Key quality characteristics selection for imbalanced production data using a two-phase bi-objective feature selection method

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

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

[b] *Management Innovation and Evaluation Research Center, Tianjin University of Commerce, Tianjin 300134, China*

[c] *College of Management and Economics, Tianjin University, Tianjin 300072, China*

Abstract:

The selection of key quality characteristics (KQCs) that are significantly associated with product quality is essential for improving product quality. Production lines generally yield a larger number of regular products than do premium products, which creates an imbalance in production datasets and complicates KQC selection. In this study, a KQC selection method with an excellent ability to predict product quality is proposed based on a two-phase bi-objective feature selection method. The KQC selection model is established as a bi-objective problem of maximizing feature (i.e., quality characteristic) importance and minimizing percentage of selected features, and the geometric mean (G-mean) is selected as the feature importance metric for imbalanced data. To solve this model, a two-phase multi-objective optimization method is proposed; this method yields a set of candidate solutions (KQC sets) using an improved direct multi-search (DMS) strategy and uses the ideal point method (IPM) to select the final KQC sets from the candidate solutions. The experimental results indicate that the proposed method is effective for selecting KQCs for imbalanced production data.

Keywords: quality management; direct multi-search (DMS); feature selection; imbalanced data; multi-objective optimization

https://doi.org/10.1016/j.ejor.2018.10.051

[*] Corresponding author.
E-mail address: andali1989@163.com (An-Da Li) ,yzhang@tjcu.edu.cn (Yang Zhang)

1. Introduction

In modern manufacturing processes, a large number of product features and assembly features are collected from production lines, and these features determine the final quality levels (e.g., regular or premium) of products (Lee & Thornton, 1996). In this paper, these features are referred to as “quality characteristics” (QCs). However, in practice, all QCs are not equally important because certain QCs are redundant or irrelevant to the quality level (Jeong & Cho, 2006). Therefore, the selection of key quality characteristics (KQCs) that are significantly related to product quality is an essential task prior to the implementation of quality improvement tools, e.g., statistical process control (SPC) and design of experiments (DOEs). One method for identifying KQCs is the use of traditional statistical methods, such as multivariable linear regression (MLR) or analysis of variance (ANOVA); however, the high dimensionality and feature redundancy of production data can represent significant challenges for these methods (Pierre & Tuv, 2011). Feature selection, which has been employed for dimension reduction in machine learning for many years, aims to select key features with excellent predictive power for class labels (Du & Hu, 2018; Ghaddar & Naoum-Sawaya, 2018; Guyon & Elisseeff, 2003; Jain et al., 2018; Maldonado et al., 2017; Xue et al., 2016). Recent studies have introduced feature selection into KQC selection problems due to its high performance with high-dimensional data (Anzanello et al., 2009, 2012; A.-D. Li et al., 2016; Tian et al., 2013). When applying feature selection methods, each product is treated as an instance, and the QCs and the quality level correspond to the features and the class label, respectively.

Feature selection is actually an optimization problem that is usually built as a mono-objective model or bi-objective model (Bertolazzi et al., 2016). A mono-objective model defines feature selection as a problem of maximizing the accuracy of feature subsets. To solve this model, heuristic search algorithms, such as sequential forward selection (SFS) and sequential backward selection (SBS) (Gunal et al., 2009; Kohavi & John, 1997; W. Li et al., 2017), and mono-objective evolutionary algorithms (EAs), such as genetic algorithms (GAs) and particle swarm optimization (PSO), have been employed (Liang et al., 2015; Unler & Murat, 2010; Zhang et al., 2015; Zhu et al., 2007). A bi-objective model defines feature selection as a problem of maximizing the accuracy and minimizing the feature subset size. In this model, a second objective (i.e., minimizing the feature subset size) is added to improve the ability of the feature selection methods to reduce irrelevant or redundant features. To solve this model, mono-objective EAs, such as GAs and PSO (Ahila et al., 2015; C.-L. Huang & Wang, 2006; Wang et al., 2010; Welikala et al., 2015; Xue et al., 2014), or multi-objective EAs, such as niched Pareto GA (NPGA), nondominated sorting genetic algorithm (NSGA), nondominated sorting genetic algorithm II (NSGA-II) and multi-objective PSO (MOPSO) (Ahuja & Ratnoo, 2015; de Almeida Ribeiro et al., 2015; Emmanouilidis et al., 2000; A.-D. Li et al., 2016; Oliveira et al., 2006; Xue et al., 2013), are employed. Compared with mono-objective EAs, multi-objective EAs do not need to transform the multi-objective optimization (MOO) problem into a mono-objective problem, which requires competent domain knowledge. Therefore, multi-objective EAs are more suitable for the bi-objective feature selection model (Abido, 2006; de La Iglesia, 2013).

The previously mentioned feature selection methods adopt accuracy to measure the importance of a feature subset. However, accuracy is a biased measure of imbalanced datasets (López et al., 2013; Wasikowski & Chen, 2010). For example, let D be a dataset with a set F of features that includes 10 positive instances and 90 negative instances. If a classifier classifies all 100 instances into the negative class with the feature subset $FS \subseteq F$, then the accuracy rate will be 90%, and the Type I error will be 0%. However, the classifier misclassifies all positive instances, which produces a 100% Type II error. Although the accuracy rate is very high, we cannot conclude that the feature subset FS is effective. Therefore, metrics that can comprehensively measure the ability of a feature subset to classify the products in both the positive class and the negative class are needed. Recently, the metrics sensitivity and specificity or Type I and Type II errors have been adopted instead of accuracy to measure the importance of a feature subset (García-Nieto et al., 2009; Nag & Pal, 2016; Pacheco et al., 2013; Tan et al., 2014). Sensitivity and specificity (or Type I and Type II errors) measure the classification performance of positive instances and negative instances, respectively; thus, feature subsets with excellent classification ability are identified in the case of imbalanced data. However, building a feature selection model using sensitivity and specificity (or Type I and Type II errors) enables an additional objective function to be optimized, which represents a challenge for MOO methods because the search ability of these methods substantially deteriorates as the number of objectives increases (Hughes, 2005; Ishibuchi et al., 2008).

Generally, a production dataset is imbalanced because a majority of products are regular, and a minority of products are premium (or a majority of products are conforming and a minority of products are nonconforming). However, most existing KQC selection applications with feature selection

(Anzanello et al., 2009; A.-D. Li et al., 2016; Tian et al., 2013) only adopt accuracy to measure QC importance, which can lead to biased KQC selection results in the cases of imbalanced data. Therefore, effective feature selection algorithms that can handle imbalanced data are required for KQC selection problems.

To overcome the previously mentioned limitations, we build a bi-objective feature selection model for KQC selection using imbalanced production data. Compared with the previously mentioned feature selection models, this model maximizes the geometric mean (G-mean) of sensitivity and specificity and minimizes the feature subset size. The adoption of the G-mean has two advantages: (a) the feature importance of imbalanced data can be effectively measured because a low value of either sensitivity or specificity produces a low value of accuracy; and (b) using the G-mean as the objective function causes the model to become a bi-objective problem, which is beneficial for optimization methods. To solve this model, we propose a two-phase optimization method. First, we propose an improved direct multi-search (IDMS), which is an MOO method, to obtain a set of candidate solutions. The IDMS is established by embedding a two-step mutation-based strategy (i.e., mutation-poll step) in the direct multi-search (DMS) method (Custódio et al., 2011) to improve the global search ability. Second, the ideal point method (IPM) (Freimer & Yu, 1976) is employed to select the final solutions (KQC sets).

The remainder of this paper is organized as follows. Section 2 describes the KQC selection model. Section 3 proposes a two-phase optimization method to solve the proposed KQC selection model. Section 4 provides the experimental results and discussion. Section 5 presents the conclusions and implications for future studies.

2. KQC selection model

2.1 Feature (QC) importance metrics

In feature selection methods based on the wrapper model (Kohavi & John, 1997), accuracy is generally employed to measure the importance of feature subsets. For imbalanced data, however, accuracy is a biased measure, as stated in the introduction. In this paper, the classification metric G-mean (Kubat & Matwin, 1997; López et al., 2013) for imbalanced data is introduced as the feature importance metric.

The confusion matrix for two-class classification is given in Table 1, where “TP”, “TN”, “FP” and “FN” denote the numbers of true positive, true negative, false positive, and false negative instances, respectively. Based on Table 1, the G-mean can be obtained as indicated in Eq. (1), where the sensitivity and specificity are defined as indicated in Eqs. (2) and (3). As shown in Eq. (1), the G-mean is a classification metric calculated as the geometric mean of sensitivity and specificity, which measure the classification performance for the positive and negative classes, respectively. The G-mean offers significant advantages for imbalanced data compared with accuracy, and the metric can better reflect the classification performance of imbalanced data because a low predictive performance of either positive or negative instances can distinctly decrease the value of the G-mean.

$$\text{G-mean} = \sqrt{\text{Sensitivity} \times \text{Specificity}} \tag{1}$$

$$\text{Sensitivity} = 1 - \text{Type II error} = \frac{TP}{TP + FN} \tag{2}$$

$$\text{Specificity} = 1 - \text{Type I error} = \frac{TN}{TN + FP} \tag{3}$$

Table 1. Confusion Matrix.

		Predicted Class	
		Positive	Negative
True	Positive	True Positive (TP)	False Negative (FN)
Class	Negative	False Positive (FP)	True Negative (TN)

2.2 Proposed model

Assume that a production dataset D has M products (instances) and that each product has N QCs (features) $QS = \{Q_1, Q_2, ..., Q_N\}$ and one quality level (class label) $C \in \{1, -1\}$. In this paper, the quality level (i.e., regular) of the majority of instances is denoted by "-1", and the quality level (i.e., premium) of the minority of instances is denoted by "1". The observation of the ith ($i = 1, ..., M$) product is denoted by $\boldsymbol{E}_i = (e_{i,1}, ..., e_{i,N}, c_i)$, where $e_{i,1}, ..., e_{i,N}$ represents the values of $Q_1, Q_2, ..., Q_N$, and c_i represents the value of C. With the dataset, the KQCs can be identified via feature selection methods.

In this paper, the KQC selection model is constructed to select the KQCs that are strongly related to product quality and eliminate as many redundant or irrelevant QCs as possible. The KQC selection model is constructed as a bi-objective feature selection model as follows:

$$\min\ f_1(X) = 1 - \text{G-mean}(X)\,, \tag{4}$$

$$\min f_2(X) = \frac{\#X}{N}\,, \tag{5}$$

$$\text{s.t. } X \subseteq QS, X \neq \varnothing\,, \tag{6}$$

where X denotes a nonempty feature subset of QS, $\#X$ denotes the size of X (the number of features in X), and N denotes the number of original features. In the model, the G-mean is employed to measure feature importance: the first objective is defined as $1-\text{G-mean}$, which is equivalent to maximizing the G-mean, and the second objective is defined as minimizing the percentage of selected features, which is defined to improve the algorithm's ability to reduce the irrelevant or redundant features. Given the feature set QS with N features, $2^N - 1$ feasible solutions are identified; thus, the search space of this model exponentially expands with an increase in the size of QS. In this paper, a metaheuristic search strategy is proposed to solve this model.

In our model, the value of $f_2(X)$ can be easily obtained by counting the number of features in X, and the value of $f_1(X)$ is estimated based on the training set. An extensively applied estimation method for $f_1(X)$ is inner *K*-fold cross-validation (CV) (Bermejo et al., 2014; J. Huang et al., 2007; Kohavi & John, 1997). In this paper, we use the inner 5-fold CV employed by Kohavi and John (1997) to estimate the objective function $f_1(X)$.

3. Optimization approach

3.1 Framework of the proposed method

The DMS is a novel derivative-free MOO algorithm that was recently proposed by Custódio et al. (2011). This algorithm is extended from the mono-objective direct search using the concept of Pareto dominance to maintain non-dominated solutions. The main DMS strategy is to survey the local region of a current best solution (poll center) to obtain better solutions at each iteration, which is termed the "poll step". The study by Türkşen et al. (2013) showed that the DMS method is effective in feature selection problems. In this paper, we attempt to solve the KQC selection model based on a DMS. However, two problems associated with using the DMS strategy may negatively affect the KQC selection results. First, the DMS search strategy (poll step), which is similar to hill climbing, is straightforward. However, the algorithm may easily be trapped in local optima. Therefore, the DMS search strategy must be diversified to improve the global search ability. Second, the DMS is a type of MOO algorithm that identifies a set of non-dominated solutions. From a practitioner's point of view, a reduction in the number of final solutions is necessary. Therefore, measures should be implemented to reduce the number of solutions obtained by the DMS.

To resolve the DMS limitations, we propose an MOO method based on an improved DMS and IPM (denoted by IDMS-IPM) to solve the KQC selection model. Fig. 1 shows the framework of the IDMS-IPM, which includes two phases. First, an IDMS is proposed to search for a set of non-dominated (candidate) solutions, and in the IDMS a mutation-poll step is embedded in the DMS strategy to improve the search ability. Second, to reduce the number of final solutions, the IPM is adopted to select the final solutions from the candidate solutions obtained by the IDMS (for more details, see A.-D. Li et al. (2016)).

According to Fig. 1, the mutation-poll step is only performed when the poll step fails to update the non-dominated solutions during the iterations. This step is an assisted search step that helps the algorithm escape from local optima. In the mutation-poll step, we first adopt a mutation process for the non-dominated solutions to search for new solutions (*refer to step 3-1 in Fig. 1*). Mutation (Bala et al., 1995; Oh et al., 2004) is a key step in increasing the diversity of the population in GAs. It can help the IDMS escape from local optima, which improves the global search ability. If the mutation process fails, we conduct an additional poll step (*refer to step 3-2 in Fig. 1*) in which one of the inferior solutions in S' is selected as the poll center at each iteration. By adding this additional poll step, new solutions are polled from both the current best solutions and inferior solutions, which helps the algorithm avoid local optima. S' is updated by adding the new mutated solutions that fail to update the non-dominated set S (*refer to step 3-1 in Fig. 1*).

In the following sections, the details of the IDMS-IPM, including the *solution representation, initialization, poll step, mutation-poll step,* and *stopping criteria* are introduced.

3.2 Solution representation

In this paper, we adopt the method employed by Türkşen et al. (2013) to represent the feature subsets with real-valued IDMS-IPM solutions. Given the production data D, a real vector $\boldsymbol{X} = (x_1, x_2, ..., x_N)$, $x_i \in [0,1]$, $i = 1,..., N$, is used to represent a solution X (i.e., feature subset) defined in the KQC selection model. Each x_i in $\boldsymbol{X}$ corresponds to a binary bit b_i defined as

$$b_i = \begin{cases} 1, x_i \geq 0.5 \\ 0, x_i < 0.5 \end{cases}, i = 1, ..., N, \tag{7}$$

where $b_i = 1$ denotes the *i*th feature selected and $b_i = 0$ denotes the *i*th feature eliminated. Thus, the $\boldsymbol{X}$ corresponds to only one feature subset.

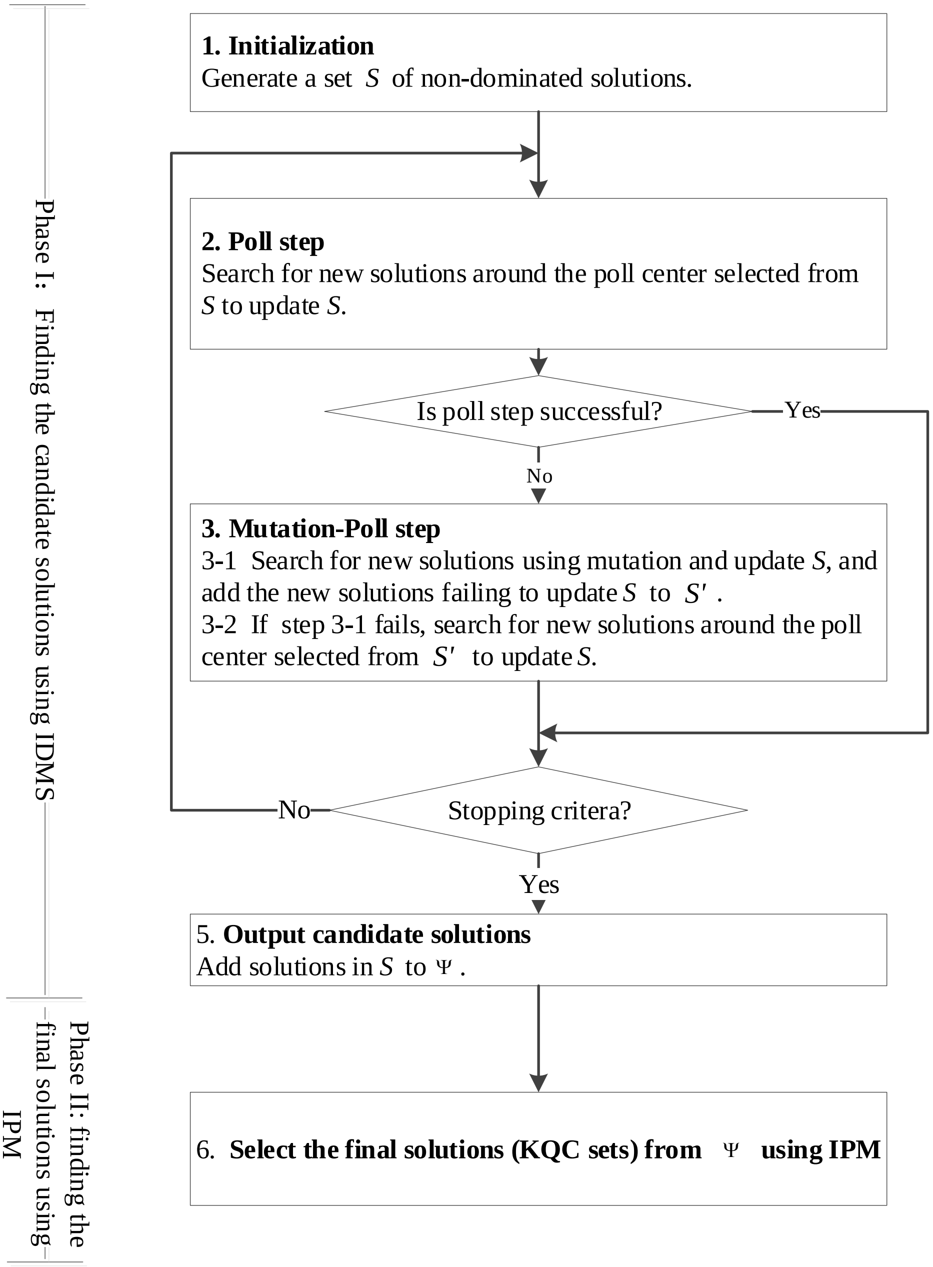

Phase I: Finding the candidate solutions using IDMS
1. Initialization
Generate a set S of non-dominated solutions.
2. Poll step
Search for new solutions around the poll center selected from S to update S.
Is poll step successful?
Yes
No
3. Mutation-Poll step
3-1 Search for new solutions using mutation and update S, and add the new solutions failing to update S to S′.
3-2 If step 3-1 fails, search for new solutions around the poll center selected from S′ to update S.
Stopping critera?
No
Yes
5. Output candidate solutions
Add solutions in S to Ψ.
Phase II: finding the final solutions using IPM
6. Select the final solutions (KQC sets) from Ψ using IPM

Fig. 1. Framework of the IDMS-IPM.

3.3 Initialization

In the initialization process, the set of non-dominated points is initialized as $S = \{(\boldsymbol{X}^0; \alpha^0)\}$, and the set of dominated points is initialized as $S' = \varnothing$, where $\boldsymbol{X}^0$ denotes a random generated feasible solution and α^0 denotes the initial step size parameter for $\boldsymbol{X}^0$. Note that in the IDMS-IPM, a point is defined as $(\boldsymbol{X};\alpha)$, where $\boldsymbol{X}$ denotes a solution (feature subset), and α is a parameter that defines the searching radius of the poll step.

3.4 Poll step

In the poll step, the first point in S is selected as the poll center, which we denote by $(\boldsymbol{X};\alpha)$. A set S_t of new solutions around the poll center is obtained as

$$S_t = \{(\boldsymbol{X} + \alpha \boldsymbol{d};\alpha) \mid \boldsymbol{d} \in \boldsymbol{D}\}, \tag{8}$$

where $\boldsymbol{D}$ represents a positive spanning set of $\boldsymbol{X}$, and each $\boldsymbol{d}$ is a vector in $\boldsymbol{D}$ (refer to Custódio et al. (2011)). The new solutions in S_t are then used to update the current non-dominated set S.

Additionally, we update the step size parameters α and reorder the points in S for the next iteration. If the poll step succeeds, the α of each new point in S from the poll step is updated as a random number in $[\alpha, \gamma\alpha]$, or the step size parameter α is updated as a random number in $[\beta_1\alpha, \beta_2\alpha]$, where $\gamma \geq 1$, $0 < \beta_1 \leq \beta_2 < 1$ are user-defined parameters. We then order the points in S according to each point's α in descending order, and if the poll center $(\boldsymbol{X};\alpha)$ is still in S, we arbitrarily move it to the last position in S to ensure that it will not be the poll center in the next poll step.

3.5 Mutation-poll step

The procedure of the mutation-poll step can be summarized by the *mutation-poll* function shown in Algorithm 1. The *mutation-poll* function first conducts a mutation process to search for new solutions to update the current best solution set S. The solutions that fail to update S are added to S', and a poll center is selected from S' for an additional poll step (line 25 in Algorithm 1). The process of the additional poll step is the same as that introduced in 3.4. S' follows the first in first out (FIFO) rule to maintain a constant size η, with $\eta = 20$ in the experiments. The $\mathit{filter}(S_f, S_t)$ in line 10 is a function that eliminates the dominated solutions from the set $S_f \cup S_t$; for more details, see Custódio et al. (2011).

```
function $Mutation-poll(S, S')$
    Input  : $S$ and $S'$
    Output: updated $S$ and $S'$
    /* Initialization. */
    $S_t \leftarrow \emptyset$;
    /* Search for new solutions with mutation operator
       SSOM and add them to $S_t$ . */
    for each $(\boldsymbol{X};\alpha) \in S$ do
        $\boldsymbol{X}' \leftarrow SSOM(\boldsymbol{X})$;
        $S_t \leftarrow S_t \cup (\boldsymbol{X}';\alpha^0)$;
    end
    /* Update the $S$ with $S_t$ , and determine whether the
       mutation succeeds. */
    $S_f \leftarrow S, S \leftarrow filter(S_f, S_t)$;
    if $S_f \neq S$ then
        $Success \leftarrow \boldsymbol{True}$;
    else
        $Success \leftarrow \boldsymbol{False}$;
    end
    /* Add new solutions failing to update $S$ to $S'$. */
    for each $(\boldsymbol{X};\alpha) \in S_t$ do
        if $(\mathbf{X};\alpha) \notin S$ then
            $S' \leftarrow (\boldsymbol{X};\alpha) \cup S'$;
        end
    end
    /* Conduct the additional poll step. */
    if $Success = \boldsymbol{False}$ then
        Let $(\boldsymbol{X};\alpha)$ be a random point selected from $S'$;
        Search for new solutions using a poll step where $(\boldsymbol{X};\alpha)$ is the
         poll center;
    end
    return updated $S$ and $S'$;
end
```

Algorithm 1. Pseudocode of the mutation-poll step.

The choice of mutation operator is essential for the mutation-poll step. In this paper, the subset size oriented mutation (SSOM) operator (Oh et al., 2004), which is designed for feature selection tasks, is adopted. The SSOM operator was originally designed for binary encoded solutions; thus, we adapt the operator to real-valued solutions for the IDMS-IPM. In the solution $\boldsymbol{X} = (x_1, x_2, ..., x_N)$, $x_i \in [0,1]$, $i = 1,...,N$, and a bit x_i will mutate to x_i' as

$$x_i' = \begin{cases} rand(0.5,1), x_i < 0.5 \\ rand(0,0.5), x_i \geq 0.5 \end{cases}, \tag{9}$$

where $rand(a,b)$ denotes a random value from the uniform distribution on the interval (a,b). The mutation probabilities of the bits in $[0.5,1]$ differ from those of the bits in $[0,0.5)$. Let $C_1 = \{x_i \mid x_i \in [0.5,1]; i = 1,..., N\}$ and $C_2 = \{x_i \mid x_i \in [0,0.5); i = 1,..., N\}$ represent two sets that contain the two types of bits; let p_m and p_m' represent the mutation probabilities of bits in C_1 and C_2, respectively; and let $\#C_1$, $\#C_2$ represent the sizes of C_1 and C_2, respectively. Then, p_m' is calculated as

$$p_m' = {}^{\#C_1}\!\big/_{\#C_2} \bullet p_m, \tag{10}$$

where p_m is a user-defined parameter.

3.6 Stopping criteria

The stopping criteria can be based on a maximum number of iterations or a maximum number of function evaluations. Compared with most EAs, such as GAs or PSO, the number of function evaluations of the IDMS-IPM at different iterations varies. Thus, for a comparison with the benchmark methods, we set a predefined number of function evaluations to represent the stopping criteria in this paper. Note that with the iteration of the IDMS-IPM or DMS, the step size parameters for the solutions in S converge to 0, and the poll step will hardly search for new solutions. In the DMS, a threshold value (a very small value) for the step size parameters in S is employed to terminate the iterations. Similarly, we apply the threshold value α_ε to terminate the IDMS-IPM, even when the predefined maximum function evaluations are not reached. In the experiments, we let $\alpha_\varepsilon = 1\times10^{-3}$ as suggested by Custódio et al. (2011).

As stated in section 2.2, the evaluation of the objective function f_1 requires an inner 5-fold CV process, which is time-consuming. To improve the time efficiency of the proposed IDMS-IPM, we adopt a strategy used by Custódio et al. (2011) to eliminate the calls of the inner *K*-fold CVs for evaluating the objective function f_1 without changing the IDMS-IPM performance. This strategy uses a list that stores the objective functions of each evaluated solution $\boldsymbol{X}$ during the iterations. Therefore, the objective function values of a new solution from the poll or mutation-poll step can be obtained directly from the list if this solution has already been evaluated.

4. Experimental studies

4.1 Experimental design

To evaluate the proposed method, four imbalanced production datasets, LATEX, ADPN, PAPER and SPIRA, are employed in the experiments. The LATEX, ADPN and SPIRA datasets were first used by Gauchi and Chagnon (2001) for variable selection with a prediction purpose, and PAPER was first used by Wold et al. (2001) to test the effectiveness of partial least regression (PLS) tools. LATEX was collected from the polymerization process of latex production, in which the QCs include the catalyst level and temperature. ADPN was collected from a sub-phase of nylon manufacturing, where the QCs include pressure, flow and temperature. PAPER was gathered from a paper recycling process in which the QCs include the temperature and concentration at various times. SPIRA was collected from an antibiotics production process in which the QCs include temperature level, stirring power, and peak oxygen consumption. For each dataset, the instances are classified as "regular" quality (labeled "-1") and "premium" quality (labeled "1") according to the cut-off limits on the response variable y provided by the authors, Gauchi and Chagnon (2001) and Wold et al. (2001), and the datasets were first used by Anzanello et al. (2009) for KQC (variable) selection with a classification purpose. Detailed information on the four datasets is recorded in Table 2. Note that another dataset named OXY, which was used by Anzanello et al. (2009), contains only 30 instances. In this paper, we implement the widely used 10-fold CV to validate the proposed method, and 30 instances are equivalent to 3 test instances in each fold of the experiment. Therefore, the number of test instances is insufficient to validate the effectiveness of each method, and we do not use the OXY dataset in the experiments in this paper.

Table 2. Details of the datasets.

Dataset	Number of QCs	Number of instances	Number of positive (premium quality) instances	Number of negative (regular quality) instances
LATEX	117	262	78	184
ADPN	100	71	20	51
PAPER	54	384	33	351
SPIRA	96	145	50	95

In this paper, nine benchmark methods are used in the experiments, and they can be classified into three categories. The first category includes three PLS-based variable (feature) selection methods: PLS-MCP (Anzanello et al., 2009), PLS-PVS (Anzanello et al., 2012) and PLS-SFFS (Tian et al., 2013). These methods first sort the variables according to the PLS-based variable importance metrics and then adopt various strategies to select the final variables. The second category includes two conventional feature selection algorithms: SFS and SBS. The third category includes four feature selection methods based on the MOO: DMS-IPM, NSPSOFS, CMDPSOFS and NSGAII-IPM. The DMS-IPM adopts the same framework as the IDMS-IPM with the exception that it employs DMS as the search algorithm in phase I. The NSPSOFS and CMDPSOFS methods were recently proposed by Xue et al. (2013) and are based on two MOPSO methods, NSPSO and CMDPSO. The NSGAII-IPM was recently proposed by A.-D. Li et al. (2016) for KQC selection and is based on a typical multi-objective GA method, NSGA-II. The NSPSOFS, CMDPSOFS and NSGAII-IPM methods adopt the same feature selection models, i.e., maximizing classification accuracy and minimizing feature subset size. Because the NSPSOFS and CMDPSOFS methods return a set of non-dominated solutions, we also apply the IPM to select final solutions from the solutions obtained by these two methods for comparison. The experiments with SFS and SBS are run using Waikato Environment for Knowledge Analysis (WEKA) 3.7.13 (Hall et al., 2009). The experiments with other methods are conducted in MATLAB R2016b. A naïve Bayesian (NB) classifier (John & Langley, 1995) is selected as the learning algorithm due to its high performance and simplicity, and it is invoked from WEKA. All experiments are run on an Intel Core PC with a 3.4 GHz CPU and 16 GB of RAM.

In the experiments, the default settings of the PLS-MCP (Anzanello et al., 2009), PLS-PVS (Anzanello et al., 2012) and PLS-SFFS (Tian et al., 2013) methods are obtained from published works. For SFS and SBS, the default settings in WEKA are employed. The parameters commonly employed in the IDMS-IPM and DMS-IPM are set to $\alpha^0 = 2$, $\beta_1 = \beta_2 = 0.95$ and $\gamma = 1$, and the mutation parameter in the IDMS-IPM is set to $p_m = 1/N$, where N denotes the number of original QCs. The same stopping criteria are adopted for the DMS-IPM and IDMS-IPM. These two methods stop when 100,000 objective function evaluations are performed or a threshold value of $\alpha_\varepsilon = 1\times10^{-3}$ is attained. These parameters are set after several parameter-tuning experiments. For the NSPSOFS and CMDPSOFS methods, the maximum number of iterations and population size are set to 1,000 and 100, respectively; thus, the number of expected objective function evaluations would be equal to that in the DMS-IPM and IDMS-IPM. Other settings are equivalent to those in the experiments performed by Xue et al. (2013): for the NSPSOFS method, the inertia weight is $w = 0.7298$, and the acceleration parameters are $c_1 = c_2 = 1.49618$; for the CMDPSOFS method, w is a random value from $[0.1, 0.5]$, and the mutation rate is $p = 1/N$. For the NSGAII-IPM, the maximum number of iterations and population size are also set to 1,000 and 100, respectively, and we set the crossover probability $p_c = 0.9$ and mutation probability $p_m = 1/N$, as adopted by A.-D. Li et al. (2016).

We adopt the commonly employed validation method 10-fold CV (Rodriguez et al., 2010) to run the experiments. At each 10-fold CV, 10 runs of experiments are conducted. The proposed IDMS-IPM and benchmark methods based on MOO are actually stochastic optimization methods; thus, the results may vary over different 10-fold CVs. To comprehensively test the effectiveness of the proposed method, we repeat the 10-fold CV 3 times, which results in $3\times10=30$ runs of experiments. Although the SFS, SBS and PLS-based methods are deterministic methods and their results for different 10-fold CVs are the same, we still run the 10-fold CV 3 times using these deterministic methods for comparison. It is worth noting that the 10-fold CV and the inner 5-fold CV (stated in section 2.2) involve similar processes, but they have different objectives. The inner 5-fold CV is used on the training set, which is used as an objective function evaluation tool, a part of the KQC selection method. The 10-fold CV is used on the whole dataset, which is used to evaluate the effectiveness of methods.

4.2 Comparison of KQC selection results with benchmark methods

In this section, the final KQC selection results yielded by the IDMS-IPM and the benchmark methods are compared. Four performance metrics, sensitivity, specificity, accuracy and retained KQC percentage, are employed to validate the proposed method. The classification accuracy and retained KQC percentage are common measures employed to evaluate the effectiveness of KQC selection methods (Anzanello et al., 2012; A.-D. Li et al., 2016). Because the production datasets are imbalanced, we apply sensitivity and specificity to measure the classification ability of selected KQCs as well as accuracy, as used by Anzanello et al. (2012). For each metric, we conduct Wilcoxon signed-rank tests (Demšar, 2006; Wilcoxon, 1945) between the IDMS-IPM and the benchmark methods to determine whether the differences are statistically significant.

A. Comparison with PLS-based methods

Table 3 presents the obtained performance metrics of the IDMS-IPM and PLS-based benchmark methods, i.e., PLS-MCP, PLS-PVS and PLS-SFFS, where “mean” and “standard deviation” denote the average and standard deviation values of the performance metrics from 30 runs of experiments. First, according to the average results over the four datasets, the IDMS-IPM performs best for three of four performance metrics, i.e., sensitivity, accuracy and retained KQC percentage. IDMS-IPM also obtains a specificity rate similar to the highest one obtained by the PLS-MCP. Moreover, the IDMS-IPM obtains both high sensitivity and specificity rates, whereas the PLS-based methods obtain much lower sensitivity rates than specificity rates. Second, the performance of the IDMS-IPM is more stable than that of the PLS-based methods because the PLS-based methods present considerable variation in performance for different datasets. PLS-based methods can obtain slightly better results for the LATEX and SPIRA than the IDMS-IPM can. However, the sensitivity rates of the PLS-based methods for the ADPN and PAPER datasets are much lower than that of the IDMS-IPM because they are affected by the data imbalance. Therefore, the selected KQCs of these PLS-based methods demonstrate poor performance in classifying the minority instances (premium products), and we cannot conclude that they accurately select KQCs. Finally, the P-values also indicate that, in most cases, the IDMS-IPM obtains a significantly lower KQC percentage than do the PLS-based methods. According to the abovementioned results, we can conclude that the IDMS-IPM performs more robustly on different imbalanced production datasets than do the PLS-based methods.

B. Comparison with the SFS and SBS

Table 4 shows the obtained performance metrics and the results of the Wilcoxon signed-rank tests for the DMS-IPM, SFS and SBS. The IDMS-IPM obtains significantly higher sensitivity rates in most cases and obtains specificity and accuracy rates similar to those of the benchmark methods. SFS and SBS obtain much lower sensitivity rates than specificity rates, which shows that these two methods are affected by the data imbalance. Additionally, the results demonstrate that the IDMS-IPM selects significantly fewer KQCs than do SFS and SBS. The abovementioned results show that the IDMS-IPM performs better than SFS and SBS in KQC selection using imbalanced production data.

C. Comparison with the multi-objective feature selection methods

Table 5 shows the performance metrics and results of the Wilcoxon signed-rank tests obtained for the IDMS-IPM and the benchmark multi-objective feature selection methods. First, the results show that IDMS-IPM performs the best compared with the studied multi-objective feature selection methods. The IDMS-IPM obtains high rates of sensitivity, specificity and accuracy, which are slightly higher than those of the DMS-IPM. In comparison, the NSPSOFS, CMDPSOFS and NSGAII-IPM obtain similar specificity and accuracy rates to those of the IDMS-IPM but much lower sensitivity rates than that of the IDMS-IPM; furthermore, the *P*-values from the Wilcoxon signed-rank tests indicate that the differences in sensitivity rates between the IDMS-IPM and these benchmark methods are significant. These results show that the data imbalance actually affects the classification performances of the NSPSOFS, CMDPSOFS and NSGAII-IPM. Second, all multi-objective feature selection methods tested in this section are effective in reducing the number of QCs, as the retained KQC percentages of these methods are lower than those of the SFS, SBS and PLS-based methods in most cases. This finding denotes an advantage of the multi-objective-based KQC selection in eliminating the number of QCs.

The comparisons above show that the proposed IDMS-IPM can effectively select KQCs for imbalanced production datasets. First, the IDMS-IPM can obtain high sensitivity rates, specificity rates and accuracy rates, indicating that the selected KQCs effectively predict the products' quality. In comparison, the benchmark methods (except for the DMS-IPM) are generally negatively affected by the data imbalance, and the selected KQCs exhibit a poor predictive ability for the premium products (minority instances). The results imply that the KQC selection model in this paper can handle the problem of data imbalance. Second, the abovementioned results indicate that the IDMS-IPM is effective in reducing the number of QCs.

Table 3. Comparison of the performance metrics (%) obtained by the IDMS-IPM and PLS-based benchmark methods for the LATEX, ADPN, PAPER and SPIRA datasets.

		IDMS-IPM				PLS-MCP				PLS-PVS				PLS-SFFS			
		SEN	SPE	ACC	RKP	SEN	SPE	ACC	RKP	SEN	SPE	ACC	RKP	SEN	SPE	ACC	RKP
LATEX	Mean	74.46	**84.26**	81.36	**3.56**	83.57	80.47	81.31	23.42	**87.32**	81.52	**83.21**	3.68	75.71	83.10	80.93	5.64
	Standard deviation	17.38	9.47	8.17	0.63	14.81	6.95	6.71	31.62	9.70	7.90	5.45	0.39	14.13	6.40	4.15	1.28
	P-value					**1.47--**	**0.28++**	74.06	**0.09++**	**0.02--**	**5.31+**	25.19	48.50	49.24	42.56	63.54	**0.00++**
ADPN	Mean	**90. 00**	82. 22	**84. 38**	**2. 27**	55.00	**88.33**	79.11	2.80	55.00	**88.33**	79.11	2.80	65.00	86.00	80.18	4.10
	Standard deviation	20. 00	12. 69	8. 44	0. 44	41.53	13.10	12.37	1.33	41.53	13.10	12.37	1.33	39.05	15.62	13.18	0.94
	P-value					**0.01++**	**1.95--**	11.18	**9.19+**	**0.01++**	**1.95--**	11.18	**9.19+**	**0.05++**	47.67	16.78	**0.00++**
PAPER	Mean	**88.33**	88.03	88.02	**5.68**	62.50	**92.31**	**89.82**	90.56	80.00	84.33	83.84	7.78	77.50	89.45	88.51	22.59
	Standard deviation	15.90	4.58	4.43	0.82	25.89	5.43	5.07	5.82	17.56	6.15	5.13	3.39	21.10	5.87	4.79	5.09
	P-value					**0.02++**	**0.03--**	**8.21-**	**0.00++**	**2.64++**	**0.27++**	**0.04++**	**0.92++**	**0.92++**	38.89	94.93	**0.00++**
SPIRA	Mean	75.56	85.04	81.68	**3.72**	**80.00**	**87.56**	**84.90**	5.31	78.00	**87.56**	84.19	3.75	76.00	84.22	81.38	5.42
	Standard deviation	16.98	13.02	10.53	0.79	12.65	12.19	8.55	4.92	14.00	12.19	8.14	0.95	14.97	11.87	5.49	0.91
	P-value					**1.56--**	12.50	**0.39--**	27.78	12.50	12.50	**3.13--**	88.38	100.00	81.69	97.01	**0.00++**
AVERAGE		**82.09**	84.89	**83.86**	**3.81**	70.27	**87.17**	83.79	30.52	75.08	85.44	82.59	4.50	73.55	85.69	82.75	9.44

SEN, SPE, ACC, and RKP denote the sensitivity rate, specificity rate, accuracy rate and retained KQC percentage, respectively.

+ and ++ denote that IDMS-IPM statistically outperforms benchmark methods at significance level $\alpha = 10\%$ and $\alpha = 5\%$.

- and -- denote that benchmark methods statistically outperform IDMS-IPM at significance level $\alpha = 10\%$ and $\alpha = 5\%$.

Table 4. Comparison of the performance metrics (%) obtained by the IDMS-IPM, SFS and SBS for the LATEX, ADPN, PAPER and SPIRA datasets.

		IDMS-IPM				SFS				SBS			
		SEN	SPE	ACC	RKP	SEN	SPE	ACC	RKP	SEN	SPE	ACC	RKP
LATEX	Mean	**74.46**	84.26	**81.36**	**3.56**	49.82	**90.70**	78.62	6.58	63.93	82.13	76.75	80.17
	Standard deviation	17.38	9.47	8.17	0.63	17.79	5.63	6.72	2.48	15.23	9.36	8.72	13.55
	P-value					**0.00++**	**0.03--**	**2.08++**	**0.00++**	**1.17++**	21.31	**1.38++**	**0.00++**
ADPN	Mean	**90.00**	**82.22**	**84.38**	**2.27**	75.00	78.00	77.32	5.30	65.00	80.00	75.71	25.80
	Standard deviation	20.00	12.69	8.44	0.44	25.00	20.88	13.22	1.27	32.02	12.65	14.36	9.87
	P-value					**0.39++**	11.32	**1.55++**	**0.00++**	**0.01++**	18.07	**0.15++**	**0.00++**
PAPER	Mean	**88.33**	88.03	**88.02**	**5.68**	58.33	84.36	82.06	7.41	80.83	**88.88**	87.99	47.78
	Standard deviation	15.90	4.58	4.43	0.82	30.28	7.60	7.40	3.31	20.43	7.17	5.59	18.27
	P-value					**0.01++**	**6.72+**	**0.61++**	**1.38++**	**6.05+**	58.64	39.00	**0.00++**
SPIRA	Mean	**75.56**	85.04	81.68	3.72	74.00	**87.56**	**82.81**	**3.65**	62.00	79.22	73.38	53.85
	Standard deviation	16.98	13.02	10.53	0.79	15.62	8.90	5.38	1.07	24.41	11.32	11.53	17.20
	P-value					50.78	32.75	49.23	60.73	**1.21++**	**8.82+**	**1.22++**	**0.00++**
AVERAGE		**82.09**	84.89	**83.86**	**3.81**	64.29	**85.15**	80.20	5.73	67.94	82.56	78.46	51.90

SEN, SPE, ACC, and RKP denote the sensitivity rate, specificity rate, accuracy rate and retained KQC percentage, respectively.

+ and ++ denote that IDMS-IPM statistically outperforms benchmark methods at significance level $\alpha = 10\%$ and $\alpha = 5\%$.

- and -- denote that benchmark methods statistically outperform IDMS-IPM at significance level $\alpha = 10\%$ and $\alpha = 5\%$.

Table 5. Comparisons of the performance metrics (%) obtained by the IDMS-IPM, DMS-IPM, NSPSOFS, CMDPSOFS, and NSGAII-IPM for the LATEX, ADPN, PAPER and SPIRA datasets.

		IDMS-IPM				DMS-IPM				NSPSOFS				CMDPSOFS				NSGAII-IPM			
		SEN	SPE	ACC	RKP	SEN	SPE	ACC	RKP	SEN	SPE	ACC	RKP	SEN	SPE	ACC	RKP	SEN	SPE	ACC	RKP
LATEX	Mean	**74.46**	84.26	**81.36**	3.56	73.27	83.32	80.31	3.65	59.25	87.12	78.95	3.22	61.77	**87.76**	80.13	**3.13**	63.89	86.51	79.92	3.48
	Standard deviation	17.38	9.47	8.17	0.63	19.47	9.13	8.22	0.62	23.51	8.95	9.13	0.93	18.94	8.44	5.49	0.89	19.53	9.62	5.99	0.62
	P-value					35.55	25.36	34.96	59.05	**0.06++**	11.14	**7.53+**	**3.95--**	**0.90++**	**1.28--**	36.72	**4.74--**	**1.02++**	12.71	22.72	60.95
ADPN	Mean	**90.00**	82.22	**84.38**	**2.27**	85.00	81.22	82.24	2.37	75.00	85.63	82.63	3.00	74.06	**86.31**	82.85	2.73	74.45	85.27	82.36	2.70
	Standard deviation	20.00	12.69	8.44	0.44	22.91	13.21	8.13	0.48	29.19	10.56	8.62	0.63	32.47	11.83	9.43	0.44	28.45	13.54	8.82	0.46
	P-value					25.00	72.66	24.22	50.78	**0.39++**	17.46	49.00	**0.02++**	**0.10++**	**7.40-**	56.63	**0.05++**	**0.20++**	36.77	31.84	**0.10++**
PAPER	Mean	**88.33**	88.03	88.02	5.68	87.22	87.08	87.08	**5.62**	68.33	91.10	89.07	10.25	71.67	90.46	88.79	9.69	62.33	**91.66**	**89.12**	9.63
	Standard deviation	15.90	4.58	4.43	0.82	17.18	4.95	4.91	1.12	19.52	4.89	4.63	1.90	21.79	4.76	3.71	1.56	24.02	4.70	5.06	1.39
	P-value					100	15.29	22.45	100	**0.04++**	**0.12--**	25.26	**0.00++**	**0.10++**	**0.39--**	51.64	**0.00++**	**0.01++**	**0.06--**	13.57	**0.00++**
SPIRA	Mean	**75.56**	85.04	**81.68**	3.72	71.33	**85.30**	80.46	4.06	62.94	84.32	76.96	3.85	66.78	84.67	78.44	**2.95**	67.04	82.61	77.24	4.06
	Standard deviation	16.98	13.02	10.53	0.79	14.31	12.79	8.24	1.09	19.48	11.97	8.35	1.66	21.87	11.74	8.02	0.76	17.98	13.54	8.18	0.91
	P-value					19.38	86.33	76.98	16.65	**0.02++**	92.50	**1.95++**	96.74	**0.07++**	84.78	**9.65+**	**0.37--**	**0.49++**	30.18	**2.48++**	**7.39+**
AVERAGE		**82.09**	84.89	**83.86**	**3.81**	79.21	84.23	82.52	3.92	66.38	87.04	81.90	5.08	68.57	**87.30**	82.55	4.63	66.93	86.51	82.16	4.97

SEN, SPE, ACC, and RKP denote the sensitivity rate, specificity rate, accuracy rate and retained KQC percentage, respectively.

+ and ++ denote that IDMS-IPM statistically outperforms benchmark methods at significance level $\alpha = 10\%$ and $\alpha = 5\%$.

- and -- denote that benchmark methods statistically outperform IDMS-IPM at significance level $\alpha = 10\%$ and $\alpha = 5\%$.

4.3 Computation time

Table 6 shows the average computation time per experimental run of each method. The IDMS-IPM and DMS-IPM may terminate before the predefined maximum function evaluations if the step size parameters reach the threshold value of 0.001; thus, we list the average number of function evaluations per experimental run for these two methods to better analyze the results.

According to Table 6, the PLS-based methods, i.e., PLS-MCP, PLS-PVS and PLS-SFFS, require much less computation time than do other methods. This finding is related to the use of the PLS to obtain the weight of each feature, which is time-efficient because a learning algorithm does not need to be built to evaluate feature importance. The other methods, i.e., the SFS, SBS and multi-objective feature selection methods, require much more time than the PLS-based methods because they are based on the wrapper model, which requires considerable computation time in a function evaluation process. Among these wrapper methods, SFS and SBS are more time-efficient than the other methods because these two methods adopt hill climbing search strategies with fewer iterations for optimization. However, the drawback of hill climbing is that it can be easily trapped in local optima. The IDMS-IPM, NSPSOFS, CMDPSOFS and NSGAII-IPM present similar computation times for the datasets. However, for all datasets except ADPN, the IDMS-IPM presents a slightly lower computation time. This finding indicates that the adopted strategy (stated in section 3.6) of reducing the calls of function evaluations is effective for computation time reduction. The DMS-IPM requires less computation time than do the other multi-objective feature selection methods because it terminates with fewer function evaluations. The results indicate that the number of function evaluations is directly related to the computation time of the KQC (feature) selection methods.

In summary, the proposed IDMS-IPM is not as time-efficient as the PLS-based methods, SFS and SBS, and the computation time of the proposed method is similar to that of existing multi-objective feature selection methods. These results are associated with the adopted wrapper models, which improve the time cost of each function evaluation, and with the metaheuristic search strategy, which improves the number of function evaluations. Thus, simplifying the evaluation process for the objective function and improving the search strategy are two approaches to reducing the time complexity of KQC selection methods.

Table 6. Average computation time (seconds) for each method.

Dataset	IDMS-IPM (number of function evaluations)	DMS-IPM (number of function evaluations)	PLS-MCP	PLS-PVS	PLS-SFFS	SFS	SBS	NSPSOFS	CMDPSOFS	NSGAII-IPM
LATEX	3.52E+3 (1.00E+5)	3.11E+3 (9.92E+4)	1.81E+1	1.91E+1	5.54E+0	8.85E+1	2.12E+3	3.62E+3	4.18E+3	3.83E+3
ADPN	8.93E+2 (1.00E+5)	4.78E+2 (8.65E+4)	2.13E+0	2.12E+0	8.00E-1	1.25E+1	3.69E+2	8.41E+2	9.37E+2	8.08E+2
PAPER	1.76E+3 (1.00E+5)	5.44E+2 (4.25E+4)	5.88E+0	5.94E+0	3.08E+0	1.33E+1	4.87E+2	4.73E+3	4.58E+3	4.33E+3
SPIRA	1.42E+3 (1.00E+5)	1.08E+3 (9.04E+4)	6.15E+0	5.87E+0	1.59E+0	1.16E+1	8.96E+2	1.88E+3	1.71E+3	1.72E+3

4.4 Comparison of the search abilities between the IDMS and DMS

The abovementioned analyses indicate that the IDMS-IPM and DMS-IPM can well handle data imbalances, thus demonstrating the effectiveness of the KQC selection model proposed in this paper. The difference between the IDMS-IPM and DMS-IPM is that they adopt different search strategies (i.e., IDMS and DMS) in the first phases. Therefore, we compare the search abilities of the IDMS and DMS to further verify the proposed IDMS-IPM. The comparisons are performed as follows. First, we compare the search results of the IDMS and DMS, where the obtained objective functions f_1 and f_2 of the candidate solutions obtained in the first phases of IDMS-IPM and DMS-IPM are employed. Second, we compare the convergence properties of the IDMS and DMS using the records during the iterations.

A. Comparison of the search results

We adopt the Two Set Coverage (SC) metric proposed by Zitzler and Thiele (1998) to compare the search results between the IDMS and DMS. Let S_1 and S_2 represent two solution sets obtained by two MOO methods. The metric $SC(S_1, S_2)$ is defined as the proportion of solutions in S_2 that are covered by the (dominated by or equal to) solutions in S_1 and is calculated as

$$SC(S_1, S_2) = \frac{\#\{s \in S_2 \mid \exists s' \in S_1 : s' \prec s \text{ or } s' = s\ \}}{\# S_2}, \tag{11}$$

where "$s' \prec s$" implies that the solution s' dominates s and $\# S$ denotes the number of elements in the set S. If each solution in S_2 is dominated by (or equal to) one or several solutions in S_1, then $SC(S_1, S_2) = 1$. Otherwise, $SC(S_1, S_2) = 0$. Generally, both $SC(S_1, S_2)$ and $SC(S_2, S_1)$ should be calculated. If $SC(S_1, S_2) = 1$ and $SC(S_2, S_1) = 0$, then S_1 is better than S_2. If $SC(S_1, S_2) > SC(S_2, S_1)$, we conclude that S_1 is relatively better than S_2. In an ideal situation, if both S_1 and S_2 are equal to the true Pareto set, then $SC(S_1, S_2) = SC(S_2, S_1)) = 1$.

Table 7 records the SC results from the comparison of the IDMS and DMS and lists the mean and standard deviation of the SC values from the 30 runs of experiments. The mean *SC*(IDMS, DMS) value is higher than the mean *SC*(DMS, IDMS) value for each dataset. The Wilcoxon signed-rank test is conducted to verify whether the *SC*(IDMS, DMS) is significantly higher than the *SC*(DMS, IDMS). The *P*-values show that the differences between the *SC*(IDMS, DMS) and *SC*(DMS, IDMS) are significant for three of four datasets (i.e., ADPN, PAPER and SPIRA) at the 5% significance level. The abovementioned results show that the search results of the IDMS are relatively better than the search results of the DMS according to the SC metric.

Table 7. Comparison of the solutions obtained by the IDMS and DMS.

	SC (IDMS, DMS)		SC (DMS, IDMS)		
	Mean	Standard deviation	Mean	Standard deviation	p-value
LATEX	**0.6908**	0.2376	0.5851	0.2797	0.2329
ADPN	**0.8597**	0.2297	0.6638	0.3472	**0.0046****
PAPER	**0.8242**	0.1963	0.4428	0.2607	**0.0001****
SPIRA	**0.8070**	0.1945	0.5562	0.2205	**0.0015****

** denotes the statistical significance at the significance level $\alpha = 5\%$.

B. Comparison of convergence properties

To compare the convergence properties of the IDMS and DMS, a metric should be defined to measure the goodness of the solutions during the iterations. For multi-objective problems, a straightforward goodness measure strategy involves using the distance from the solutions to the true Pareto front. However, the true Pareto front for the feature selection problem in this paper is unknown. Thus, we use the best solutions obtained by the IDMS and DMS in the experiments to represent the Pareto front for each experiment. The Pareto front and the solutions obtained by each iteration can be denoted by discrete points in the objective space. Fig. 2 illustrates the solutions for the Pareto front and current best solutions for an iteration. In this paper, we propose a distance metric to evaluate the similarity of the solutions for an iteration to the Pareto front. Let $S^* = \{s_1^*, s_2^*, ..., s_M^*\}$ be the set of solutions for the Pareto front and $S = \{s_1, s_2, ..., s_N\}$ be the set of solutions for an iteration. The distance between S^* and S is obtained as follows:

$$D(S^*, S) = \frac{[\sum_{s_m^* \in S^*} \min_{s_n \in S} \sqrt{\sum_{i=1}^{2} (f_i(s_m^*) - f_i(s_n))^2}]/M + [\sum_{s_n \in S} \min_{s_m^* \in S^*} \sqrt{\sum_{i=1}^{2} (f_i(s_m^*) - f_i(s_n))^2})]/N}{2}. \quad (12)$$

According to Eq. (12), we calculate the Euclidean distance between each Pareto solution in S^* and the nearest solution in S and obtain the first average distance. Then, we calculate the Euclidean distance between each solution in S and the nearest Pareto solution in S^* and obtain the second average distance. Finally, we obtain the mean value of the obtained two average distances as $D(S^*, S)$.

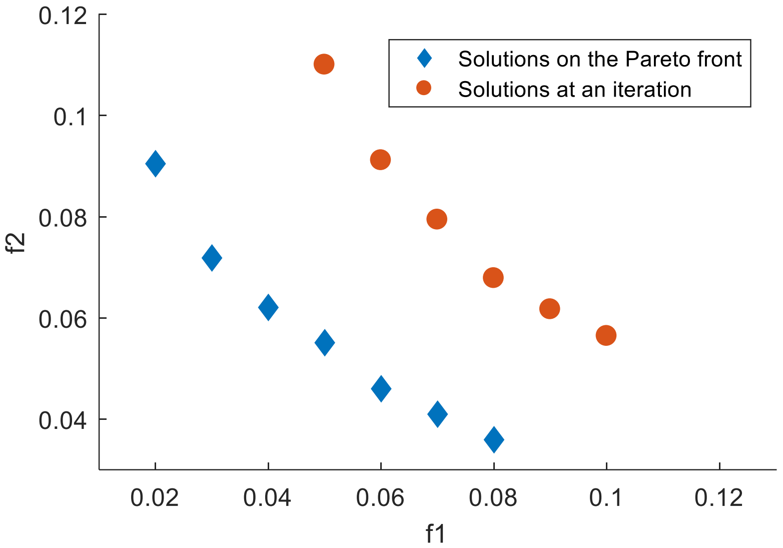


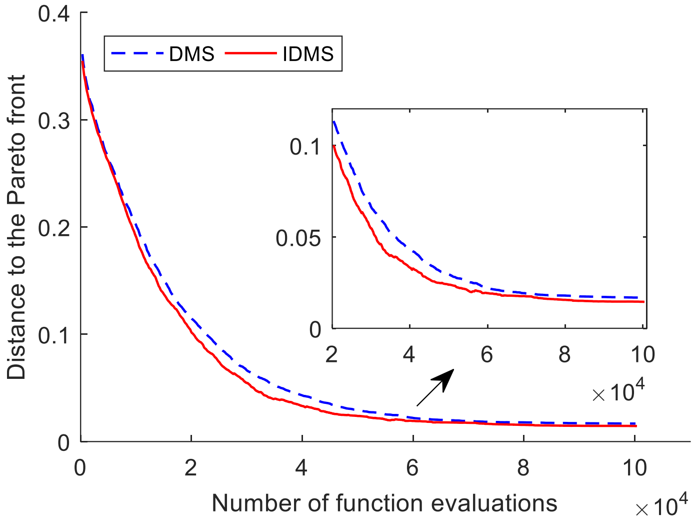
Fig. 2. Illustration of the Pareto front and solutions for an iteration.

We draw the average convergence curves for the IDMS and DMS over the 30 runs of experiments to show the convergence properties. Generally, the solutions at each iteration obtained by the population-based search methods, e.g., GAs and PSO, are compared for convergence analysis. However, for the IDMS and DMS, directly comparing the solutions at each iteration leads to biased results because the number of function evaluations of the IDMS and DMS for an iteration is uncertain. Therefore, we adopt a fair comparison strategy that compares the solutions at each number of function evaluations instead of each iteration. We first record the solutions and the number of function evaluations after each iteration and calculate the distance $D(S^*, S)$ to measure the goodness of the solutions. Then, we use linear interpolation to obtain the distances of all numbers of function evaluations. Let $D(S^*, S_{e(i)})$ be the distance at iteration $i \geq 1$, where $e(i)$ indicates the number of function evaluations after iteration i. The distance $D(S^*, S_e)$ of each number $e \in \{e \in \mathbb{Z} \mid e(i) < e < e(i+1)\}$ of function evaluations is obtained as

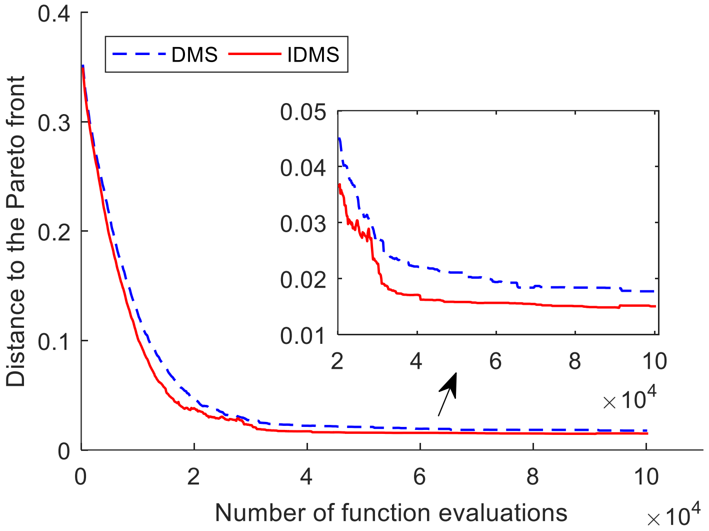
$$D(S^*, S_e) = D(S^*, S_{e(i)}) + (e - e(i)) \cdot \frac{D(S^*, S_{e(i+1)}) - D(S^*, S_{e(i)})}{e(i+1)\text{-}e(i)}. \tag{13}$$

Thus, we can average the distances over the 30 experiments for each number of function evaluations to draw the average convergence curve. Note that in the IDMS and DMS, the total number of function evaluations may differ over various experiments. Therefore, not all experimental runs reach a certain number of function evaluations to obtain the distance values. In this situation, we only use the parts of runs that can obtain distance values $D(S^*, S)$ to calculate the average distance.

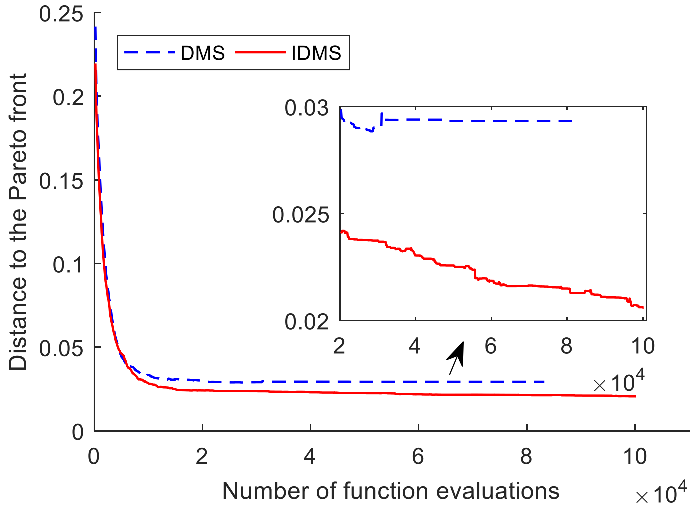
Fig. 3 shows the convergence curves (i.e., the number of function evaluations versus the distance to the Pareto front) of the IDMS and DMS for the four datasets. In the figures, we magnify the curves from function evaluations 20,000 to 100,000 to clarify the curves. First, the convergence curves of the IDMS and DMS decline rapidly in the early function evaluations and then decline slowly. In addition, the convergence speeds for various datasets are different. The two methods converge fastest for the PAPER dataset, whereas they converge most slowly for the LATEX dataset. This finding is related to the greater number of QCs in LATEX than in PAPER and the improved complexity of KQC selection problems with an increased number of original QCs. Second, the convergence speed of the IDMS is faster than that of the DMS for all datasets. The curves of the IDMS are below the curves of the DMS for all 4 datasets, which indicates better solutions are obtained by the IDMS than by the DMS at each number of function evaluations. The abovementioned results show that the search ability of the proposed IDMS is improved compared with that of the DMS.



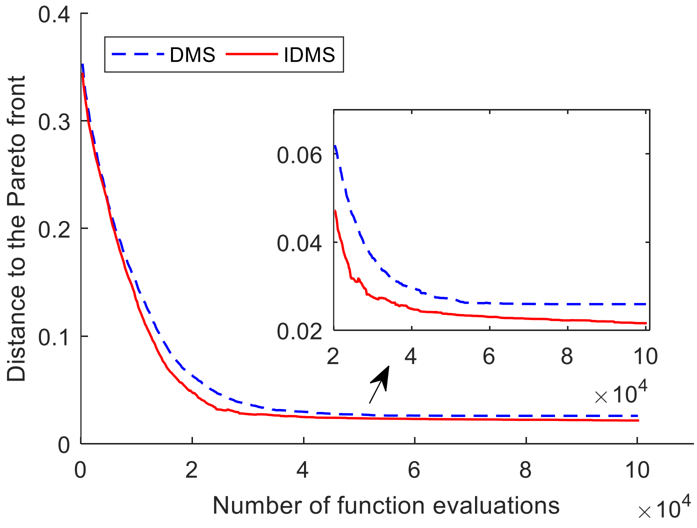
(a) LATEX



(b) ADPN



(c) PAPER



(d) SPIRA

Fig. 3. Convergence curves of the IDMS and DMS for the datasets.

5. Conclusions

Production datasets are generally imbalanced because the number of instances for various quality levels substantially differs. To select KQCs for imbalanced production data, this paper proposes a two-phase feature selection method. First, a concise KQC selection model is constructed as a bi-objective optimization problem of maximizing the feature (QC) importance and minimizing the percentage of selected features. In this model, we adopt the G-mean instead of accuracy to measure the importance of feature subsets because the G-mean is an unbiased metric for imbalanced data. Then, the IDMS-IPM is proposed to solve the model, with the IDMS used to search for a set of candidate solutions and the IPM used to select the final solutions from the candidate solution set. The experimental results demonstrate that the IDMS-IPM is effective for selecting KQCs because, compared with the benchmark methods, the method can significantly increase the sensitivity rates without greatly decreasing the specificity rates. Moreover, the IDMS-IPM can also effectively reduce the number of QCs. In the IDMS, a two-step mutation-based metaheuristic strategy is embedded in the DMS to improve the search ability. The experiments show that the search ability of the IDMS is improved compared with that of the DMS for KQC selection problems.

The proposed method is designed to solve KQC selection problems with two quality levels. In practice, products may be classified into more than two quality levels. Therefore, extending the proposed method to KQC selection problems with a larger number of quality levels is needed. The proposed method is performed in a supervised manner, in which the quality level of each training instance is given. We intend to extend the feature selection to semi-supervised KQC selection problems. The results in section 4.3 indicate the high time complexity of the proposed method. Thus, improving the time efficiency of the proposed method to enhance its practicability for large-scale data is an objective of future studies.

Acknowledgments

The authors would like to thank the editor and anonymous referees for the constructive comments and suggestions. They are also grateful to Ana Luisa Custódio for the code of the standard DMS method. This research was supported by the research project of the Tianjin Municipal Education Commission (No. 161072), *Research on Key Quality Characteristics Identification for Complex Products Based on Imbalanced Manufacturing Data*, and the National Natural Science Foundation of China (Grant nos. 71661147003 and 71401123).

References

Abido, M. A. (2006). Multiobjective evolutionary algorithms for electric power dispatch problem. *Evolutionary Computation, IEEE Transactions on, 10*(3), 315-329.

Ahila, R., Sadasivam, V., & Manimala, K. (2015). An integrated PSO for parameter determination and feature selection of ELM and its application in classification of power system disturbances. *Applied Soft Computing, 32*, 23-37.

Ahuja, J., & Ratnoo, S. (2015). Optimizing Feature Subset and Parameters for Support Vector Machine Using Multiobjective Genetic Algorithm. *Journal of Intelligent Systems, 24*(2), 145-160.

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

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

Bala, J., Huang, J., Vafaie, H., DeJong, K., & Wechsler, H. (1995). *Hybrid learning using genetic algorithms and decision trees for pattern classification.* Paper presented at the IJCAI (1).

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

Bertolazzi, P., Felici, G., Festa, P., Fiscon, G., & Weitschek, E. (2016). Integer programming models for feature selection: New extensions and a randomized solution algorithm. *European Journal of Operational Research, 250*(2), 389-399.

Custódio, A. L., Madeira, J. A., Vaz, A. I. F., & Vicente, L. N. (2011). Direct multisearch for multiobjective optimization. *SIAM Journal on Optimization, 21*(3), 1109-1140.

de Almeida Ribeiro, L., da Silva Soares, A., de Lima, T. W., Jorge, C. A. C., da Costa, R. M., Salvini, R. L., . . . Gabriel, P. H. R. (2015). Multi-objective Genetic Algorithm for Variable Selection in Multivariate Classification Problems: A Case Study in Verification of Biodiesel Adulteration. *Procedia Computer Science, 51*, 346-355.

de La Iglesia, B. (2013). Evolutionary computation for feature selection in classification problems. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 3*(6), 381-407.

Demšar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research, 7*(1), 1-30.

Du, W. S., & Hu, B. Q. (2018). A fast heuristic attribute reduction approach to ordered decision systems. *European Journal of Operational Research, 264*(2), 440-452.

Emmanouilidis, C., Hunter, A., & MacIntyre, J. (2000). *A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator.* Paper presented at the Evolutionary Computation, 2000. Proceedings of the 2000 Congress on.

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

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

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

Ghaddar, B., & Naoum-Sawaya, J. (2018). High dimensional data classification and feature selection using support vector machines. *European Journal of Operational Research, 265*(3), 993-1004.

Gunal, S., Gerek, O. N., Ece, D. G., & Edizkan, R. (2009). The search for optimal feature set in power quality event classification. *Expert Systems with Applications, 36*(7), 10266-10273.

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

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

Huang, C.-L., & Wang, C.-J. (2006). A GA-based feature selection and parameters optimization for support vector machines. *Expert Systems with applications, 31*(2), 231-240.

Huang, J., Cai, Y., & Xu, X. (2007). A hybrid genetic algorithm for feature selection wrapper based on mutual information. *Pattern Recognition Letters, 28*(13), 1825-1844.

Hughes, E. J. (2005). *Evolutionary many-objective optimisation: many once or one many?* Paper presented at the Evolutionary Computation, 2005. The 2005 IEEE Congress on.

Ishibuchi, H., Tsukamoto, N., & Nojima, Y. (2008). *Evolutionary many-objective optimization: A short review.* Paper presented at the IEEE congress on evolutionary computation.

Jain, I., Jain, V. K., & Jain, R. (2018). Correlation feature selection based improved-Binary Particle Swarm Optimization for gene selection and cancer classification. *Applied Soft Computing, 62*, 203-215.

Jeong, B., & Cho, H. (2006). Feature selection techniques and comparative studies for large-scale manufacturing processes. *The International Journal of Advanced Manufacturing Technology, 28*(9-10), 1006-1011.

John, G. H., & Langley, P. (1995). *Estimating continuous distributions in Bayesian classifiers.* Paper presented at the Proceedings of the Eleventh conference on Uncertainty in artificial intelligence.

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

Kubat, M., & Matwin, S. (1997). *Addressing the curse of imbalanced training sets: one-sided selection.* Paper presented at the ICML.

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

Lee, D. J., & Thornton, A. C. (1996). *The identification and use of key characteristics in the product development process.* Paper presented at the 1996 ASME Design Engineering Technical Conference.

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

Li, W., Pu, X., Tsung, F., & Xiang, D. (2017). A robust self-starting spatial rank multivariate EWMA chart based on forward variable selection. *Computers & Industrial Engineering, 103*, 116-130.

Liang, D., Tsai, C.-F., & Wu, H.-T. (2015). The effect of feature selection on financial distress prediction. *Knowledge-Based Systems, 73*, 289-297.

Maldonado, S., Pérez, J., & Bravo, C. (2017). Cost-based feature selection for Support Vector Machines: An application in credit scoring. *European Journal of Operational Research, 261*(2), 656-665.

Nag, K., & Pal, N. R. (2016). A Multiobjective Genetic Programming-Based Ensemble for Simultaneous Feature Selection and Classification. *IEEE transactions on cybernetics, 46* (2), 499-510.

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

Oliveira, L. S., Morita, M., & Sabourin, R. (2006). Feature selection for ensembles applied to handwriting recognition. *International Journal of Document Analysis and Recognition (IJDAR), 8*(4), 262-279.

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

Pierre, E. S., & Tuv, E. (2011). *Robust, non-redundant feature selection for yield analysis in semiconductor manufacturing.* Paper presented at the Industrial Conference on Data Mining.

Rodriguez, J. D., Perez, A., & Lozano, J. A. (2010). Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(3), 569-575.

Türkşen, Ö., Vieira, S. M., Madeira, J. F., Apaydin, A., & Sousa, J. M. (2013). Comparison of Multi-objective Algorithms Applied to Feature Selection *Towards Advanced Data Analysis by Combining Soft Computing and Statistics* (pp. 359-375): Springer.

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

Tian, W.-m., He, Z., & Yan, W. (2013). Key Process Variable Identification for Quality Classification Based on PLSR Model and Wrapper Feature Selection. In R. Dou (Ed.), *Proceedings of 2012 3rd International Asia Conference on Industrial Engineering and Management Innovation (IEMI2012)* (pp. 263-270). Berlin, Heidelberg: Springer Berlin Heidelberg.

Unler, A., & Murat, A. (2010). A discrete particle swarm optimization method for feature selection in binary classification problems. *European Journal of Operational Research, 206*(3), 528-539.

Wang, J., Zhao, Y., & Liu, P. (2010). *Effective feature selection with Particle Swarm Optimization based one-dimension searching.* Paper presented at the Systems and Control in Aeronautics and Astronautics (ISSCAA), 2010 3rd International Symposium on.

Wasikowski, M., & Chen, X.-W. (2010). Combating the small sample class imbalance problem using feature selection. *IEEE Transactions on knowledge and data engineering, 22*(10), 1388-1400.

Welikala, R., Fraz, M., Dehmeshki, J., Hoppe, A., Tah, V., Mann, S., . . . Barman, S. (2015). Genetic algorithm based feature selection combined with dual classification for the automated detection of proliferative diabetic retinopathy. *Computerized Medical Imaging and Graphics, 43*, 64-77.

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

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

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

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

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

Zhang, Y., Gong, D., Hu, Y., & Zhang, W. (2015). Feature selection algorithm based on bare bones particle swarm optimization. *Neurocomputing, 148*, 150-157.

Zhu, Z., Ong, Y.-S., & Dash, M. (2007). Wrapper–filter feature selection algorithm using a memetic framework. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 37*(1), 70-76.

Zitzler, E., & Thiele, L. (1998). *Multiobjective optimization using evolutionary algorithms — A comparative case study*, Berlin, Heidelberg.