Multi-objective feature selection using hybridization of a genetic algorithm and direct multisearch for key quality characteristic selection

An-Da Li [a,*], Bing Xue [b], Mengjie Zhang [b]

[a] *School of Management, Tianjin University of Commerce, Tianjin 300134, China*

[b] *Evolutionary Computation Research Group, School of Engineering and Computer Science, Victoria University of Wellington, Wellington 6140, New Zealand*

Abstract: A multi-objective feature selection approach for selecting key quality characteristics (KQCs) of unbalanced production data is proposed. We define KQC (feature) selection as a bi-objective problem of maximizing the quality characteristic (QC) subset importance and minimizing the QC subset size. Three candidate feature importance measures, the geometric mean (GM), F_1 score and accuracy, are applied to construct three KQC selection models. To solve the models, a two-phase optimization method for selecting the candidate solutions (QC subsets) using a novel multi-objective optimization method (GADMS) and the final KQC set from the candidate solutions using the ideal point method (IPM) is proposed. GADMS is a hybrid method composed of a genetic algorithm (GA) and a local search strategy named direct multisearch (DMS). In GADMS, we combine binary encoding with real value encoding to utilize the advantages of GAs and DMS. The experimental results on four production datasets show that the proposed method with GM performs the best in handling the data imbalance problem and outperforms the benchmark methods. Moreover, GADMS obtains significantly better search performance than the benchmark multi-objective optimization methods, which include a modified nondominated sorting genetic algorithm II (NSGA-II), two multi-objective particle swarm optimization algorithms and an improved DMS method.

Keywords: feature selection; multi-objective optimization; unbalanced data; key quality characteristics; quality improvement

* Corresponding author.
E-mail address: adli@tjcu.edu.cn (An-Da Li).
https://doi.org/10.1016/j.ins.2020.03.032

1. Introduction

Feature selection is an effective dimensionality reduction technique in machine learning and data mining. Feature selection aims to select informative features (or variables) while eliminating irrelevant or redundant features [17, 44]. Recently, feature selection has been employed to select key quality characteristics (KQCs), including the key process parameters, assembly parameters and product parameters, of production data collected from production lines [3, 4, 27, 28, 35]. In these applications, quality characteristics (QCs) are treated as features, and the quality level of products is treated as the class label. KQC selection can be helpful in two aspects. First, KQC selection is an inevitable process that occurs before the implementation of quality control and improvement tools, e.g., statistical process control (SPC) and design of experiments (DOE), because the most critical QCs are identified for simplifying subsequent SPC or DOE processes. Second, KQC selection can be a useful preprocessing step before the application of a learning algorithm for product quality classification [3, 4]. Therefore, identifying KQCs with a tuned feature selection method based on the production data is very beneficial in practice.

Feature selection can be divided into filter methods and wrapper methods according to the evaluation criteria [44]. Filter methods select key features using the intrinsic data properties that are independent of any learning algorithm, whereas feature importance measures can be based on distance [33], information theory [20, 37], rough set theory [8, 49], etc. On the other hand, wrapper methods adopt a learning algorithm to evaluate the importance of a feature subset, and various search strategies are utilized to find the best feature subset with the highest importance value. Two commonly employed heuristic search methods are sequential forward selection (SFS) and sequential backward selection (SBS) [25]. The former starts from an empty set and sequentially adds critical features to the set, while the latter sequentially eliminates noisy or redundant features from a full set. Wrapper methods generally achieve better classification performance than filter methods, but they are more time consuming than filter methods, especially for large scale data [6]. Because the production data (with hundreds of instances) in this paper are not excessively large considering the computational time of a wrapper method, we will build a wrapper-based KQC selection method in this work due to its

high performance.

Wrapper-based feature selection can be defined as a bi-objective optimization model for maximizing the classification accuracy and minimizing the feature subset size from an optimization point of view. This model is an NP-hard optimization problem that has a very large solution space [1]. Therefore, an increasing number of studies use evolutionary computational methods, such as genetic algorithms (GAs) and particle swarm optimization (PSO), to solve wrapper-based feature selection problems. The optimization methods can be further divided into two categories, single-objective methods (e.g., GAs [19, 45] and PSO [5, 43, 47]) and multi-objective methods (e.g., nondominated sorting genetic algorithm II (NSGA-II) [27], multi-objective PSO (MOPSO) algorithms [2, 42], multi-objective artificial bee colony optimization [21] and multi-objective differential evolution algorithms [48]). Single-objective methods should convert the bi-objective problem into a single-objective optimization problem by constructing an integrated objective function with two objectives, which is difficult without domain knowledge. In comparison, multi-objective methods can simultaneously optimize multiple objectives, which is a distinct advantage. The GAs and PSO algorithms generally achieve satisfactory performance in global searches because they can quickly find the desirable regions of solutions; however, they are weak in further exploiting the identified regions [31, 36]. Therefore, combining the multi-objective GAs or PSO algorithms with local search strategies to further improve the search performance is needed to optimize the bi-objective feature selection models.

Data imbalance is observed in real-world classification tasks when the number of instances that belong to various classes is extensively diversified [24]. In the scenario of data with two classes, *imbalance* means that the number of instances of the majority class is greater than the number of instances of the minority class. The production data are actually unbalanced because the number of products on different quality levels (e.g., premium and regular) from production lines differs considerably. Data imbalance can produce biased KQC selection results but most existing KQC selection methods [3, 27, 35] that use feature selection strategies have not addressed this issue. In these methods, overall accuracy is often employed as the QC (feature) importance measure. However, a high value of overall accuracy is not equivalent to an acceptable classification performance because a high accuracy rate for the majority class is

sufficient for obtaining a high value of overall accuracy according to its definition. In most practical applications, it is preferred that the classification methods perform well for the minority class, with a small classification performance loss for the majority class [7]. For instance, in the quality control scenario, whether the defective products (minority class instances) can be detected in the production processes is more critical than the conforming products (majority class instances) [41].

To solve the data imbalance problem, some feature selection studies have adopted the true positive rate (TPR) and the true negative rate (TNR) instead of (or in addition to) accuracy to measure the feature importance [11, 15, 30, 32, 34]. The idea of these studies is the use of the classification performance for the minority class (e.g., TPR) as one additional objective function to solve the evaluation bias problem of overall accuracy. However, additional objective functions indicate additional challenges for optimization methods, because the optimization performance of multi-objective optimization methods decreases with an increase in the number of objective functions [22]. To reduce the number of objectives, Li et al. [28] adopted the geometric mean (GM) of the TPR and TNR as the feature importance measure of unbalanced data. Since either a low value of TPR or TNR will significantly decrease the GM value, the GM is more sensitive to unbalanced data. However, in this paper, the effects of the GM are not directly compared with other potential classification performance measures for feature selection. It is beneficial to compare the effects of various classification performance measures in evaluating the feature importance on unbalanced data.

GAs comprise one type of metaheuristic algorithm and are commonly applied for NP-hard problems. They simulate the evolutionary process of chromosomes with selection, crossover and mutation operators and have excellent global search abilities. However, the weakness of standard GAs is evident in the local search; therefore, improving the performance of GAs by local search strategies is needed [31]. Direct multisearch (DMS) is a recently proposed multi-objective optimization algorithm that adopts a local search strategy named “poll step” to obtain new solutions around the current nondominated (best) solutions [9]. DMS is a numerical competitive algorithm for multi-objective optimization problems that has shown excellent convergence behaviors for continuous optimization problems. Several studies have focused on improving the global search performance of DMS. In [10], Custódio and Madeira proposed the

MultiGLODS algorithm, which adopts several strategies, including new searches initialization using a multistart strategy and a promising subregion exploring strategy to enhance the global search performance of DMS. In [28], Li et al. proposed an improved DMS (IDMS) that embeds a mutation operator, which is commonly employed in GAs, to improve the global search performance of DMS for the multi-objective KQC selection problem. Based on this analysis, we will establish a new multi-objective optimization algorithm that combines the search strategies of GAs and DMS for KQC selection. We further investigate if combining the complete evolving strategies (selection, crossover and mutation) in GAs can achieve better search performance than the IDMS. Furthermore, we will investigate if introducing the DMS strategy in GAs can improve the performance of GAs.

In this work, KQC (feature) selection models are established as a bi-objective optimization problem of maximizing the QC subset importance and minimizing the QC subset size. Three QC subset importance measures (GM, F_1 score and accuracy) are adopted to establish three candidate KQC selection models. To solve the models, we propose an optimization method named GADMS-IPM, which combines the multi-objective optimization method GADMS and the ideal point method (IPM):

- GADMS is proposed by combining a GA and the local search strategy DMS. To better inherit the advantages of GA and DMS methods, both binary encoding and real value encoding methods are employed. The GA process employs binary-encoded solutions to update the nondominated set, and the DMS process employs real-encoded solutions to update the nondominated set. A conversion method is proposed to convert these two types of encoded solutions.
- To sort the solutions during the optimization process of GADMS, a modified fast nondominated sorting method for feature selection is employed. This method detects duplicate solutions and lowers their importance at each generation to retain the population diversity for improving the search performance.
- In each DMS process, a local search is performed to update the nondominated set. First, a poll center (solution) is selected from the current nondominated set. Second, the algorithm searches around the poll center to generate new solutions for updating the nondominated set.
- The GADMS method applies a caching strategy to reduce the time cost. In the caching strategy, a cache is applied to store the evaluated objective function values, which reduces the calls of the function evaluation process.
- As a multi-objective optimization method, GADMS obtains a set of nondominated

solutions as candidate KQC sets. Therefore, we adopt the IPM [14] to select the final KQC set from the candidate solutions from the perspective of practical utility.

The experimental results on four unbalanced production datasets show that the GADMS-IPM method with the GM measure outperforms the GADMS-IPM methods with the other two measures (F_1 score and accuracy) and benchmark methods. The results also show that the proposed multi-objective optimization method GADMS obtains better search performance than the four benchmark methods, i.e., a modified NSGA-II (MNSGAII) [27], two MOPSO algorithms (NSPSO and CMDPSO) [42] and the IDMS method [28].

The remainder of this paper is organized as follows. Section 2 briefly describes the KQC selection models. Section 3 presents the proposed optimization method GADMS-IPM. Section 4 describes the experimental design and settings. Section 5 presents the KQC selection results and discussions. Section 6 discusses the search ability of GADMS. Section 7 further analyzes the effectiveness of the proposed method using synthetic data. Section 8 discusses the conclusions and future work.

2. KQC selection models

Let D ($K \times N$ matrix) be a production dataset with N QCs (features) and K products (instances). The QCs can be denoted by a QC set $Q = \{q_1, q_2, \ldots, q_N\}$, where each q_i, $i = 1, 2, \ldots, N$ denotes a QC in Q. The products are classified into two classes in terms of the quality levels (i.e., high quality and low quality). The objective of KQC selection is to select the critical QCs that are relevant to the product quality and eliminate as many of the redundant or irrelevant QCs as possible. Thus, the KQC selection model can be defined as a feature selection problem of

$$\begin{array}{ll} \max & J(X) \\ \min & |X| \\ s.t. & X \subseteq Q, X \neq \emptyset \end{array} \tag{1}$$

where X denotes a QC subset, $|X|$ denotes its size (number of QCs contained), and the function J evaluates the importance of X. Generally, J is a measure that estimates the correlation between X and the quality level (class label) in a filter method and a classification performance measure that estimates X's predictive power of the quality level in a wrapper method. In this paper, we utilize the wrapper framework to build the feature selection approach. In the following paragraphs, we introduce the commonly employed classification performance

measures, from which we obtain three importance measures.

Because the production datasets in this paper have two quality levels, the learning task is a binary classification problem. The confusion matrix for the binary classification problem is shown in Table 1, where TP, TN, FN and FP represent the number of positive instances correctly classified to the positive (minority) class, negative instances correctly classified to the negative class, positive instances incorrectly classified to the negative class, and negative instances incorrectly classified to the positive class, respectively.

Table 1. Confusion matrix.

		Predicted Class	
		Positive (minority)	Negative (majority)
True Class	Positive (minority)	TP	FN
	Negative (majority)	FP	TN

According to Table 1, the accuracy rate (ACC) can be calculated as

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}. \tag{2}$$

Accuracy is generally used to measure the feature subset importance in feature selection problems. However, this accuracy is not sensitive to the instances of the minority class when the dataset is unbalanced.

For unbalanced data, the GM [26] of the TPR and TNR is a commonly applied performance measure, which is defined as

$$GM = \sqrt{TPR \cdot TNR}, \tag{3}$$

$$TPR = recall = \frac{TP}{TP + FN}, \tag{4}$$

$$TNR = \frac{TN}{TN + FP}. \tag{5}$$

TPR measures the ratio of correctly classified positive instances to all actual positive instances, and TNR measures the ratio of correctly classified negative instances to all actual negative instances. Compared with the accuracy, either a low TPR value (referred to as recall in the information retrieval context) or TNR value yields a low GM value. A high GM value requires high TPR and TNR values. Therefore, the GM is more sensitive to the instances of the minority class than the accuracy.

The F_1 score is another commonly employed classification performance measure for unbalanced data [29]. This score is defined as the harmonic mean of precision and recall as

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}, \tag{6}$$

where the precision is defined as

$$precision = \frac{TP}{TP + FP}. \tag{7}$$

The F_1 score (larger is better) is a value in $[0,1]$ that measures whether the instances of the minority class are correctly classified without too many instances of the majority class that are misclassified to the minority class. As previously stated, whether the products with the minority quality level (e.g., defective products) can be precisely detected is considerably important for the production processes; this notion is consistent with the objective of the F_1 score. Therefore, applying the F_1 score to measure the QC importance also makes sense. Note that the harmonic mean expression for F_1 is not defined when $TP = 0$ since both the numerator and the denominator of Eq. (6) are equal to 0. In this extreme case, the value of F_1 is defined as 0 [29]. Similarly, if all instances are classified into the negative class, both TP and FP will be equal to 0, which makes both the numerator and the denominator in Eq. (7) equal to 0. This value denotes an undesirable classification result. In this case, therefore, we define the value of precision to be 0.

According to this analysis, the GM and the F_1 score are two sensitive performance measures of the unbalanced data, and accuracy is a commonly employed performance measure for feature importance evaluation. In this paper, we want to investigate the performance of GM, F_1 score and accuracy for KQC selection using unbalanced production data. Thus, these three measures are adopted to establish three bi-objective KQC selection models, as shown in Table 2. In the table, the first objective function is $1 - measure$; thus, each objective function is minimized, which is beneficial for common optimization algorithms.

Table 2. Three defined KQC selection models.

	Model-GM	Model-F_1	Model-ACC
$\min f_1 =$	$1 - GM(X)$	$1 - F_1(X)$	$1 - ACC(X)$
$\min f_2 =$	$\lvert X \rvert$	$\lvert X \rvert$	$\lvert X \rvert$
s.t.	$X \subseteq Q, X \neq \emptyset$	$X \subseteq Q, X \neq \emptyset$	$X \subseteq Q, X \neq \emptyset$

3. Optimization approach

3.1 Outline of the proposed two-phase optimization approach

In Section 2, we established three bi-objective KQC selection models to maximize a QC subset importance measure (GM, F_1 score or accuracy) and minimize the QC subset size. In this section, we propose a hybrid multi-objective optimization method named GADMS that combines the GA and DMS to solve the models. As one type of multi-objective approach, the GADMS method obtains a set of nondominated solutions. From a practical point of view, reducing the number of final solutions is necessary. Therefore, the IPM is adopted in this paper to select the final KQC set from the nondominated solutions obtained by GADMS. The entire method is referred to as GADMS-IPM.

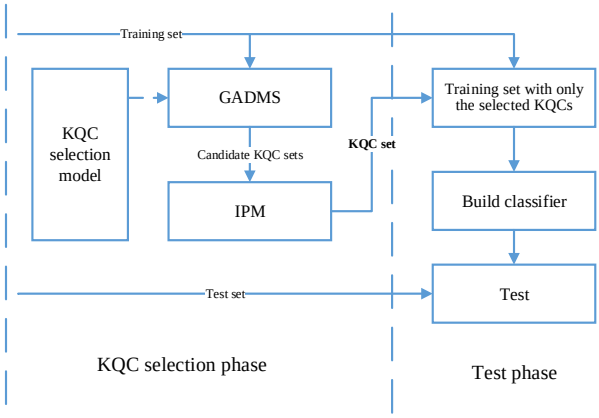

Fig. 1. Framework of GADMS-IPM.

Fig. 1 shows the framework of the GADMS-IPM method, which contains both the KQC selection and test phases. In the KQC selection phase, the original production dataset is divided into a training set and a test set. The training set is then input to the proposed GADMS-IPM method to select the KQC set. In the test phase, the training set with the selected KQCs is applied to build a classifier, and then the classification results for the test set are acquired to evaluate the effectiveness of the KQC set. The details of the two phases of GADMS-IPM, i.e., GADMS and IPM, are introduced in Sections 3.2 and 3.3.

3.2 Proposed multi-objective optimization method (GADMS)

GAs are one type of metaheuristic algorithm with competent global search performance, and DMS can apply the poll step to update solutions by performing a local search around the nondominated solutions in the optimization process. To inherit the global and local search advantages of both GAs and DMS, we build a hybrid multi-objective optimization method named GADMS, which will be applied to the defined KQC selection models. Fig. 2 shows the

flowchart of GADMS. According to Fig. 2, the set NS, which stores the nondominated solutions, is created as the connection between the GA process and the DMS process. In this paper, we set the maximum size of NS equal to the population size N_p. If the number of nondominated solutions at a generation exceeds the size of NS, only the first N_p solutions obtained from the sorting method are retained. Both the GA process and the DMS process can update NS. At each generation, the GA process evolves new solutions (O_t) based on the population (P_t), whereas the DMS process focuses only on searching around NS to generate new solutions (S_{poll}). The new solutions evolved by the GA process (O_t) or the DMS process (S_{poll}) are combined with NS to form the sorting pool R. The solutions in R are then sorted according to their objective values. We can update NS by replacing the solutions in it with the nondominated solutions in R. Note that the DMS process performs only at generations where the GA process fails to update NS, which means that the DMS process is an assistant step for generating new solutions. The proposed GADMS method is a metaheuristic algorithm. Since the GA process adopts a binary encoding strategy for the solutions, the convergence cannot be guaranteed for GADMS. The results of the convergence property analysis for DMS in [9] do not apply to GADMS. In the following sections, details of GADMS, including solution encoding, the solution sorting strategy, genetic operators and the poll step, are described.

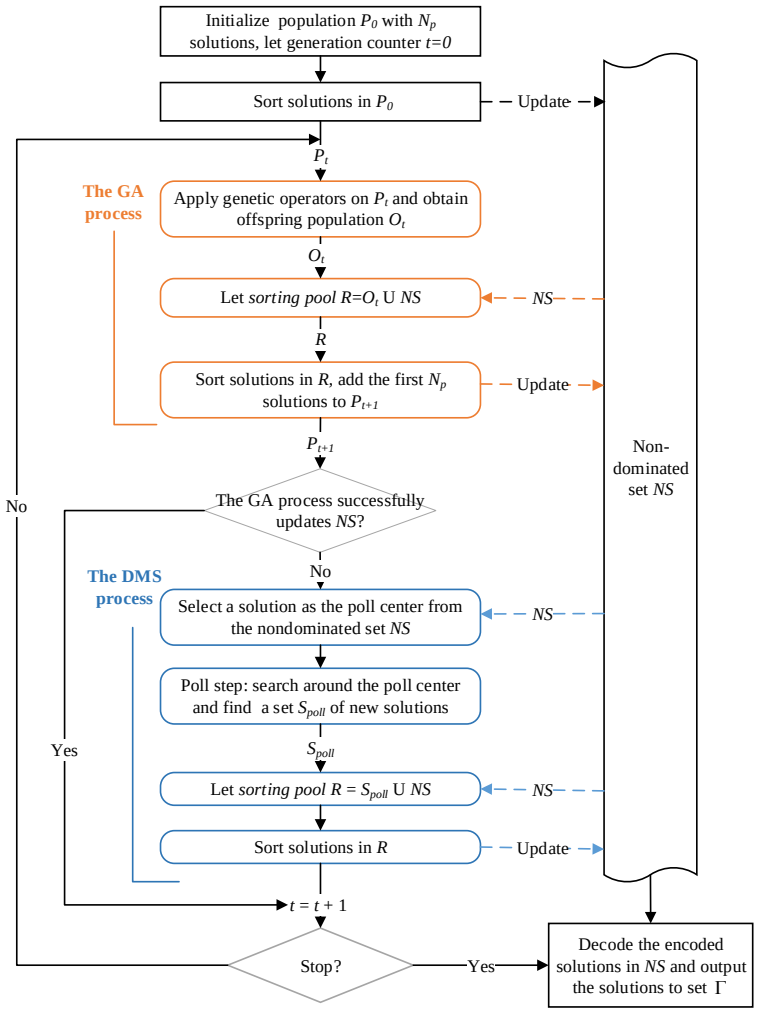

Fig. 2. Flowchart of GADMS.

A. Solution encoding

In feature selection applications that employ GAs, binary encoding is often employed to represent a solution (individual) because a binary vector can be easily applied to represent a feature subset and simplify the crossover and mutation operations. However, in DMS, a real-valued vector is required to denote a solution since DMS is designed for real-valued problems. In this paper, we employ binary encoding and real value encoding strategies to inherit the advantages of both GAs and DMS. Specifically, binary-encoded solutions are utilized in the GA process, and real-encoded solutions are employed in the DMS process. Because the nondominated set NS is the connection between the GA process and the DMS process, the solutions in NS should be encoded in binary and real-valued ways. The binary-encoded solutions in NS from the GA process should be converted to real-encoded solutions, and vice versa.

Binary encoding strategy. Let N be the total number of QCs. A QC subset (solution) X in the KQC selection model is encoded by an *N*-bit binary vector $\boldsymbol{X_B} = (x_{b1}, x_{b2}, \ldots, x_{bN})$, in which a bit of "1" denotes the selection of the corresponding QC (feature) and a bit of "0" denotes the elimination of the corresponding QC.

Real value encoding strategy. A real-encoded solution is denoted by a vector of N real numbers, in which each number corresponds to a QC. Let $\boldsymbol{X_R} = (x_{r1}, x_{r2}, \ldots, x_{rN})$ be a real-encoded solution, then $x_{ri} \in (0.5,1]$ denotes the selection of the ith QC and $x_{ri} \in [0,0.5]$ denotes the elimination of the *i*th QC, where $i = 1, \ldots, N$.

Encoding conversion strategy. A binary-encoded solution $\boldsymbol{X_B} = (x_{b1}, x_{b2}, \ldots, x_{bN})$ is converted to a real-encoded solution $\boldsymbol{X_R} = (x_{r1}, x_{r2}, \ldots, x_{rN})$ with each element

$$x_{ri} = \begin{cases} rand(0,0.5), if\ x_{bi} = 0 \\ rand(0.5,1), if\ x_{bi} = 1 \end{cases}, i = 1, \ldots, N, \tag{8}$$

where $rand(a, b)$ denotes a random uniform value in (a, b). Conversely, a real-encoded solution $\boldsymbol{X_R} = (x_{r1}, x_{r2}, \ldots, x_{rN})$ is converted to a binary-encoded solution $\boldsymbol{X_B} = (x_{b1}, x_{b2}, \ldots, x_{bN})$ with each bit

$$x_{bi} = \begin{cases} 0, x_{ri} \in [0,0.5] \\ 1, x_{ri} \in (0.5,1] \end{cases}, i = 1, \ldots, N. \tag{9}$$

An example to illustrate the binary and real value encodings is shown in Fig. 3, where the total number of QCs is assumed to be $N = 10$.

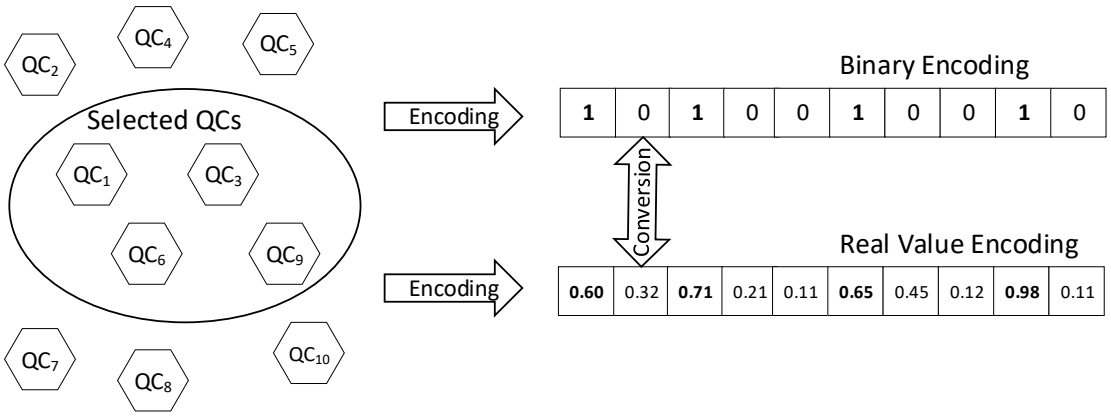

Fig. 3. Illustration of binary encoding and real value encoding.

B. Sorting solutions

Sorting the solutions in descending order of their goodness is critical for a population-based multi-objective optimization method. One extensively applied sorting method is the fast nondominated sorting method and the crowding distance measure proposed in [12]. The fast

nondominated sorting method divides the solutions in the population into different dominance ranks (lower is better). The solutions with the same rank are in the same nondominated front. In this sorting method, first, the nondominated solutions in the population are identified and assigned a rank valued 1. Second, the nondominated solutions in the reduced population (by eliminating the solutions with rank of 1) are identified and assigned a rank valued 2. The process continues until the ranks of all of the solutions are assigned. The crowding distance measure evaluates the density of each solution. The crowding distance for a solution is calculated as the average side length of the cuboid composed of its two nearest solutions in the same nondominated front. With the dominance ranks and crowding distances, either two solutions in the population can be compared. Given solutions $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, let r_1 and r_2 be the dominance ranks, and let cd_1 and cd_2 be the crowding distances. $\boldsymbol{X}_1$ is better than $\boldsymbol{X}_2$ when $r_1 < r_2$ or ($r_1 = r_2$ & $cd_1 > cd_2$). Additional details about the fast nondominated sorting method and the crowding distance calculation can be obtained from [12].

For feature selection, which is a combinatorial optimization problem, the standard fast nondominated sorting method does not have a step to detect and eliminate the duplicate solutions in the population, which may reduce the population diversity. For instance, assume that we have a population P, which contains several identical solutions (actually refer to the same feature subset, denoted by X) with high fitness values. If the selection operation of GAs is directly performed on P, X will have more chances than the solutions with similar fitness values being selected as the parents, which produces a large number of similar offspring solutions. Consequently, the population diversity is reduced. Therefore, reducing the number of duplicate solutions in the population is necessary for a population-based optimization method. In this paper, we adopt the modified fast nondominated sorting method [27] proposed for feature selection. This sorting method is capable of detecting the duplicate solutions and changing their ranks to improve the population diversity. The algorithmic description that corresponds to the modified fast nondominated sorting method is shown in Algorithm 1. In the modified sorting method, first, the solutions in the sorting pool are sorted according to the dominance ranks and crowding distances obtained by the standard fast nondominated sorting and crowding distance calculation methods. Second, an additional step is added to increase the dominance ranks of duplicate solutions, and the solutions are sorted based on the updated

dominance ranks. In the GA process, the first N_p solutions in the sorted pool are always selected to be the population of the next generation. Thus, this modification reduces the possibility that some reasonable but duplicate solutions are selected as the parents too many times, which can generate a large number of similar offspring solutions.

Input: Sorting pool R;
Output: The sorted pool R_s;

1. Use the standard fast nondominated sorting method to assign the dominance rank $r(\boldsymbol{X_B})$ and obtain the crowding distance $cd(\boldsymbol{X_B})$ for each solution $\boldsymbol{X_B} \in R$;
2. Sort the solutions in R according to the dominance ranks and crowding distances;
3. Modify the dominance ranks of duplicate solutions;
   3.1 Let $R_U = \emptyset$, $R_r = \emptyset$, and $r_m = \max_{\boldsymbol{X_B} \in R} r(\boldsymbol{X_B})$;
   3.2 **For** each solution $\boldsymbol{X_B} \in R$
   3.3 If $\boldsymbol{X_B} \in R_U$, let $r(\boldsymbol{X_B}) = r_m + 1$ and $R_r = R_r \cup \{\boldsymbol{X_B}\}$; or let $R_U = R_U \cup \{\boldsymbol{X_B}\}$;
   3.4 **End** For
4. Let $R_s = R_U \cup R_r$;
5. **Return** the sorted pool R_s;

Algorithm 1. Pseudocode of the modified fast nondominated sorting method.

Let M be the number of objectives and N_R be the size of the sorting pool R; the time complexity of the traditional sorting method (steps 1 and 2 in Algorithm 1) is $O(M{N_R}^2)$, as introduced by Deb et al. [12]. Moreover, the time complexity of step 3 in Algorithm 1 is $O(N_R^2)$. Thus, the time complexity of the modified fast nondominated sorting method is $O\left(M{N_R}^2\right) + O(N_R^2) \cong O\left(M{N_R}^2\right)$.

C. Genetic operators

Crossover operator. The standard one-point crossover operator is one of the most commonly employed crossover operators in GAs. Let $\boldsymbol{X_B^m} = (x_{b1}^m, x_{b2}^m, \dots, x_{bN}^m)$ and $\boldsymbol{X_B^n} = (x_{b1}^n, x_{b2}^n, \dots, x_{bN}^n)$ be two parent solutions. The standard one-point crossover operator produces the two offspring solutions $\boldsymbol{X_B^{m\prime}} = (x_{b1}^m, \dots, x_{bi}^m, x_{b(i+1)}^n, \dots, x_{bN}^n)$ and $\boldsymbol{X_B^{n\prime}} = (x_{b1}^n, \dots, x_{bi}^n, x_{b(i+1)}^m, \dots, x_{bN}^m)$, where i is a randomly generated crossover point in $[1, N-1]$. However, this operation does not ensure success of the crossover. In Fig. 4, for example, offspring solutions are exactly the same as the parents since the bits on the right side of the

crossover point are the same. To solve this problem, we propose a modified one-point crossover operator, which identifies the positions of the bits of different values in the two parents, and then generates a crossover point within the identified positions to ensure that the generated offspring solutions are new. Let $\boldsymbol{X_B^m} = (x_{b1}^m, x_{b2}^m, \dots, x_{bN}^m)$ and $\boldsymbol{X_B^n} = (x_{b1}^n, x_{b2}^n, \dots, x_{bN}^n)$ be two paired binary-encoded parents. First, we obtain the index set $\Phi = \{l_1, l_2, \dots, l_d\}$ (in increasing order) from the two solutions, where $x_{bl_c}^m \neq x_{bl_c}^n$ for each l_c $(c = 1,2,\dots,d)$. Second, a crossover point l_c $(c = 2,3,\dots,d)$ is selected and two offspring solutions are generated as $\boldsymbol{X_B^{m\prime}} = (x_{b1}^m, \dots, x_{b(l_c-1)}^m, x_{bl_c}^n, \dots, x_{bN}^n)$ and $\boldsymbol{X_B^{n\prime}} = (x_{b1}^n, \dots, x_{b(l_c-1)}^n, x_{bl_c}^m, \dots, x_{bN}^m)$. In this paper, the binary tournament selection method is adopted to select N_p parents. In each binary tournament selection, two solutions randomly selected from the population are compared, and the better solution is selected as the parent. N_p parents are randomly paired, and the offspring solutions are generated by the modified one-point crossover with the crossover probability of p_c.

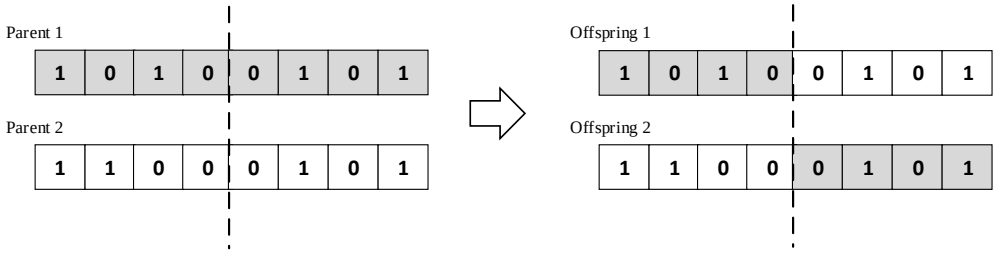

Fig. 4. Illustration of crossover.

Mutation operator. In this paper, the subset size-oriented mutation (SSOM) operator [31] for feature selection problems is employed. Let $\boldsymbol{X_B} = (x_{b1}, x_{b2}, \dots, x_{bN})$ be a binary-encoded solution, $\#bits_1$ be the number of bits of "1", $\#bits_0$ be the number of bits of "0", p_{m1} and p_{m0} be the mutation probabilities for bits of "1" and "0", respectively. Given p_{m1}, the mutation probability p_{m0} is calculated as

$$p_{m0} = \frac{\#bits_1}{\#bits_0} \cdot p_{m1}. \tag{10}$$

SSOM does not change the proportion of bits of "1" to bits of "0" from a statistical perspective, as the expected number of mutated bits of "0" and "1" are the same for a solution. This property of SSOM is beneficial for feature selection problems. For example, if a solution has fewer bits of "1" than bits of "0", the conventional bitwise mutation operator changes bits from "0" to "1" more frequently than those from "1" to "0", as each bit has the same mutation

probability. This change actually prevents the GA from eliminating the features and decreases the possibility of eliminating features for a feature selection task. In comparison, the SSOM avoids this trend by balancing the number of bits of "1" and bits of "0" to be mutated.

The time complexities of the binary tournament selection method, crossover operator and mutation operator are $O(N_p)$, $O(N_pN)$ and $O(N_pN)$, respectively, where N_p is the population size and N is the length of solutions. Therefore, the time complexity of genetic operators is equal to $O(N_p) + 2 \cdot O(N_pN) \cong O(N_pN)$.

D. Poll step

In the poll step, a poll center (solution) $\boldsymbol{X_R}$ is selected from NS, and a set S_{poll} of new solutions are searched around the selected poll center as

$$S_{poll} = \{norm(\boldsymbol{X}_R + \alpha_{\boldsymbol{X}_R}\boldsymbol{d}) | \boldsymbol{d} \in D\}, \quad (11)$$

where D is a randomly generated positive spanning set, $\boldsymbol{d}$ is a vector of N real numbers that define the search direction, and $\alpha_{\boldsymbol{X}_R}$ is the step size parameter of the solution $\boldsymbol{X_R}$. A large $\alpha_{\boldsymbol{X}_R}$ value means searching in a large space, and vice versa. In this paper, we let the size of D be $2N$ (N is the length of solutions), which means that $2N$ vectors exist in D. Thus, the poll step generates $2N$ new solutions to S_{poll}. Moreover, in Eq. (11), we use the function $norm$ to ensure that each element in the generated solution is in $[0,1]$. If an element is larger than 1, it is set to 1; if it is smaller than 0, it is set to 0. The step size parameter $\alpha_{\boldsymbol{X}_R}$ may change after the poll step. If the poll step fails to update NS, $\alpha_{\boldsymbol{X}_R}$ shrinks to $\beta \cdot \alpha_{\boldsymbol{X}_R}$, where $\beta \in (0,1)$ is a user-defined step size updating parameter. The decrease in the step size parameters during the optimization process causes the poll step gradually to perform more local searches. For each new poll solution added to NS, the value of its step size parameter inherits that of the poll center $\boldsymbol{X}_R$, which equals $\alpha_{\boldsymbol{X}_R}$. As the GA process updates NS, the initial step size value α_0 is assigned to newly added solutions by the GA process. For additional details about the poll step, please refer to [9].

Selection of the poll center. The following conditions are applied for the selection of the poll center: 1) the solutions with the largest step size parameter are obtained from NS, and 2) the solutions with the largest crowding distance are further selected from the solutions obtained in "step 1)". The first condition ensures a balanced selection of the poll center because a solution

with a smaller step size parameter denotes that its neighborhood has been sufficiently searched. Selecting a solution with a larger step size parameter as the poll center means to search in the space with fewer search attempts. The second condition means that the algorithm spends more effort on searching around the solutions in the sparse area of the nondominated front.

E. Improving the time efficiency

If the time cost of each function evaluation is not considered, the time complexity of GADMS at each generation is decided by the GA and DMS processes. Let M, N_p, N, N_R and N_{NS} be the number of objectives, population size, length of solutions, size of the sorting pool R and size of NS, respectively. The complexity of the GA process is equal to $O\left(MN_R{}^2\right) + O(N_pN)$, which is composed of the time complexities of the sorting process and genetic operators. Since the sorting pool R in the GA process is $O_t \cup NS$ and the maximum size of NS is N_p, $N_R \leq 2N_p$. Thus, the time complexity of the GA process is $O\left(MN_R{}^2\right) + O(N_pN) \cong O\left(4MN_p{}^2\right) + O(N_pN) \cong O\left(MN_p{}^2\right) + O(N_pN)$. The time complexity of the DMS process is governed by the sorting process, which is $O(MN_R^2)$. In the DMS process, the size of the sorting pool $N_R = 2N + N_{NS} \leq 2N + N_p$. Thus, the time complexity of the DMS process is $O(MN_R^2) \cong O(M\left(2N + N_p\right)^2)$. At a generation, if both the GA process and the DMS process perform, the time complexity equals $O\left(MN_p{}^2\right) + O(N_pN) + O(M\left(2N + N_p\right)^2)$. Therefore, the final time complexity of GADMS is $O(MN_p{}^2)$ if $N_p > N$; otherwise, it is $O(MN^2)$ if $N_p \leq N$.

For wrapper-based feature selection methods, the function evaluation process is time consuming since a learning algorithm is used to evaluate the objective function values. For evolutionary algorithms, it is possible that an already evaluated solution is regenerated during the subsequent generations, and re-evaluating this solution is useless. Therefore, we adopt a caching strategy to improve the time efficiency of GADMS. The function values of each evaluated solution are stored in the cache L_e. For a newly generated solution during the optimization process, the objective function values are directly obtained from L_e if the solution is in L_e. Otherwise, the objective function values of the new solution are evaluated.

3.3 Ideal point method (IPM)

The IPM [14] is adopted to select the final KQC set from the nondominated solutions

(candidate KQC sets) obtained by GADMS. The procedure of the IPM is shown in Algorithm 2, which includes three steps. First, the values of the objective functions are normalized using the Z-score normalization method to solve the problem of incommensurability between the two objective functions. Second, the ideal point is defined according to the minimum values of f_1^N and f_2^N of the candidate KQC sets. Third, the solution with the minimum Euclidean distance to the ideal point is selected as the final KQC set.

Input: A set Γ of candidate KQC sets obtained by GADMS;

Output: KQC set X^*;

1. For each solution $X \in \Gamma$, let $f_i^N(X) = (f_i(X) - \overline{f_i(X)})/\sigma(f_i(X)), i = 1,2$;
2. Let the ideal point be $(f_1^*, f_2^*) = (\min_{X\in\Gamma}(f_1^N(X)), \min_{X\in\Gamma}(f_2^N(X))$;
3. **Return** $X^* = arg \min_{X\in\Gamma} (\sqrt{\sum_{i=1}^{2}(f_i^N(X) - f_i^*)^2})$;

Algorithm 2. Pseudocode of IPM.

The GADMS-IPM framework is used to optimize the three bi-objective KQC selection models with the GM, F_1 score and accuracy measures, respectively, which forms three KQC selection methods. The three methods with the GM, F_1 score and accuracy measures are denoted by GADMS-IPM(G), GADMS-IPM(F) and GADMS-IPM(A), respectively.

4. Design of Experiments

4.1 Datasets

Four production datasets, i.e., ADPN, LATEX, PAPER, and SPIRA, are employed in the experiments. ADPN, LATEX and SPIRA were utilized by Gauchi and Chagnon [16], and PAPER was employed by Wold et al. [39] to build prediction models with partial least squares (PLS) regression. ADPN was collected from a subphase of nylon production (i.e., adiponitrile production process); its QCs include pressure, temperature, flow, etc. LATEX was collected from the polymerization process of latex production; its QCs include the catalyst level, temperature, reactive concentration, etc. PAPER was collected from the process of paper recycling; its QCs include temperature and concentration measures at various time points. SPIRA was collected from the production process of an antibiotic; its QCs include the

temperature level, peak oxygen consumption, stirring power, etc. Anzanello et al. [3] divided the instances in these datasets into two classes according to the response variable and built a variable selection method for product quality classification. In particular, the instances (products) are divided into two quality levels: premium (minority class) and regular (majority class). The details of the datasets are shown in Table 3.

Table 3. Details of the datasets.

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

4.2 Benchmark methods and parameter settings

Six benchmark methods, SFS [25], SBS [25], NSPSOFS [42], CMDPSOFS [42], NSGAII-IPM [27] and IDMS-IPM [28], are employed in the experiments. SFS and SBS [25] are two conventional feature selection methods based on greedy search strategies. NSPSOFS and CMDPSOFS [42] are feature selection methods based on two MOPSO algorithms (NSPSO and CMDPSO), where the two objectives are to maximize the accuracy and minimize the feature (QC) subset size. Note that these two methods do not contain a process for selecting a single solution from the obtained nondominated solutions. Thus, the IPM is used to select the final solution after NSPSOFS and CMDPSOFS are performed. NSGAII-IPM [27] is a KQC selection method based on MNSGAII and IPM. Similar to NSPSOFS and CMDPSOFS, NSGAII-IPM attempts to maximize the accuracy and minimize the feature subset size. IDMS-IPM [28] is a KQC selection method based on IDMS and IPM. IDMS-IPM attempts to maximize the GM measure and minimize the feature subset size.

The parameters employed by GADMS-IPM are listed as follows: population size $N_p = 100$, crossover probability $p_c = 0.9$, mutation probability $p_{m1} = 1/N$ (N is the number of original QCs) and step size updating parameter $\beta = 0.9$. Additionally, the initial step size value is set to $\alpha_0 = 1$, as suggested by Custódio et al. [9]. For GADMS-IPM, the number of function evaluations at each generation varies; thus, the termination criterion for GADMS-IPM is set to a maximum number of evaluations of 10,000 to fairly compare with other methods. In

NSPSOFS, the inertia weight is set to $w = 0.7298$ and the acceleration parameters are set to $c_1 = c_2 = 1.49618$. In CMDPSOFS, the inertia weight is set to $w \in [0.1,0.5]$, and the mutation rate is set to $p = 1/N$. These parameters are the same parameters used in [42]. In NSPSOFS and CMDPSOFS, the population size and number of generations are set to 100, which yields the same number of function evaluations in GADMS-IPM. In NSGAII-IPM, the crossover probability is set to $p_c = 0.9$, and the mutation probability is set to $p_m = 1/N$, as shown in [27]. Both the population size and the number of generations are set to 100 to ensure the same number of function evaluations employed in GADMS-IPM. In IDMS-IPM, the maximum number of function evaluations is set to 10,000, which is the same as that in GADMS-IPM. Other settings are the same as those utilized in [28]. For SFS and SBS, the default settings in the Waikato Environment for Knowledge Analysis (Weka) [18] are employed.

All the experiments were conducted on an Intel Core PC with a 3.4 GHz CPU and 8 GB main memory. The GADMS-IPM, NSPSOFS, CMDPSOFS, NSGAII-IPM and IDMS-IPM methods are implemented in MATLAB R2016b. SFS and SBS are implemented in Weka 3.7.13. Because all of the KQC selection methods require a learning algorithm to evaluate the solutions in the KQC selection phase and validate the selection results in the test phase, we use the naïve Bayesian (NB) classifier [23] as the learning algorithm since it is a high performance and concise method. The NB classifier employed by each method is invoked from Weka. Note that the PAPER dataset is highly unbalanced because the imbalance ratio of the majority class instances to the minority class instances exceeds 10 (351/33), which is significantly larger than the ratios of the other datasets. This imbalance may have a negative effect because the trained NB classifier is unreliable [7]. To avoid this effect, a modified NB classifier is adopted on PAPER. In the modified classifier, the training set is balanced by a simple upsampling method that reproduces the minority class instances for $[\frac{\#maIns}{\#miIns}] - 1$ times (where $\#maIns$ and $\#miIns$ denote the number of majority class instances and the number of minority class instances) and uses it as the input to train the original NB classifier.

To verify the performance of the proposed method, 10-fold stratified cross-validation (CV) [40] is used to generate the training and test sets. In 10-fold CV, the dataset is randomly divided into ten folds. Then, a selection process that selects a fold as the test set and combines the

remaining nine folds as the training set is performed. This selection process repeats ten times until each fold is selected as the test set once. Thus, ten pairs of the training set and test set are generated. Each pair of the training set and test set is input into the KQC selection and test phases, as shown in Fig. 1. Because our proposed method and the benchmark methods, except for SFS and SBS, are based on stochastic searching strategies, we repeat the CV 3 times to comprehensively test the effectiveness of the methods. Therefore, $10 \times 3=30$ runs of experiments on each dataset are conducted, which yields 30 test results. In the KQC selection phase, an inner 5-fold CV, which is a commonly employed evaluation method in wrapper-based feature selection methods [6, 25], is applied to the training set to evaluate the objective function f_1 (GM, F_1 score or accuracy) of a given QC subset X.

To evaluate the KQC selection results, two types of performance measures are employed in the experiments. The first type includes the TPR, TNR, accuracy, GM, and F_1 score, as described in Section 2. These measures are used to comprehensively evaluate the product quality prediction performance of the KQCs selected by each method. The TPR measures the classification performance of the minority class instances (premium products), while the TNR measures the classification performance of the majority class instances (regular products). The accuracy and GM are integrated classification performance measures for instances of both majority and minority classes. The F_1 score is an integrated classification performance measure of recall and precision. The second type is the number of selected KQCs, which measures whether a feature selection method can effectively eliminate irrelevant or redundant QCs.

5. KQC selection results and analysis

5.1 Comparison among the three proposed KQC selection methods

In this section, the three proposed methods, i.e., GADMS-IPM(G), GADMS-IPM(F) and GADMS-IPM(A), are compared. Table 4 shows the KQC selection results, where the average and standard deviation of each performance measure over 30 runs of experiments are listed, and the Wilcoxon signed-rank test [38], with a significance level of 0.05, is used to test whether the differences between GADMS-IPM(G) and the other two methods are statistically significant. In the table, the p-value of each comparison is listed in the "p-value" columns, where "+" or "-" denote that GADMS-IPM(G) obtains significantly better results or

significantly worse results, respectively.

According to Table 4, GADMS-IPM(G) obtains better classification results than the other two methods. First, GADMS-IPM(G) can generally obtain better results with the TPR, GM and F_1 score than the other two methods according to the mean values, and in 13 of the 24 cases, the differences in these three measures are significant. This finding shows that GADMS-IPM(G) significantly improves the classification performance of the minority class instances. Second, GADMS-IPM(G) generally obtains mean TNR values that are similar to those of GADMS-IPM(F) and GADMS-IPM(A). The statistical significance tests show that GADMS-IPM(G) obtains a significantly lower TNR value than the two benchmark methods on PAPER and a significantly lower TNR value than GADMS-IPM(A) on LATEX. However, GADMS-IPM(A) and GADMS-IPM(F) obtain substantially lower TPR values than TNR values. GADMS-IPM(G) may obtain slightly lower TNR values but obtains considerably higher TPR values than the other two methods. This result shows that GADMS-IPM(A) and GADMS-IPM(F) tend to classify more instances to the majority class due to the data imbalance. Third, GADMS-IPM(G) can obtain slightly better accuracy rates than the other two methods on the datasets, except for PAPER. For the number of selected KQCs, we discover that the three methods select similar numbers of KQCs because all of the methods reduce the number of QCs from 117, 100, 54 and 96 to fewer than 5 QCs using the four datasets. This finding implies that the three KQC selection models achieve similar performance in reducing the number of selected KQCs. Considering the classification performance and the number of selected KQCs, the results show that the GADMS-IPM(G) method (using GM in the KQC selection model) performs the best. Compared with the other two methods, this method can classify the instances of both the minority and majority classes more effectively with a similar number of KQCs.

Table 4. Results (%) of the classification performance measures and number of selected KQCs (NO.) acquired by GADMS-IPM methods.

Dataset	Measure	GADMS-IPM(G)		GADMS-IPM(F)			GADMS-IPM(A)		
		Mean	Std.	Mean	Std.	p-value	Mean	Std.	p-value
LATEX	TPR	**72.74**	18.93	68.75	14.99	0.1202	63.27	18.75	0.0149+
	TNR	83.69	10.75	84.79	9.73	0.1270	**87.55**	8.01	0.0056-
	ACC	**80.46**	8.69	80.07	7.54	0.5845	80.32	5.61	0.8280
	GM	**76.83**	12.10	75.58	9.52	0.2761	73.15	10.52	0.1161
	F_1	**68.60**	15.57	67.25	12.50	0.2975	64.62	12.58	0.1790
	NO.	3.80	0.54	4.07	0.81	0.1219	**3.73**	0.63	0.8066
ADPN	TPR	**83.33**	26.87	75.00	28.14	0.1250	72.50	33.13	0.0654
	TNR	**83.33**	14.40	82.00	14.92	0.2891	81.16	12.97	0.1816
	ACC	**83.43**	8.24	80.18	8.86	0.0977	78.90	7.73	0.0449+

	GM	**79.90**	17.92	74.76	17.64	0.1404	70.07	25.32	0.0499+
	F_1	**72.46**	17.41	66.44	17.87	0.1951	61.81	23.19	0.0499+
	NO.	**2.13**	0.34	2.43	0.50	0.0225+	2.50	0.50	0.0010+
PAPER	TPR	**93.06**	12.74	73.06	23.64	0.0004+	65.56	25.89	0.0001+
	TNR	87.65	4.39	89.65	4.33	0.0029-	**91.13**	4.56	0.0001-
	ACC	88.11	4.34	88.19	4.54	0.9868	**88.93**	4.95	0.3341
	GM	**90.07**	7.14	78.86	18.39	0.0008+	73.08	25.50	0.0002+
	F_1	**58.39**	13.03	52.04	19.21	0.0307+	51.30	21.09	0.0856
	NO.	**2.90**	0.54	3.77	0.67	0.0003+	4.87	0.56	0.0000+
SPIRA	TPR	**72.67**	15.04	66.00	16.45	0.0361+	66.61	21.66	0.0746
	TNR	**86.20**	12.07	83.48	14.30	0.2990	84.53	10.66	0.5066
	ACC	**81.49**	8.41	77.46	9.06	0.0578	78.26	6.06	0.0342+
	GM	**78.38**	9.04	73.09	9.45	0.0043+	73.05	11.13	0.0089+
	F_1	**73.06**	10.74	66.89	11.08	0.0075+	66.17	12.88	0.0067+
	NO.	3.57	0.92	3.43	0.76	0.5257	**3.03**	0.66	0.0132-

A comparison of the classification performance between GADMS-IPM(F) and GADMS-IPM(A) reveals that GADMS-IPM(F) generally obtains slightly better TPR, GM and F_1 score results than GADMS-IPM(A) and that the accuracy rates obtained by GADMS-IPM(F) are similar to those of GADMS-IPM(A). This finding shows that the F_1 score can better combat the data imbalance problem than accuracy in establishing the KQC selection model. However, the KQC selection results of the model with the F_1 score are worse than the results of the model with GM.

5.2 Comparison between GADMS-IPM and the benchmark methods

As discussed in Section 5.1, GADMS-IPM(G) obtains the best KQC selection performance. To further validate the proposed GADMS-IPM(G), its KQC selection results are compared with those of the benchmark methods, including SFS, SBS, NSPSOFS, CMDPSOFS, NSGAII-IPM and IDMS-IPM. In Table 5, the comparison results among SFS, SBS and GADMS-IPM(G) are shown. In Table 6, the comparison results among NSPSOFS, CMDPSOFS, NSGAII-IPM, IDMS-IPM and GADMS-IPM(G) are shown. In both tables, the p-values from the Wilcoxon signed-rank test are listed, where "+" or "-" denote whether GADMS-IPM(G) obtains significantly better results than the benchmarked methods or worse results than the benchmarked methods with a significance level of 0.05.

Table 5. Results (%) of the classification performance measures and number of selected KQCs (NO.) obtained by SFS, SBS and GADMS-IPM(G).

Dataset	Measure	GADMS-IPM(G)		SFS			SBS		
		Mean	Std.	Mean	Std.	p-value	Mean	Std.	p-value

LATEX	TPR	**72.74**	18.93	49.82	17.79	0.0000+	63.93	15.23	0.0213+
	TNR	83.69	10.75	**90.70**	5.63	0.0007-	82.13	9.36	0.3274
	ACC	**80.46**	8.69	78.62	6.72	0.1002	76.75	8.72	0.0560
	GM	**76.83**	12.10	66.12	12.12	0.0003+	71.92	10.54	0.0287+
	F_1	**68.60**	15.57	56.91	14.89	0.0009+	62.17	14.14	0.0185+
	NO.	**3.80**	0.54	7.70	2.90	0.0000+	93.80	15.85	0.0000+
ADPN	TPR	**83.33**	26.87	75.00	25.00	0.1250	65.00	32.02	0.0076+
	TNR	**83.33**	14.40	78.00	20.88	0.0520	80.00	12.65	0.0775
	ACC	**83.43**	8.24	77.32	13.22	0.0340+	75.71	14.36	0.0015+
	GM	**79.90**	17.92	73.78	12.54	0.0194+	67.51	27.25	0.0123+
	F_1	**72.46**	17.41	66.05	15.38	0.1702	58.67	27.25	0.0175+
	NO.	**2.13**	0.34	5.30	1.27	0.0000+	25.80	9.87	0.0000+
PAPER	TPR	**93.06**	12.74	58.33	30.28	0.0000+	80.83	20.43	0.0010+
	TNR	87.65	4.39	84.36	7.60	0.1466	**88.88**	7.17	0.2838
	ACC	**88.11**	4.34	82.06	7.40	0.0040+	87.99	5.59	0.4607
	GM	**90.07**	7.14	65.36	25.44	0.0000+	83.68	9.58	0.0002+
	F_1	**58.39**	13.03	37.16	17.88	0.0002+	54.92	12.45	0.0238+
	NO.	**2.90**	0.54	4.00	1.79	0.0049+	18.20	9.82	0.0000+
SPIRA	TPR	72.67	15.04	**74.00**	15.62	0.7539	62.00	24.41	0.0151+
	TNR	86.20	12.07	**87.56**	8.90	0.4416	79.22	11.32	0.0283+
	ACC	81.49	8.41	**82.81**	5.38	0.2488	73.38	11.53	0.0052+
	GM	78.38	9.04	**79.72**	7.00	0.3338	66.09	23.91	0.0082+
	F_1	73.06	10.74	**74.36**	7.91	0.3429	59.81	23.42	0.0026+
	NO.	3.57	0.92	**3.50**	1.02	0.5086	51.70	16.51	0.0000+

According to Table 5, GADMS-IPM(G) generally obtains significantly better results for the classification performance measures (except for TNR) than SFS and SBS on LATEX, ADPN and PAPER. On SPIRA, GADMS-IPM(G) obtains significantly better results for the classification performance measures than SBS and obtains slightly worse results for the classification performance measures than SFS. For the TNR measure, the results of GADMS-IPM(G) are similar to the results of SFS and SBS. The results also suggest that SFS and SBS generally obtain substantially lower TPR values than TNR values. Moreover, GADMS-IPM(G) selects significantly fewer KQCs than SBS for all of the datasets and selects significantly fewer KQCs than SFS with the datasets, except for SPIRA.

The experimental results of GADMS-IPM(G) and benchmark multi-objective feature selection methods are shown in Table 6. First, compared with the benchmark methods except for IDMS-IPM, GADMS-IPM(G) generally obtains better results for the classification performance measures. Specifically, GADMS-IPM(G) generally obtains significantly higher values for TPR, GM and F_1 and similar or even higher accuracy rates for all of the datasets. For the TNR measure, although GADMS-IPM(G) obtains a significantly lower TNR value than the

benchmark methods (except IDMS-IPM) on PAPER and obtains a significantly lower TNR value than NSGAII-IPM on LATEX, the mean TNR values of GADMS-IPM(G) are not substantially lower than those of the benchmark methods on the four datasets. In comparison, these benchmark methods obtain considerably lower TPR values than GADMS-IPM(G) on the four datasets. Second, compared with IDMS-IPM, GADMS-IPM(G) obtains similar results for the classification performance measures on LATEX and PAPER. On ADPN, GADMS-IPM(G) obtains significantly better results than IDMS-IPM for all of the classification performance measures. On SPIRA, GADMS-IPM(G) obtains significantly better GM and F_1 results than IDMS-IPM. Since IDMS-IPM attempts to maximize the same GM measure employed by GADMS-IPM(G), the results show that adopting the GM measure can improve the classification performance on the unbalanced production data. Moreover, as affected by the search performance, the obtained solutions of IDMS-IPM are inferior to those of GADMS-IPM(G), which yields worse classification results of IDMS-IPM than GADMS-IPM(G) on ADPN and SPIRA. A detailed comparison of the search performance of the multi-objective optimization methods is shown in Section 6. Third, GADMS-IPM(G) is effective in reducing the number of selected KQCs. GADMS-IPM(G) actually selects the fewest KQCs on the four datasets. Specifically, in 13 of the 16 cases, GADMS-IPM(G) selects significantly fewer KQCs. Last, the results suggest that GADMS-IPM(G) tends to select a more stable number of KQCs, as the standard deviations of "number of selected KQCs" of GADMS-IPM(G) are considerably smaller than those of the benchmark methods.

To sum up, GADMS-IPM(G) is more effective than the benchmark methods for KQC selection on the unbalanced production data. First, GADMS-IPM(G) can generally obtain high performance for all of the classification performance measures. In comparison, most benchmark methods generally obtain significantly lower TPR values than GADMS-IPM(G) due to the data imbalance. This conclusion is similar to that in Section 5.1; that is, the GM is a more suitable QC importance measure than accuracy for the unbalanced production data. Second, GADMS-IPM(G) is very effective in reducing the number of selected KQCs and can select a more stable number of KQCs over different experimental runs.

Table 6. Results (%) of the classification performance measures and number of selected KQCs

(NO.) obtained by NSPSOFS, CMDPSOFS, NSGAII-IPM, IDMS-IPM and GADMS-IPM(G).

Dataset	Measure	GADMS-IPM(G)		NSPSOFS			CMDPSOFS			NSGAII-IPM			IDMS-IPM		
		Mean	Std.	Mean	Std.	p-value	Mean	Std.	p-value	Mean	Std.	p-value	Mean	Std.	p-value
LATEX	TPR	72.74	18.93	53.98	19.38	0.0006+	60.44	23.53	0.0093+	64.11	17.91	0.0180+	**74.23**	16.40	0.5847
	TNR	83.69	10.75	86.50	9.22	0.1385	86.42	8.48	0.1460	**87.43**	8.48	0.0463-	82.85	8.33	0.3390
	ACC	80.46	8.69	76.89	7.31	0.0466+	78.73	8.42	0.1523	**80.56**	6.96	0.8972	80.31	7.29	0.8656
	GM	76.83	12.10	66.88	11.65	0.0024+	70.41	15.20	0.0720	73.86	10.52	0.1128	**77.77**	9.63	0.5666
	F_1	68.60	15.57	57.03	14.25	0.0036+	61.17	18.67	0.0634	65.52	13.06	0.2027	**68.91**	11.96	1.0000
	NO.	**3.80**	0.54	4.53	1.56	0.0250+	7.17	4.18	0.0001+	5.63	1.60	0.0000+	31.93	5.50	0.0000+
ADPN	TPR	**83.33**	26.87	53.06	37.73	0.0002+	67.89	24.60	0.0063+	67.61	31.62	0.0107+	68.06	31.33	0.0198+
	TNR	83.33	14.40	**84.78**	11.95	0.7058	83.32	16.83	0.8458	83.73	12.19	0.9010	79.42	13.57	0.0371+
	ACC	**83.43**	8.24	76.01	11.64	0.0073+	79.09	11.81	0.0562	79.23	12.56	0.1494	76.42	9.99	0.0016+
	GM	**79.90**	17.92	55.84	35.37	0.0005+	71.99	14.83	0.0191+	70.23	25.51	0.0473+	67.91	23.88	0.0060+
	F_1	**72.46**	17.41	49.17	32.49	0.0005+	64.47	15.70	0.0351+	62.45	25.69	0.0727	58.37	22.84	0.0029+
	NO.	**2.13**	0.34	4.53	2.22	0.0000+	3.90	1.68	0.0000+	2.90	0.75	0.0000+	14.13	4.78	0.0000+
PAPER	TPR	**93.06**	12.74	67.31	21.18	0.0000+	65.83	28.17	0.0002+	67.85	26.83	0.0002+	90.83	16.58	0.6875
	TNR	87.65	4.39	91.79	4.35	0.0001-	90.56	4.83	0.0004-	**92.39**	4.49	0.0000-	86.61	6.04	0.2295
	ACC	88.11	4.34	89.70	4.29	0.0569	88.40	4.64	0.7845	**90.25**	4.09	0.0054	86.97	5.76	0.1027
	GM	**90.07**	7.14	76.21	17.75	0.0000+	72.52	25.96	0.0009+	75.51	23.06	0.0018+	88.18	9.68	0.1548
	F_1	**58.39**	13.03	53.40	17.22	0.1503	49.28	21.49	0.0342+	54.03	19.55	0.2843	56.12	15.35	0.2701
	NO.	**2.90**	0.54	6.67	1.99	0.0000+	5.37	1.45	0.0000+	5.23	1.12	0.0000+	3.17	0.82	0.1396
SPIRA	TPR	**72.67**	15.04	54.67	22.02	0.0002+	61.00	21.35	0.0067+	66.44	20.33	0.0649	69.00	18.32	0.1965
	TNR	**86.20**	12.07	82.37	12.79	0.1047	85.90	10.71	0.9811	81.33	12.82	0.0441+	80.11	12.29	0.0713
	ACC	**81.49**	8.41	72.79	9.37	0.0018+	77.29	8.50	0.0861	76.14	7.98	0.0045+	76.30	8.85	0.0664
	GM	**78.38**	9.04	63.31	19.74	0.0002+	70.36	13.63	0.0080+	71.01	15.31	0.0057+	73.29	9.94	0.0324+
	F_1	**73.06**	10.74	56.10	19.70	0.0003+	63.33	16.28	0.0056+	64.62	15.16	0.0045+	66.38	12.31	0.0258+
	NO.	**3.57**	0.92	4.90	3.36	0.0271+	3.83	1.65	0.6214	3.97	1.35	0.1647	18.17	5.38	0.0000+

5.3 Computational time

Table 7 lists the average CPU time (seconds) of each method over the 30 runs on each dataset. First, the results show that SFS consumes substantially less computational time than the GADMS-IPM methods because the greedy search strategy used by SFS requires fewer searching steps than those of GADMS-IPM and the evaluations are based on a very small number of features due to forward selection. SBS generally requires more computational time than other methods because SBS searches from the QC set with all QCs and sequentially eliminates QCs to obtain the best QC subset, and the large number of QCs consumes a vast amount of time to perform each wrapper evaluation that requires a training process of the classification algorithm. Second, the three GADMS-IPM methods generally require slightly less computational time than the benchmark multi-objective feature selection methods. This finding suggests that the caching strategy of the GADMS-IPM methods for improving the time

efficiency is effective. Note that searching for the objective function values in the cache is also time consuming. This strategy may even improve the computational time. Theoretically, the larger the number of generations is, the more likely it is that we can reduce the computational time with this strategy. In general, the computational time results of the multi-objective feature selection methods are similar since these methods are designed to stop with the same number of function evaluations, and the inner 5-fold CV adopted in each evaluation is time consuming.

Table 7. CPU computational time (seconds) of each method.

Dataset	GADMS-IPM(G)	GADMS-IPM(F)	GADMS-IPM(A)	NSPSOFS	CMDPSOFS	NSGAII-IPM	IDMS-IPM	SFS	SBS
LATEX	434	453	420	520	537	579	668	**81**	2158
ADPN	79	77	71	94	89	88	89	**12**	377
PAPER	291	301	289	521	502	441	344	**12**	499
SPIRA	156	163	148	196	181	201	222	**12**	870
Average	240	249	232	333	327	327	331	**29**	976

6. Further comparison of search abilities between GADMS and benchmark methods

The proposed GADMS-IPM is composed of two phases, i.e., the use of the multi-objective optimization method GADMS to obtain a set of candidate solutions and the use of IPM to obtain a final solution. Similarly, the four benchmark methods, NSPSOFS, CMDPSOFS, NSGAII-IPM and IDMS-IPM, also adopt the multi-objective optimization methods, i.e., NSPSO, CMDPSO, MNSGAII and IDMS, in the first phase of KQC selection. The optimization performance of these multi-objective optimization methods determines if the candidate solutions for the second phase are sufficient, and this determination has a substantial impact on the final KQC selection results. In this section, the search performance of GADMS is compared with that of NSPSO, CMDPSO, MNSGAII and IDMS. Since the KQC selection model with GM performs the best on the unbalanced production data, this model is utilized for comparisons. We further collect the optimization results of NSPSO, CMDPSO and MNSGAII with Model-GM and compare the results with that of GADMS (results of the first phase of GADMS-IPM(G)). For IDMS, the search results of the first phase of IDMS-IPM are directly employed for comparisons since Model-GM has already been used by IDMS-IPM in the experiments in Section 5. The values of the two objective functions, i.e., GM and the number of selected QCs, are used to compare the search results.

6.1 Comparison metrics

To validate the optimization performance of the proposed GADMS method, we conduct two sets of comparisons in this section. First, three commonly employed quality metrics of multi-objective evolutionary algorithms are used to compare the final optimization results of the methods. These metrics include the inverted generational distance (IGD) [13], hypervolume (HV) [46], and the diversity metric (DM) proposed by Deb et al. [12]. Second, we adopt the convergence distance (CD) metric proposed in [28] to construct the convergence curves and analyze the convergence properties of the optimization methods. The CD metric calculates the distance between solutions during iterations and the Pareto solutions in the objective space. The details of these metrics are briefly introduced as follows.

IGD measures the distance between the true Pareto solutions and the obtained nondominated set in the objective space. Specifically, the Euclidean distance to the closest solution in the nondominated set for each Pareto solution is calculated, and the average of the distances is taken as the IGD value. Let PF be the set of Pareto solutions and S be the set of obtained nondominated solutions. The IGD value for S is calculated as

$$IGD(S, PF) = \frac{1}{|PF|} \sum_{p \in PF} \min_{s \in S} d(p, s), \tag{12}$$

where $||$ denotes the size of a set and $d(p, s)$ denotes the Euclidean distance between p and s in the objective space. The lower the IGD value is, the better the obtained nondominated set S is.

HV measures the hypervolume that is dominated by the obtained nondominated set in the objective space. Let S be the set of obtained nondominated solutions and r be the reference point. The HV value for S is calculated as

$$HV(S, r) = volume\left(\bigcup_{s \in S} \prod_{m=1}^{M} |f_m^s - f_m^r|\right), \tag{13}$$

where f_m^s denotes the value of the mth objective function for solution s and M is the number of objectives. Given the reference point r, the larger the HV value is, the better the nondominated set S is.

DM measures the spread degree of the nondominated set S obtained by a method. Assume that the nondominated set S contains $|S|$ solutions, $d_1, \ldots, d_{|S|-1}$ are the Euclidean

distances between the consecutive solutions in the objective space, and d_f and d_l are the Euclidean distances between the two extreme solutions and their nearest solutions in the obtained nondominated set. The DM value for S is calculated as

$$DM(S) = \frac{d_f + d_l + \sum_{i=1}^{|S|-1} \left| d_i - \overline{d} \right|}{d_f + d_l + (|S| - 1)\overline{d}}, \tag{14}$$

where $\overline{d}$ is the average of $d_1, \dots, d_{|S|-1}$. A lower DM value denotes a higher degree of the solution diversity.

To obtain the IGD value, we need to know the true Pareto front. However, the true Pareto front of the feature selection problem in this paper is unknown. Therefore, for each fold, we obtain the approximate Pareto front from the union of solutions obtained by all of the five compared methods. For the HV metric, the f_m^r $(m = 1, \dots, M)$ of the reference point is defined as $1.1 * \max_{s \in S^U} f_m^s$ (S^U is the union of the obtained solutions of the five methods), as suggested by Yuan et al. [46]. Moreover, as the scales of the objective functions can be very different, before calculating IGD, HV and DM, we normalize each objective function f_m $(m = 1, \dots, M)$ for solutions obtained by each method using the min-max normalization method, where the minimum and maximum values are obtained from the union of all of the obtained solutions of the five methods [46]. Based on these settings, we can obtain the IGD, HV and DM values of the obtained nondominated solutions on each run to compare the search performance of the methods.

CD measures the distance between a set of obtained solutions and the Pareto solutions in the objective space. Let PF be the set of Pareto solutions and S be a set of solutions. The CD value of S is calculated as

$$CD(S, PF) = \left(\frac{1}{|PF|} \sum_{p \in PF} \min_{s \in S} d(p, s) + \frac{1}{|S|} \sum_{s \in S} \min_{p \in PF} d(s, p) \right) * 0.5\,, \tag{15}$$

where $||$ denotes the size of a set and $d(p, s)$ denotes the Euclidean distance between p and s in the objective space. Because the first part in the parentheses is the definition of the IGD and the second part in the parentheses is the definition of the generational distance (GD), CD is the average of the IGD and GD. In the experiments, the solutions obtained at each generation are recorded for each method; thus, the CD value at each generation can be calculated. However,

the number of function evaluations of GADMS varies at different generations since the DMS process may be conducted after the GA process of GADMS, which hinders a comparison with the benchmark methods. Therefore, to fairly compare GADMS with the benchmark methods, the CD value at each number of function evaluations (calculated from the CD value at each generation) is used to construct the convergence curve figures of "number of function evaluations" vs. "distance to the Pareto solutions", as shown in [28]. Similar to the IGD, HV and DM metrics, the min-max normalization method is employed to normalize the scales of the two objective functions before calculating the CD value. Moreover, since 30 runs of experiments have been conducted on each dataset, the average convergence curve over the 30 runs is drawn for each method on each dataset.

6.2 Comparison of final search results

Table 8 shows the obtained IGD, HV and DM values of GADMS, NSPSO, CMDPSO, MNSGAII and IDMS. In the table, "Mean" and "Std." indicate the mean and standard deviation of IGD, HV and DM of each method over the 30 runs. The Wilcoxon signed-rank test [38] with a significance level of 0.05 is used to compare the results of IGD, HV and DM between GADMS and each benchmark method, where each "+" or "-" in the "p-value" columns indicates that GADMS obtains a significantly better or worse result. Moreover, we highlight (in bold) the best IGD, HV and DM results for each dataset.

Table 8. IGD, HV and DM values obtained by each method.

Metric	Dataset	GADMS		NSPSO			CMDPSO			MNSGAII			IDMS		
		Mean	Std.	Mean	Std.	p-value	Mean	Std.	p-value	Mean	Std.	p-value	Mean	Std.	p-value
IGD	LATEX	**0.0170**	0.0153	0.0746	0.0339	0.0000+	0.0763	0.0379	0.0000+	0.0362	0.0155	0.0002+	0.4616	0.0914	0.0000+
	ADPN	**0.0216**	0.0184	0.0524	0.0225	0.0000+	0.0756	0.0332	0.0000+	0.0440	0.0188	0.0001+	0.3316	0.1553	0.0000+
	PAPER	0.0203	0.0220	0.0413	0.0161	0.0000+	0.0366	0.0193	0.0032+	**0.0135**	0.0098	0.3493	0.0339	0.0245	0.0032+
	SPIRA	**0.0176**	0.0150	0.0646	0.0178	0.0000+	0.0639	0.0271	0.0000+	0.0346	0.0160	0.0001+	0.3082	0.1318	0.0000+
HV	LATEX	**1.1718**	0.0181	1.0544	0.0742	0.0000+	1.0286	0.0795	0.0000+	1.1061	0.0479	0.0000+	0.5025	0.1066	0.0000+
	ADPN	**1.1683**	0.0272	1.1206	0.0460	0.0000+	1.0648	0.0762	0.0000+	1.1213	0.0465	0.0000+	0.7116	0.1934	0.0000+
	PAPER	**1.1370**	0.0305	1.0617	0.0474	0.0000+	1.0717	0.0585	0.0000+	1.1302	0.0363	0.1254	1.1055	0.0381	0.0001+
	SPIRA	**1.1538**	0.0217	1.0554	0.0402	0.0000+	1.0416	0.0609	0.0000+	1.1163	0.0373	0.0001+	0.6534	0.1689	0.0000+
DM	LATEX	0.9016	0.0207	**0.8829**	0.1037	0.6143	0.9150	0.0422	0.4165	0.8951	0.0322	0.3709	0.9008	0.0487	0.8612
	ADPN	0.9421	0.0617	**0.8682**	0.0799	0.0002-	0.8795	0.0982	0.0057-	0.9147	0.0506	0.0656	0.8950	0.0623	0.0125-
	PAPER	0.8258	0.0637	**0.7494**	0.1104	0.0013-	0.8095	0.0955	0.3600	0.7872	0.0955	0.0175-	0.8023	0.0969	0.1359
	SPIRA	0.8725	0.0373	**0.8328**	0.0704	0.0057-	0.8569	0.0767	0.2210	0.8537	0.0491	0.1020	0.8825	0.0398	0.3185

According to the IGD results, GADMS obtains the best (lowest) IGD values on LATEX, ADPN and SPIRA and the second best IGD value on PAPER (slightly higher than MNSGAII). The statistical significance test results show that GADMS obtains significantly better IGD values than NSPSO, CMDPSO and IDMS for all of the four datasets and obtains significantly better IGD values than MNSGAII on 3 (i.e., LATEX, ADPN and SPIRA) of the 4 datasets. No results show that GADMS obtains significantly worse IGD values than any of the benchmark methods. According to the HV results, we determine that GADMS obtains the best (highest) HV values on all of the datasets. Further statistical significance test results show that GADMS obtains significantly better HV values than NSPSO, CMDPSO and IDMS on all of the datasets and obtains significantly better HV values than MNSGAII on the datasets, with the exception of PAPER. According to the DM metric, NSPSO obtains the best results on all of the datasets, and compared with the benchmark methods, GADMS does not show competent results. This finding shows that the obtained solutions of GADMS are not perfectly spreading on its nondominated front. GADMS obtains significantly better HV and IGD results than the benchmark methods in most cases and obtains slightly worse DM results than the benchmark methods. These results imply that the proposed GADMS method can obtain better solutions that are closer to the Pareto front than the benchmark methods, while strategies to improve the solution spreading property of GADMS are worth investigating.

6.3 Comparison of the convergence properties

The convergence curves using the datasets for each method are shown in Fig. 5, where the y-axis is the distance value measured by the CD metric and the x-axis is the number of function evaluations. First, a comparison of GADMS with NSPSO and CMDPSO indicates that GADMS converges slower in the early evolving stage, as generally in the first 2,000 function evaluations, the convergence curves of GADMS are higher than those of NSPSO and CMDPSO. However, GADMS gradually obtains lower convergence curves than NSPSO and CMDPSO with an increase in the number of function evaluations. This finding indicates a better global search performance of GADMS than NSPSO and CMDPSO. Second, a comparison of GADMS with MNSGAII reveals that GADMS obtains lower convergence curves than MNSGAII on all of the four datasets, which shows that GADMS bears better convergence performance. Specifically, the convergence curves of GADMS can decrease substantially faster than MNSGAII, which shows that GADMS bears a better

convergence speed. Third, compared with IDMS, GADMS obtains significantly better convergence curves. The convergence curves show that IDMS converges slower than GADMS. Moreover, with the given number of function evaluations in the experiments, the convergence curves show that IDMS does not reach the convergence status, which explains why even when the same KQC selection model is employed, GADMS-IPM(G) can obtain better KQC selection results than IDMS-IPM. Thus, we discover that combining all of the genetic operators with DMS (i.e., GADMS) is more effective than just combining the mutation operator of GAs with DMS (i.e., IDMS). Fourth, the MOPSO algorithms, NSPSO and CMDPSO, converge faster than the other three methods, as the convergence curves of the two MOPSO algorithms can quickly decrease to a very low level. The two MOPSO algorithms may face a premature convergence problem, since GADMS and MNSGAII gradually obtain lower levels of convergence curves than the two MOPSO algorithms with the iterations. To sum up, the proposed GADMS bears both a suitable convergence speed and convergence performance, which shows that it is an effective multi-objective optimization method for the KQC selection problem.

The results in this section and Section 6.2 illustrate the adequate search performance of the proposed GADMS method. Three reasons can explain the effectiveness of GADMS. First, since GADMS adopts all of the genetic operators in GAs to evolve new solutions, the excellent global search performance of GAs can be inherited by GADMS. Second, as a local search strategy, the DMS process of GADMS can effectively tune the current nondominated solutions by performing a local search around them. Thus, the DMS process is a reasonable supplemental solution updating mechanism in addition to the GA process to improve the search performance. Third, the DMS process focuses on updating the current nondominated solutions instead of updating all solutions. Thus, the convergence speed of GADMS can be improved.

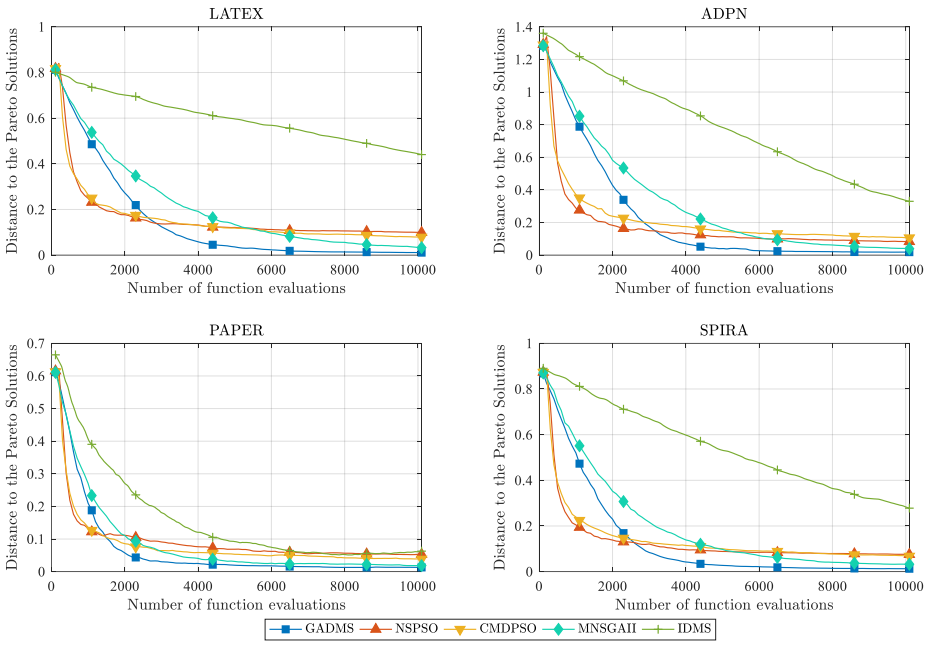

Fig. 5. Convergence curves obtained by GADMS and benchmark methods.

7. Further analysis on the synthetic datasets

The above results have shown that the proposed method is effective in KQC selection on the four unbalanced datasets collected from production processes. In this section, we further conduct two new sets of experiments using several synthetic datasets to test if the proposed method still performs effectively when the imbalance ratio of data increases and the number of noisy features increases. First, the imbalance ratios of the majority class instances to the minority class instances of LATEX, ADPN and SPIRA are approximately 2, which indicates that these three datasets are only slightly unbalanced. To further validate the effectiveness of the proposed method on unbalanced data, we increase the imbalance ratios of these three datasets. Three synthetic datasets denoted by LATEX-U, ADPN-U and SPIRA-U with higher imbalance ratios are generated based on LATEX, ADPN and SPIRA. For each synthetic dataset, we upsampled the instances of the majority class by reproducing these instances 2 times. In the experiments with the three synthetic datasets, the three GADMS-IPM variants (GADMS-IPM(G), GADMS-IPM(F) and GADMS-IPM(A)) are utilized, and the parameter settings and experimental configurations are the same as those introduced in Section 4. Since the imbalance ratios of the three synthetic datasets are significantly increased compared with the original

datasets, we adopt the modified NB classifier as that employed by PAPER in Section 4. Second, we construct four synthetic datasets denoted by LATEX-F, ADPN-F, PAPER-F and SPIRA-F based on LATEX, ADPN, PAPER and SPIRA. For each of the four synthetic datasets, N (number of QCs in the original dataset) noisy QCs from the standard normal distribution have been added to the original dataset, which doubles the number of QCs in the dataset. The GADMS-IPM(G) method is adopted on the four synthetic datasets with more noisy QCs; the results are compared with those from the original datasets. The parameter settings and experimental configurations on the four synthetic datasets are the same as those on the original datasets, as introduced in Section 4. Table 9 shows details of the synthetic datasets in the two sets of experiments.

Table 9. Details of the synthetic datasets.

Dataset	Number of instances	Number of QCs	Number of the minority class instances (premium products)	Number of the majority class instances (regular products)
LATEX-U	630	117	78	552
ADPN-U	173	100	20	153
SPIRA-U	335	96	50	285
LATEX-F	262	234	78	184
ADPN-F	71	200	20	51
PAPER-F	384	108	33	351
SPIRA-F	145	192	50	95

7.1 Results on the synthetic datasets with increased imbalance ratios

Table 10 shows the KQC selection results of the three GADMS-IPM variants for LATEX-U, ADPN-U and SPIRA-U. First, according to the mean values of the classification performance measures over the 30 runs, GADMS-IPM(G) obtains the highest values of TPR, GM and F_1 on the three datasets, which indicates that GADMS-IPM(G) better handles the data imbalance problem than GADMS-IPM(F) and GADMS-IPM(A) for the three synthetic datasets. According to the statistical significance test results, GADMS-IPM(G) obtains significantly better TPR, GM and F_1 results than both GADMS-IPM(F) and GADMS-IPM(A) on LATEX-U and obtains significantly better TPR, GM and F_1 results than GADMS-IPM(F) on SPIRA-U. Second, according to the number of selected KQCs, GADMS-IPM(G) selects fewer KQCs than the other two methods on the three datasets, and the statistical significance test results indicate that the obtained numbers of KQCs by GADMS-IPM(G) are significantly smaller than those of GADMS-IPM(F) and GADMS-IPM(A) in 5 of the 6 cases. To sum up, for the three synthetic

datasets with increased imbalance ratios, GADMS-IPM(G) performs the best in handling the data imbalance problem since compared with the other two methods, GADMS-IPM(G) obtains substantially better classification performance on the minority class instances with similar overall classification performance. Moreover, GADMS-IPM(G) performs more effectively in reducing the number of KQCs. These results are consistent with those in Section 5, which indicates that GADMS-IPM(G) with the GM measure is the most effective KQC selection method among the three GADMS-IPM variants.

Table 10. Results (%) of the classification performance measures and number of selected KQCs (NO.) on the synthetic datasets with increased imbalance ratios from GADMS-IPM(G), GADMS-IPM(F) and GADMS-IPM(A).

Dataset	Measure	GADMS-IPM(G)		GADMS-IPM(F)			GADMS-IPM(A)		
		Mean	Std.	Mean	Std.	p-value	Mean	Std.	p-value
LATEX-U	TPR	**82.50**	13.87	68.15	14.51	0.0015+	43.51	14.54	0.0000+
	TNR	81.52	5.29	84.06	4.61	0.0153-	**92.66**	4.64	0.0000-
	ACC	81.64	4.88	82.06	4.18	0.2732	**86.59**	3.90	0.0008-
	GM	**81.63**	7.62	75.14	8.57	0.0031+	62.47	10.17	0.0000+
	F_1	**52.97**	8.78	48.59	8.92	0.0427+	44.43	11.10	0.0008+
	NO.	**5.10**	1.51	7.07	1.46	0.0001+	6.73	3.03	0.0396+
ADPN-U	TPR	**78.89**	27.19	73.33	35.90	0.2891	75.83	37.35	0.4609
	TNR	89.94	10.12	**92.22**	7.23	0.3213	89.93	9.45	0.8208
	ACC	88.69	9.71	**90.11**	8.34	0.2367	88.35	8.67	0.5773
	GM	**81.73**	20.65	76.04	32.34	0.4345	74.30	33.85	0.5698
	F_1	**64.95**	26.03	63.39	30.68	0.7221	57.50	29.37	0.1675
	NO.	**3.13**	0.67	3.47	0.56	0.0253+	3.60	0.80	0.0046+
SPIRA-U	TPR	**74.67**	17.07	59.33	19.65	0.0004+	72.22	14.56	0.3672
	TNR	82.22	6.03	**84.95**	7.13	0.0709	81.54	5.40	0.8819
	ACC	81.08	5.51	**81.11**	6.55	0.8862	80.14	4.85	0.8307
	GM	**77.69**	9.68	69.79	12.61	0.0022+	76.21	8.19	0.3219
	F_1	**54.18**	10.85	48.50	13.96	0.0464+	52.16	9.47	0.2793
	NO.	**4.73**	0.77	5.90	1.01	0.0008+	5.13	0.85	0.0901

7.2 Results on the synthetic datasets with an increased number of noisy QCs

Table 11 shows the KQC selection results obtained by GADMS-IPM(G) using the original datasets and synthetic datasets with an increased number of noisy QCs, where "+" or "-" indicate that the results of the original datasets are significantly better or significantly worse than those of the synthetic datasets with a significance level of 0.05. First, the classification performance results of TPR, TNR, ACC, GM and F_1 do not have significant differences

between LATEX and LATEX-F, ADPN and ADPN-F, PAPER and PAPER-F according to the statistical significance tests. This outcome shows that the increase in noisy QCs for LATEX, ADPN and PAPER does not have a substantial impact on the classification performance of GADMS-IPM(G). However, the obtained TNR, ACC and F_1 results from SPIRA-F are significantly worse than those from SPIRA, which indicates that the increase in noisy QCs for the SPIRA dataset affects the final quality of the selected KQCs. Second, according to the number of selected KQCs, GADMS-IPM(G) selects a significantly larger number of KQCs on the four synthetic datasets than that on the original datasets. Even though the difference of the number of selected KQCs is significant for each comparison, GADMS-IPM(G) only selects slightly more KQCs on the synthetic datasets than the original datasets except for LATEX. Different from other comparisons, the number of KQCs obtained on LATEX-F is considerably larger than that obtained on LATEX. The large number of QCs on LATEX-F produces a very large solution space that requires additional computational resources for the optimization methods to reduce more irrelevant or redundant QCs.

GADMS-IPM(G) is effective in KQC selection since it obtains KQC selection results on the synthetic datasets that are similar to those on the original datasets. However, the performance of GADMS-IPM(G) is affected by the increase in the number of QCs with current parameter settings for GADMS-IPM(G). To improve the final KQC selection results using the datasets with a larger number of QCs, a larger population size or number of generations is required. Moreover, it could be also helpful for improving the performance of GADMS-IPM(G) to combine a feature weighting method to guide the population initialization or to filter some irrelevant QCs to narrow the search space, which is worthy of future studies.

Table 11. Results (%) of the classification performance measures and number of selected KQCs (NO.) on the original datasets and synthetic datasets with additional noisy QCs from GADMS-IPM(G).

Measure	Dataset	Mean	Std.	Dataset	Mean	Std.	p-value
TPR		72.74	18.93		**74.79**	17.76	0.4373
TNR		**83.69**	10.75		82.59	8.48	0.4052
ACC	LATEX	**80.46**	8.69	LATEX-F	80.35	7.70	0.7848
GM		76.83	12.10		**77.82**	10.71	0.5165
F_1		68.60	15.57		**68.90**	13.33	0.8382

NO.		**3.80**	0.54		11.73	2.57	0.0000+
TPR	ADPN	**83.33**	26.87	ADPN-F	81.67	27.34	1.0000
TNR		83.33	14.40		**84.67**	15.00	0.4824
ACC		83.43	8.24		**83.95**	9.80	0.7129
GM		**79.90**	17.92		79.80	18.95	0.9343
F_1		72.46	17.41		**72.78**	19.87	0.8276
NO.		**2.13**	0.34		3.37	1.54	0.0004+
TPR	PAPER	**93.06**	12.74	PAPER-F	91.67	15.81	1.0000
TNR		87.65	4.39		**88.32**	4.55	0.4092
ACC		88.11	4.34		**88.55**	4.28	0.3092
GM		**90.07**	7.14		89.53	8.60	0.4920
F_1		58.39	13.03		**58.99**	14.33	0.9622
NO.		**2.90**	0.54		3.57	0.67	0.0013+
TPR	SPIRA	**72.67**	15.04	SPIRA-F	71.33	21.09	0.5777
TNR		**86.20**	12.07		80.26	11.49	0.0079+
ACC		**81.49**	8.41		77.19	8.64	0.0273+
GM		**78.38**	9.04		74.23	11.86	0.1169
F_1		**73.06**	10.74		67.35	14.13	0.0485+
NO.		**3.57**	0.92		6.23	2.30	0.0000+

8. Conclusions and future work

Selecting KQCs related to product quality is essential for product quality improvement and control. Generally, production datasets collected from production lines are unbalanced since the number of products of different quality levels differs considerably. To select KQCs based on unbalanced production data, this paper proposes a multi-objective feature selection approach named GADMS-IPM. First, the KQC selection task is defined as a bi-objective feature selection problem of maximizing the classification performance and minimizing the QC subset size. Three candidate KQC selection models are obtained based on the bi-objective problem, where the classification performance is measured by GM, F_1 score or accuracy. We propose a hybrid multi-objective optimization method named GADMS, which combines the GA search strategy and the local search strategy DMS, to search for a set of nondominated solutions for the defined models. Last, we adopt the IPM to select the final solution (KQC set) from the solutions obtained by GADMS.

The proposed methods are tested on four unbalanced production datasets. The results show that the KQC selection model using the GM measure obtains the best results since it significantly improves the classification performance of the minority class instances (premium

products) without an obvious decrease in the classification performance of the majority class instances (regular products). The GADMS-IPM method with GM is also compared with two conventional feature selection methods, SFS and SBS, and four multi-objective feature selection methods, NSGAII-IPM, NSPSOFS, CMDPSOFS and IDMS-IPM. The results show that the GADMS-IPM method obtains better classification results with only a few KQCs, while the benchmark methods are generally affected by the data imbalance because they obtain a significantly lower classification performance for the minority class than that for the majority class. Moreover, the search ability of GADMS is compared with that of the benchmark multi-objective optimization methods, i.e., NSPSO, CMDPSO, MNSGAII and IDMS. The results show that GADMS obtains better search results and a better convergence property than the benchmark methods. Moreover, we find that although the proposed GADMS-IPM method requires more computational time than SFS, it requires relatively lower computational time than the benchmark multi-objective feature selection methods and SBS. This finding suggests that the caching strategy adopted in GADMS-IPM for improving the time efficiency is effective.

In practice, the quality of a product can be denoted by a continuous response variable, which entails a regression problem. Therefore, we plan to build a KQC selection method for regression tasks. Moreover, developing a filter-based QC subset importance measure to reduce the computational time of a QC importance evaluation in KQC selection is worth investigating. Furthermore, combining feature weighting strategies with the proposed wrapper-based KQC selection method to improve the efficiency is one of our research interests.

Acknowledgments

The authors would like to thank the editor and anonymous referees for the constructive comments and suggestions. This work was partly supported by the Humanities and Social Sciences Youth Fund of Ministry of Education of China, under Grant 19YJC630071; National Natural Science Foundation of China (NSFC), under Grant 71661147003; Research Project of the Tianjin Municipal Education Commission (*Research on Key Quality Characteristics Identification for Complex Products Based on Imbalanced Manufacturing Data*, Grant number 161072); Marsden Fund of New Zealand Government, under Contracts VUW1509 and VUW1615; and University Research Fund at Victoria University of Wellington, under Grant number 216378/3764.

References

[1] E. Amaldi and V. Kann, *On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems*, Theoretical Computer Science 209 (1-2) (1998) 237-260.

[2] M. Amoozegar and B. Minaei-Bidgoli, *Optimizing multi-objective PSO based feature selection method using a feature elitism mechanism*, Expert Systems with Applications 113 (2018) 499-514.

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

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

[5] H. Banka and S. Dara, *A Hamming distance based binary particle swarm optimization (HDBPSO) algorithm for high dimensional feature selection, classification and validation*, Pattern Recognition Letters 52 (2015) 94-100.

[6] P. Bermejo, J. A. Gámez, and J. M. Puerta, *Speeding up incremental wrapper feature subset selection with Naive Bayes classifier*, Knowledge-Based Systems 55 (2014) 140-147.

[7] U. Bhowan, M. Johnston, M. Zhang, and X. Yao, *Evolving diverse ensembles using genetic programming for classification with unbalanced data*, IEEE Transactions on Evolutionary Computation 17 (3) (2013) 368-386.

[8] H. Chen, T. Li, X. Fan, and C. Luo, *Feature selection for imbalanced data based on neighborhood rough sets*, Information Sciences 483 (2019) 1-20.

[9] A. L. Custódio, J. A. Madeira, A. I. F. Vaz, and L. N. Vicente, *Direct multisearch for multi-objective optimization*, SIAM Journal on Optimization 21 (3) (2011) 1109-1140.

[10] A. L. Custódio and J. F. A. Madeira, *MultiGLODS: global and local multi-objective optimization using direct search*, Journal of Global Optimization 72 (2) (2018) 323-345.

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

[12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, *A fast and elitist multi-objective genetic algorithm: NSGA-II*, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182-197.

[13] K. Deb and H. Jain, *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 (4) (2014) 577-601.

[14] M. Freimer and P. Yu, *Some new results on compromise solutions for group decision problems*, Management Science 22 (6) (1976) 688-693.

[15] J. García-Nieto, E. Alba, L. Jourdan, and E. Talbi, *Sensitivity and specificity based multi-objective approach for feature selection: Application to cancer diagnosis*, Information Processing Letters 109 (16) (2009) 887-896.

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

[17] I. Guyon and A. Elisseeff, *An introduction to variable and feature selection*, Journal of

Machine Learning Research 3 (2003) 1157-1182.

[18] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, *The WEKA data mining software: an update*, ACM SIGKDD Explorations Newsletter 11 (1) (2009) 10-18.

[19] T. M. Hamdani, J.-M. Won, A. M. Alimi, and F. Karray, *Hierarchical genetic algorithm with new evaluation function and bi-coded representation for the selection of features considering their confidence rate*, Applied Soft Computing 11 (2) (2011) 2501-2509.

[20] E. Hancer, B. Xue, and M. Zhang, *Differential evolution for filter feature selection based on information theory and feature ranking*, Knowledge-Based Systems 140 (2018) 103-119.

[21] E. Hancer, B. Xue, M. Zhang, D. Karaboga, and B. Akay, *Pareto front feature selection based on artificial bee colony optimization*, Information Sciences 422 (2018) 462-479.

[22] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, *Evolutionary many-objective optimization: A short review*, in: *IEEE Congress on Evolutionary Computation*, Citeseer, 2008, pp. 2419-2426.

[23] G. H. John and P. Langley, *Estimating continuous distributions in Bayesian classifiers*, in: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers Inc., 1995, pp. 338-345.

[24] M.-J. Kim, D.-K. Kang, and H. B. Kim, *Geometric mean based boosting algorithm with over-sampling to resolve data imbalance problem for bankruptcy prediction*, Expert Systems with Applications 42 (3) (2015) 1074-1082.

[25] R. Kohavi and G. H. John, *Wrappers for feature subset selection*, Artificial Intelligence 97 (1) (1997) 273-324.

[26] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, *An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics*, Information Sciences 250 (2013) 113-141.

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

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

[29] Z. C. Lipton, C. Elkan, and B. Naryanaswamy, *Optimal thresholding of classifiers to maximize F1 measure*, in: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2014, pp. 225-239.

[30] K. Nag and N. R. Pal, *A multi-objective genetic programming-based ensemble for simultaneous feature Selection and classification*, IEEE Transactions on Cybernetics 46 (2) (2016) 499-510.

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

[32] J. Pacheco, S. Casado, F. Angel-Bello, and A. Álvarez, *Bi-objective feature selection for discriminant analysis in two-class classification*, Knowledge-Based Systems 44 (2013) 57-64.

[33] M. Robnik-Šikonja and I. Kononenko, *Theoretical and empirical analysis of ReliefF and RReliefF*, Machine Learning 53 (1-2) (2003) 23-69.

[34] C. J. Tan, C. P. Lim, and Y. N. Cheah, *A multi-objective evolutionary algorithm-based ensemble optimizer for feature selection and classification with neural network models*, Neurocomputing 125 (2014) 217-228.

[35] W.-m. Tian, Z. He, and W. Yan, *Key process variable identification for quality classification based on PLSR model and wrapper feature selection*, in: R. Dou (Eds.), *Proceedings of 2012 3rd International Asia Conference on Industrial Engineering and Management Innovation (IEMI2012)*, Springer Berlin Heidelberg, 2013, pp. 263-270.

[36] B. Tran, B. Xue, and M. Zhang, *Variable-length particle swarm optimisation for feature selection on high-dimensional classification*, IEEE Transactions on Evolutionary Computation 23 (3) (2019) 473-487.

[37] Z. Wang, M. Li, and J. Li, *A multi-objective evolutionary algorithm for feature selection based on mutual information with a new redundancy measure*, Information Sciences 307 (2015) 73-88.

[38] F. Wilcoxon, *Solution comparisons by ranking methods*, Biometrics Bulletin 1 (6) (1945) 80-83.

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

[40] T.-T. Wong, *Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation*, Pattern Recognition 48 (9) (2015) 2839-2846.

[41] C. Wu, F. Liu, and B. Zhu, *Control chart pattern recognition using an integrated model based on binary-tree support vector machine*, International Journal of Production Research 53 (7) (2015) 2026-2040.

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

[43] B. Xue, M. Zhang, and W. N. Browne, *Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms*, Applied Soft Computing 18 (2014) 261-276.

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

[45] D. Yilmaz Eroglu and K. Kilic, *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 (2017) 18-32.

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

[47] X. Zhang, Q. Zhang, M. Chen, Y. Sun, X. Qin, and H. Li, *A two-stage feature selection and intelligent fault diagnosis method for rotating machinery using hybrid filter and wrapper method*, Neurocomputing 275 (2018) 2426-2439.

[48] Y. Zhang, D.-w. Gong, X.-z. Gao, T. Tian, and X.-y. Sun, *Binary differential evolution with self-learning for multi-objective feature selection*, Information Sciences 507 (2020) 67-85.

[49] P. Zhou, X. Hu, P. Li, and X. Wu, *Online streaming feature selection using adapted Neighborhood Rough Set*, Information Sciences 481 (2019) 258-279.