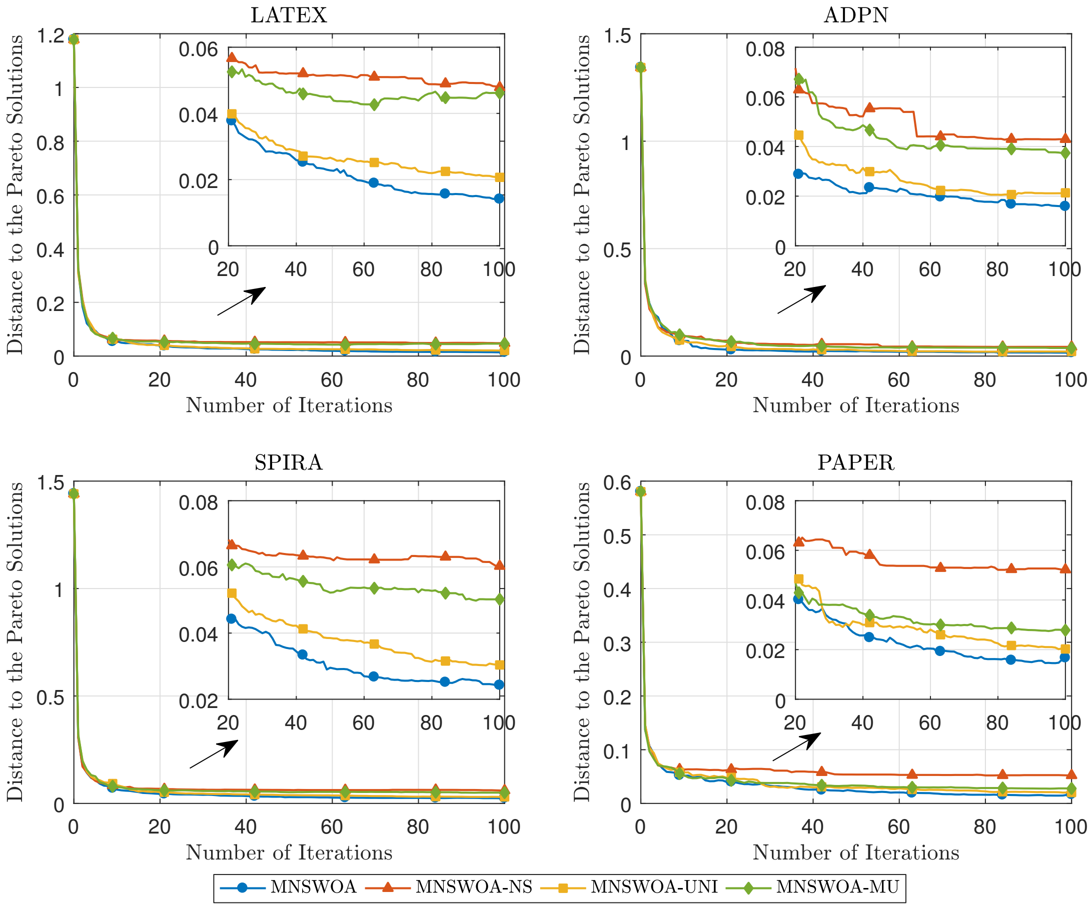

Figure 3: Convergence curves obtained by MNSWOA and its variants.

selection strategy and mutation operations adopted by MNSWOA are all effective for improving the search performance.

The above comparisons have shown that MNSWOA obtains a better convergence property than the benchmark methods based on GA and PSO strategies and the new components used in MNSWOA are effective for improving its search performance. Several reasons can explain the effectiveness of MNSWOA:

- MNSWOA comprehensively combines the three types of "whale optimization algorithm" search strategies during the evolutionary process. These strategies include both the direct learning (encircling prey and search for prey) and spiral learning (spiral position updating) mechanisms as well as both the local search (encircling prey and spiral position updating) and global search (search for prey) mechanisms, which produce good convergence performance of MNSWOA.

- Compared with the traditional fast nondominated sorting approach, the modified sorting approach used in MNSWOA can effectively reduce the probability of duplicate solutions to be added to the next iteration, which substantially improves the swarm diversity and results in better convergence performance.

- The uniform selection strategy ensures similar chances for the candidate solutions to be selected as the reference solution. In comparison, the random reference selection strategy may select some candidates many more times than others. This means that the limited computational resources are allocated too many on some candidate solutions that have already been sufficiently searched around, which has a negative effect on the overall search performance of the MNSWOA variant with the random reference selection strategy.

- The mutation operations enrich the solution updating strategies in MNSWOA and are helpful to make MNSWOA escape from the local optima during the evolutionary process. This can substantially improve the global search ability of MNSWOA, especially for FS problems that have a very large search space.

8. Conclusions

In modern production processes, identifying KQCs that strongly affect the product quality is essential for quality control and improvement. This paper proposes a multiobjective wrapper-based FS method (MNSWOA-IPM) for KQC identification based on the unbalanced production data. The FS problem is defined as maximizing the GM measure and minimizing the feature subset size. The GM measure is used instead of the commonly used accuracy as the feature (QC) importance measure to address the imbalance problem of production data. Then, a two-phase optimization method named MNSWOA-IPM is proposed to solve the defined FS problem. In MNSWOA-IPM, MNSWOA is used to find a set of nondominated solutions, from which IPM is used to select the final solution. MNSWOA inherits the solution updating strategies of WOA to evolve new solutions during the evolutionary process and adopts a modified fast nondominated sorting approach to sort solutions in the multiobjective scenario. Moreover, the uniform selection strategy and mutation operations are adopted in MNSWOA to enhance its search performance.

We conduct two sets of experiments to validate the proposed FS (i.e., KQC identification) method. According to the first set of experiments, MNSWOA-IPM obtains substantially better FS results than the benchmark methods, including two conventional FS methods (SFS and SBS), the single objective WOA based FS method (WOA-CM) and three recently proposed multiobjective FS methods (NSGAII-IPM, NSPSOFS and CMDPSOFS). The proposed MNSWOA-IPM generally obtains better classification performance with fewer features (KQCs) than the benchmark methods. Moreover, MNSWOA-IPM takes less computational time than the benchmark methods except for SFS. According to the second set of experiments, MNSWOA obtains substantially better search performance than the benchmark multiobjective optimization methods, including MNSGAII, SPEA2, MOEA/D, NSPSO and CMDPSO. Meanwhile, the the modified sorting approach, the uniform selection strategy and the mutation operations adopted in MNSWOA are all effective for improving its search performance.

The output variable of production data may be a continuous variable that reflects the product quality instead of the discrete class label addressed in this paper. This requires a FS method for the regression model to perform KQC identification. In the future, we will focus on building a FS method based on MNSWOA for regression models. Moreover, proposing a filter-based feature subset importance measure to build a more time efficient multiobjective FS method is also worth studying.

Acknowledgements

The authors would like to thank the editor and anonymous referees for the constructive comments and suggestions. This work was supported by the Humanities and Social Sciences Youth Fund of Ministry of Education of China [Grant numbers 19YJC630071 and 19YJC630221]; and the National Natural Science Foundation of China (NSFC) [Grant number 71661147003].

Appendix A. Time complexity of the modified nondominated sorting approach

The modified sorting approach in Algorithm 2 contains three time consuming parts, i.e., the traditional fast nondominated sorting process (line 2), the nondomination rank increasing process (lines 4 to 11) and the resorting process (line 14). The time complexity of the traditional fast nondominated sorting process is $O(M_o(N^c)^2)$ (Deb et al., 2002), where M_o denotes the number of objectives and N^c denotes the number of solutions to be sorted. The time complexity of the resorting sorting process is $O((N^c)\log(N^c))$.

To obtain the time complexity of the nondomination rank increasing process, we need to first analyze the duplicate solution detection process (lines 5 to 10) inside the "for" loop. To decide whether $Decode(\mathbf{X}_i)$ is the same as any solution in $Decode(\mathbb{S}_{sorted})$, it is required to compare $Decode(\mathbf{X}_i)$ with every solution in $Decode(\mathbb{S}_{sorted})$. Thus, the time complexity is $O(N^cD)$ since the maximum number of features in $Decode(\mathbb{S}_{sorted})$ is D and the maximum number of solutions in $Decode(\mathbb{S}_{sorted})$ is N^c. However, we can use the information of objective functions to reduce the time complexity of this part. A $Decode(\mathbf{X}_i)$ can be the same as another solution in $Decode(\mathbb{S}_{sorted})$ only if these two solutions have the same objective function values. Therefore, $Decode(\mathbf{X}_i)$ only needs to be compared to solutions in $Decode(\mathbb{S}_{sorted})$ with the same objective function values. $Decode(\mathbb{S}_{sorted})$ is a set without any duplicate solutions. Thus, the number of solutions in $Decode(\mathbb{S}_{sorted})$ that have the same objective function values as $Decode(\mathbf{X}_i)$ is generally a very small number $R << D$ (or N^c). Hence, in common cases, the time complexity of the duplicate solution detection process can be reduced from $O(N^cD)$ to $O(RD) \cong O(D)$. Considering the outer "for" loop in line 4, the time complexity of the nondomination rank increasing process is $O(N^cD)$.

According to the above analysis, the time complexity of the sorting approach is $O(M_o(N^c)^2)+O(N^cD)+O((N^c)\log(N^c))$. Since, in this paper, the number of objectives $M_o = 2$ and $N^c = 2N$, the time complexity of the modified nondominated sorting approach can be transformed to $O(8(N)^2) + O(2ND) + O((2N)\log(2N)) \cong O(N^2) + O(ND)$.

References

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

Amoozegar, M., Minaei-Bidgoli, B., 2018. Optimizing multi-objective PSO based feature selection method using a feature elitism mechanism. Expert Systems with Applications 113, 499 – 514.

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, 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, 97 – 105.

Bhowan, U., Johnston, M., Zhang, M., Yao, X., 2013. Evolving diverse ensembles using genetic programming for classification with unbalanced data. IEEE Transactions on Evolutionary Computation 17, 368–386.

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

Deb, K., Jain, H., 2014. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation 18, 577–601.

Ekbal, A., Saha, S., 2012. Multiobjective optimization for classifier ensemble and feature selection: an application to named entity recognition. International Journal on Document Analysis and Recognition (IJDAR) 15, 143–166.

Elaziz, M.A., Mirjalili, S., 2019. A hyper-heuristic for improving the initial population of whale optimization algorithm. Knowledge-Based Systems 172, 42 – 63.

Eroglu, D.Y., Kilic, K., 2017. A novel hybrid genetic local search algorithm for feature selection and weighting with an application in strategic decision making in innovation management. Information Sciences 405, 18 – 32.

Fawcett, T., 2006. An introduction to ROC analysis. Pattern Recognition Letters 27, 861 – 874.

Freimer, M., Yu, P.L., 1976. Some new results on compromise solutions for group decision problems. Management Science 22, 688–693.

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, 171 – 193.

Guilln, A., Pomares, H., Gonzlez, J., Rojas, I., Valenzuela, O., Prieto, B., 2009. Parallel multiobjective memetic RBFNNs design and feature selection for function approximation problems. Neurocomputing 72, 3541 – 3555.

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, 10–18.

Hancer, E., Xue, B., Zhang, M., Karaboga, D., Akay, B., 2018. Pareto front feature selection based on artificial bee colony optimization. Information Sciences 422, 462 – 479.

de la Hoz, E., de la Hoz, E., Ortiz, A., Ortega, J., Martínez-Álvarez, A., 2014. Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps. Knowledge-Based Systems 71, 322 – 338.

Huang, B., Buckley, B., Kechadi, T.M., 2010. Multi-objective feature selection by using NSGA-II for customer churn prediction in telecommunications. Expert Systems with Applications 37, 3638 – 3646.

Hughes, E.J., 2005. Evolutionary many-objective optimisation: many once or one many?, in: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, 2-4 September 2005, Edinburgh, UK, pp. 222–227.

Ishibuchi, H., Tsukamoto, N., Nojima, Y., 2008. Evolutionary many-objective optimization: A short review, in: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2008, June 1-6, 2008, Hong Kong, China, pp. 2419–2426.

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

Kohavi, R., John, G.H., 1997. Wrappers for feature subset selection. Artificial Intelligence 97, 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.

Lee, D.J., Thornton, A.C., 1996. The identification and use of key characteristics in the product development process, in: 1996 ASME Design Engineering Technical Conference.

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, 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, B., lin Zhang, P., Tian, H., shan Mi, S., sheng Liu, D., quan Ren, G., 2011. A new feature extraction and selection scheme for hybrid fault diagnosis of gearbox. Expert Systems with Applications 38, 10000 – 10009.

Li, D., Jiang, H., 2019. On feature selection in network flow based traffic sign tracking models. Computers & Industrial Engineering 127, 657 – 664.

Li, W., Xiang, D., Tsung, F., Pu, X., 2020b. A diagnostic procedure for high-dimensional data streams via missed discovery rate control. Technometrics 62, 84–100. doi:10.1080/00401706.2019.1575284.

Mafarja, M., Mirjalili, S., 2018. Whale optimization approaches for wrapper feature selection. Applied Soft Computing 62, 441 – 453.

Mafarja, M.M., Mirjalili, S., 2017. Hybrid whale optimization algorithm with simulated annealing for feature selection. Neurocomputing 260, 302 – 312.

Manochandar, S., Punniyamoorthy, M., 2018. Scaling feature selection method for enhancing the classification performance of support vector machines in text mining. Computers & Industrial Engineering 124, 139 – 156.

Min, S.H., Lee, J., Han, I., 2006. Hybrid genetic algorithms and support vector machines for bankruptcy prediction. Expert Systems with Applications 31, 652 – 660.

Mirjalili, S., Lewis, A., 2016. The whale optimization algorithm. Advances in Engineering Software 95, 51 – 67.

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, 1496–1509.

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, 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, 1424–1437.

Pacheco, J.A., Casado, S., Ángel-Bello, F., Alvarez, A.M., 2013. Bi-objective feature selection for discriminant analysis in two-class classification. Knowledge-Based Systems 44, 57 – 64.

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, 1226–1238.

Peres, F.A.P., Fogliatto, F.S., 2018. Variable selection methods in multivariate statistical process control: A systematic literature review. Computers & Industrial Engineering 115, 603 – 619.

Pierre, E.S., Tuv, E., 2011. Robust, non-redundant feature selection for yield analysis in semiconductor manufacturing, in: Industrial Conference on Data Mining, Springer. pp. 204–217.

Robnik-Šikonja, M., Kononenko, I., 2003. Theoretical and empirical analysis of ReliefF and RReliefF. Machine Learning 53, 23–69.

Rosales-Prez, A., Garca, S., Gonzalez, J.A., Coello Coello, C.A., Herrera, F., 2017. An evolutionary multiobjective model and instance selection for support vector machines with pareto-based ensembles. IEEE Transactions on Evolutionary Computation 21, 863–877.

Shang, Y., Zi, X., Tsung, F., He, Z., 2014. Lasso-based diagnosis scheme for multistage processes with binary data. Computers & Industrial Engineering 72, 198 – 205.

Tan, C.J., Lim, C.P., Cheah, Y., 2014. A multi-objective evolutionary algorithm-based ensemble optimizer for feature selection and classification with neural network models. Neurocomputing 125, 217 – 228. Advances in Neural Network Research and Applications Advances in Bio-Inspired Computing: Techniques and Applications.

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

Wold, S., Sjstrm, M., Eriksson, L., 2001. PLS-regression: a basic tool of chemometrics. Chemometrics and Intelligent Laboratory Systems 58, 109 – 130.

Xue, B., Fu, W., Zhang, M., 2014a. Differential evolution (DE) for multi-objective feature selection in classification, in: Genetic and Evolutionary Computation Conference, GECCO ’14, Vancouver, BC, Canada, July 12-16, 2014, Companion Material Proceedings, pp. 83–84.

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, 1656–1671.

Xue, B., Zhang, M., Browne, W.N., 2014b. Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. Applied Soft Computing 18, 261 – 276.

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, 606–626. doi:10.1109/TEVC.2015.2504420.

Yan, X., Jia, M., 2019. Intelligent fault diagnosis of rotating machinery using improved multiscale dispersion entropy and mrmr feature selection. Knowledge-Based Systems 163, 450 – 471.

Yang, H., Kumara, S., Bukkapatnam, S.T., Tsung, F., 2019. The internet of things for smart manufacturing: A review. IISE Transactions 51, 1190–1216. doi:10.1080/24725854.2018.1555383.

Yu, L., Liu, H., 2004. Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research 5, 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, 16–37.

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

Zhang, Y., Cheng, S., Shi, Y., wei Gong, D., Zhao, X., 2019. Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm. Expert Systems with Applications 137, 46 – 58.

Zhang, Y., Wang, S., Phillips, P., Ji, G., 2014. Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. Knowledge-Based Systems 64, 22 – 31.

Zhu, Y., Liang, J., Chen, J., Ming, Z., 2017. An improved NSGA-III algorithm for feature selection used in intrusion detection.

Knowledge-Based Systems 116, 74 – 85.

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.

Zitzler, E., Thiele, L., 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation 3, 257–271.

Zouache, D., Abdelaziz, F.B., 2018. A cooperative swarm intelligence algorithm based on quantum-inspired and rough sets for feature selection. Computers & Industrial Engineering 115, 26 – 36.