A Pattern Mining Based Evolutionary Algorithm for Large-Scale Sparse Multi-Objective Optimization Problems

Ye Tian, Chang Lu, Xingyi Zhang, *Senior Member, IEEE*, Fan Cheng, and Yaochu Jin, *Fellow, IEEE*

***Abstract*—In real-world applications, there exist a lot of multi-objective optimization problems whose Pareto optimal solutions are sparse, i.e., most variables of these solutions are zero. Generally, many sparse multi-objective optimization problems (SMOPs) contain a large number of variables, which pose grand challenges for evolutionary algorithms to find the optimal solutions efficiently. To address the curse of dimensionality, this paper proposes an evolutionary algorithm for solving large-scale SMOPs, which aims to mine the sparse distribution of Pareto optimal solutions and thus considerably reduces the search space. More specifically, the proposed algorithm suggests an evolutionary pattern mining approach to detect the maximum and minimum candidate sets of the nonzero variables in Pareto optimal solutions, and use them to limit the dimensions in generating offspring solutions. For further performance enhancement, a binary crossover operator and a binary mutation operator are designed to ensure the sparsity of solutions. According to the experimental results on eight benchmark problems and four real-world problems, the proposed algorithm is superior over existing evolutionary algorithms in solving large-scale SMOPs.**

***Index Terms*—Sparse multi-objective optimization, evolutionary algorithm, pattern mining, genetic operator**

I. Introduction

MANY optimization problems in scientific and engineering areas are characterized by multiple objectives and sparse optimal solutions, which are known as sparse multi-objective optimization problems (SMOPs) [1]. The objectives of SMOPs are conflicting with each other to some extent, hence there does not exist a single solution making all the objectives optimal; instead, a set of trade-off solutions, known as Pareto optimal solutions can be found for SMOPs, in which the increase of one objective will lead to the deterioration of another. For example, the neural network training problem [2] aims to maximize the classification accuracy and minimize the model complexity, where a more complex model usually corresponds to a more powerful approximation ability and a higher risk of overfitting; the portfolio optimization problem [3] aims to maximize the expected return and minimize the risk, where a portfolio with higher return should take on more risk. Notably, the Pareto optimal solutions of SMOPs are sparse, i.e., most variables of these solutions are zero. In neural network training, sparse network structures are expected to alleviate overfitting [4], which means that most weights to be optimized should be zero; in portfolio optimization, only a small number of instruments can be selected to construct the portfolio [5].

SMOPs widely exist in many problems in science and technology including machine learning [6], data mining [7], software engineering [8], network science [9], signal processing [10], portfolio optimization [11], and power grid fault diagnosis [12]. Most real-world SMOPs are pursued based on a large dataset, which means that they are also large-scale multi-objective optimization problems (LMOPs). However, most multi-objective evolutionary algorithms (MOEAs) for solving general LMOPs are inefficient on SMOPs. This is because these MOEAs (e.g., MOEA/DVA [13] and LMEA [14]) consume a large number of function evaluations for determining the interactions between decision variables, making them impractical for solving SMOPs whose objective evaluation is relatively expensive. In addition, although some other MOEAs divide the decision variables randomly [15] or need not divide the decision variables [16], they are likely to get trapped in local optima [17] and cannot solve SMOPs with binary variables.

As a consequence, the strategies used for solving LMOPs are inapplicable to large-scale SMOPs. By contrast, it is desirable to reduce the difficulties of large-

Manuscript received –. This work was supported in part by the National Key Research and Development Project under Grant 2018AAA0100105, in part by the National Natural Science Foundation of China under Grant 61672033, Grant 61822301, Grant 61876123, Grant 61906001, and Grant U1804262, in part by the Hong Kong Scholars Program under Grant XJ2019035, in part by the Anhui Provincial Natural Science Foundation under Grant 1808085J06 and Grant 1908085QF271, in part by the State Key Laboratory of Synthetical Automation for Process Industries under Grant PAL-N201805, and in part by a Royal Society International Exchanges Program under Grant IEC\NSFC\170279. *(Corresponding author: Xingyi Zhang.)*

Y. Tian is with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, Institutes of Physical Science and Information Technology, Anhui University, Hefei 230601, China (email: field910921@gmail.com).

C. Lu, X. Zhang, and F. Cheng are with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230601, China (email: lucy_lc@outlook.com; xyzhanghust@gmail.com; chengfan@mail.ustc.edu.cn).

Y. Jin is with the Department of Computer Science, University of Surrey, Guildford, Surrey, GU2 7XH, U.K. (email: yaochu.jin@surrey.ac.uk).

scale SMOPs by factoring their sparse nature into the optimization process. Following this idea, this paper proposes a novel pattern mining approach to assist MOEAs in solving large-scale SMOPs. The core idea of the proposed approach is to estimate the sparse distribution of Pareto optimal solutions by data mining technique, thus highly reducing the decision space and alleviating the difficulties of large-scale SMOPs. In comparison to many dimensionality reduction techniques in machine learning, the proposed pattern mining approach is parameterless and has a low probability of being trapped in local optima. Specifically, this paper contains the following three main contributions:

1) An evolutionary pattern mining approach is proposed to mine the maximum and minimum candidate sets of the nonzero variables in Pareto optimal solutions, where the variables in a maximum candidate set indicate that these variables *could be zero or nonzero* in the Pareto optimal solutions, and the variables in a minimum candidate set indicate that these variables *should be nonzero* in the Pareto optimal solutions. When generating each offspring solution, the variables in the maximum candidate set is determined by genetic operators, the variables in the minimum candidate set are set to one, while all the other variables are fixed to zero. Therefore, the decision space searched by genetic operators is substantially reduced since only the variables in the maximum candidate set and outside the minimum candidate set need to be searched.
2) A binary crossover operator and a binary mutation operator are proposed to more effectively search for the optimal values of the variables in maximum candidate set. In contrast to the general operators that flip each decision variable with the same probability, the proposed operators flip each decision variable with different probabilities. The different flipping probabilities can ensure the sparsity of offspring solutions, while keeping the total crossover probability and mutation probability unchanged.
3) Based on the proposed pattern mining approach and genetic operators, an MOEA is developed for solving large-scale SMOPs, which performs the proposed genetic operators on only the dimensions determined by the maximum and minimum candidate sets mined by the proposed pattern mining approach. The proposed MOEA is tested on eight benchmark SMOPs and four real-world SMOPs, which exhibits significantly better performance than the existing MOEAs including those for LMOPs.

The rest of this paper is organized as follows. In Section II, some basic concepts related to sparse multi-objective optimization and pattern mining technique are given. In Section III, the proposed MOEA is described in detail. In Section IV, the experimental results are presented and analyzed. Finally, conclusions are drawn and future work is outlined in Section V.

II. Background

A. Sparse Multi-objective Optimization

An unconstrained multi-objective optimization problem can be mathematically defined as

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_M(\mathbf{x})) \\ \text{s.t.} \quad & \mathbf{x} \in \Omega \end{aligned}, \tag{1}$$

where $\mathbf{x} = (x_1, \cdots, x_D) \in \Omega$ is a solution consisting of D decision variables, $\Omega \subseteq \mathbb{R}^D$ is the decision space, $\mathbf{f} : \Omega \rightarrow \Lambda \subseteq \mathbb{R}^M$ consists of M objectives, and Λ is the objective space. A solution $\mathbf{x}$ is said to dominate another solution $\mathbf{y}$ if and only if $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ for every $i \in \{1, \cdots, M\}$ and $f_j(\mathbf{x}) < f_j(\mathbf{y})$ for at least one $j \in \{1, \cdots, M\}$. A solution is called a Pareto optimal solution if it is not dominated by any solution in Ω.

An LMOP usually refers to the MOPs having more than 100 decision variables. On the other hand, an SMOP indicates that most decision variables in the Pareto optimal solutions are zero; that is, the number of nonzero variables d in each Pareto optimal solution is much smaller than the total number of decision variables D. Since the decision space Ω exponentially increases with D, MOEAs usually need much more function evaluations for solving a problem with more decision variables, which is known as the curse of dimensionality [16]. However, due to the relatively expensive objective evaluation of many large-scale SMOPs, only the function evaluations sufficient for solving a d-variable problem may be available for solving a D-variable SMOP, and most existing MOEAs are inefficient for solving SMOPs since $d \ll D$.

Fortunately, the sparse nature of large-scale SMOPs is known in advance, which means that an MOEA can properly optimize only the nonzero variables. This way, the MOEA only needs to solve a problem with d variables rather than D variables, for which the available function evaluations are sufficient. However, the nonzero variables in the Pareto optimal solutions are of course unknown to the MOEA, hence some efficient techniques should be considered to find the correct nonzero variables during the evolutionary process. In recent years, the frequent pattern mining technique has received much attention and shown promising in many scenarios [18], which aims to mine the most frequent pattern in a transaction dataset for recommendation. For solving large-scale SMOPs, the transaction dataset refers to the Pareto optimal solutions and the most frequent pattern refers to the set of nonzero variables. Since the population progressively approximates the sparse Pareto optimal solutions during the evolutionary process, more and more variables in the population will become zero, and it is desirable to detect the remaining nonzero variables by performing pattern mining on the current population. By doing so, many zero variables can be

TABLE I
EXAMPLE OF THE CALCULATION OF *Support* AND *Occupancy*.

$\mathcal{T} = \{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3, \mathbf{t}_4, \mathbf{t}_5, \mathbf{t}_6\}$ $\mathbf{t}_1 = \{A, B, C, D, E, F\}$ $\mathbf{t}_2 = \{A, B, C, G\}$ $\mathbf{t}_3 = \{B, G\}$ $\mathbf{t}_4 = \{A, B, C\}$ $\mathbf{t}_5 = \{A, B, D, E, F\}$ $\mathbf{t}_6 = \{A, B, C\}$
$\mathbf{x} = \{A, B, C\}$, $\mathcal{T}_\mathbf{x} = \{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_4, \mathbf{t}_6\}$
$f_{supp}(\mathbf{x}) = \frac{4}{6}$, $f_{occu}(\mathbf{x}) = \frac{1}{4} \times (\frac{3}{6} + \frac{3}{4} + \frac{3}{3} + \frac{3}{3})$

ignored and the efficiency of finding the correct nonzero variables is significantly improved.

In the next subsection, an introduction to pattern mining technique and its advantages over other dimensionality reduction techniques are presented.

B. Pattern Mining

Since the concept of frequent pattern mining was proposed [19], much work has been dedicated to suggesting new objectives to evaluate the qualities of patterns, including *support* [20], *occupancy* [18], *area* [21], *utility* [22], just to name a few. The *support* and *occupancy* are the two most widely used objectives, which can be calculated by

$$\begin{aligned} \text{Maximize } f_{supp}(\mathbf{x}) &= \frac{|\mathcal{T}_\mathbf{x}|}{|\mathcal{T}|} \\ f_{occu}(\mathbf{x}) &= \frac{1}{|\mathcal{T}_\mathbf{x}|} \sum_{\mathbf{t} \in \mathcal{T}_\mathbf{x}} \frac{|\mathbf{x}|}{|\mathbf{t}|} \end{aligned}, \tag{2}$$

where $\mathbf{x}$ is the decision vector denoting a pattern, $\mathcal{T}$ denotes the transaction dataset to be mined, and $\mathcal{T}_\mathbf{x}$ denotes the set of transactions $\mathbf{x}$ occurs in:

$$\mathcal{T}_\mathbf{x} = \{\mathbf{t} \in \mathcal{T} | \mathbf{x} \subseteq \mathbf{t}\}. \tag{3}$$

The first objective $f_{supp}(\mathbf{x})$ is the *support* of $\mathbf{x}$ and the second objective $f_{occu}(\mathbf{x})$ is the *occupancy* of $\mathbf{x}$. Table I gives an example to illustrate the calculation of *support* and *occupancy*, where the transaction dataset $\mathcal{T}$ contains six transactions $\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_6$ consisting of seven items $A, B, \cdots, G$, and the solution $\mathbf{x}$ consists of three items A, B, C. In short, *support* indicates the frequency $\mathbf{x}$ occurs in the transaction dataset $\mathcal{T}$ and *occupancy* indicates the occupancy rate of $\mathbf{x}$ in $\mathcal{T}_\mathbf{x}$. Taking the recommendation task on e-commerce Website as an example, an item set (i.e., a pattern) has high *support* if it appears in many historical shopping lists, and it has high *occupancy* if it includes most items in each shopping list. Certainly, the most frequent and complete item set is expected to be found and suggested to customers [21]. However, *support* and *occupancy* are intuitively conflicting with each other since a pattern containing few items can appear in many shopping lists but occupy a small portion of these shopping lists, and vice versa.

Many heuristics have been proposed to solve the frequent pattern mining problem, such as the Apriori algorithm [23], the FP-Growth algorithm [24], the maximal frequent itemset algorithm [25], and the dominant and frequent itemset mining algorithm [18]. In recent years, evolutionary multi-objective optimization techniques have also been employed to solve the frequent pattern mining problem [7], [21]. In contrast to most heuristics, MOEAs can find various patterns in a single run without tuning thresholds or striking a balance between multiple objectives. Therefore, this work suggests an efficient evolutionary pattern mining approach to estimate the nonzero variables in Pareto optimal solutions. Although some other dimensionality reduction techniques in machine learning have been adopted in MOEAs such as principal component analysis [26] and restricted Boltzmann machine [27], evolutionary pattern mining has the following three advantages:

- The proposed evolutionary pattern mining approach can find the maximum and minimum candidate sets of the nonzero variables, rather than a direct approximation of the nonzero variables. Since the Pareto optimal solutions are not known a priori, a direct approximation of the nonzero variables based on the current non-dominated solutions may make the population more easily trapped in local optima. On the contrary, the maximum and minimum candidate sets of the nonzero variables are less greedy than the direct approximation, which can achieve a better balance between exploration and exploitation.
- By using the proposed evolutionary pattern mining approach, the proposed MOEA can find a set of maximum candidate sets and a set of minimum candidate sets, rather than a single maximum candidate set and a single minimum candidate set. When an offspring solution is generated, a maximum candidate set and a minimum candidate set are randomly selected and used to limit the dimensions of the offspring solution. By doing so, the population diversity can be enhanced and the probability of being trapped in local optima is further decreased.
- The proposed evolutionary pattern mining approach does not include any additional hyperparameters, and the degree of dimensionality reduction (i.e., the number of nonzero variables) also does not need to be predefined. By contrast, the degree of dimensionality reduction in many other techniques (e.g., the hidden layer size of restricted Boltzmann machine) should be predefined, which is difficult to be determined in advance since the Pareto optimal solutions are unknown.

As a consequence, the proposed evolutionary pattern mining approach is parameter-less and quite flexible. To the best of our knowledge, only few single-objective evolutionary algorithms have adopted traditional pattern mining approach (which is not based on evolutionary

Algorithm 1: Framework of the proposed PM-MOEA

Input: N (population size), N' (population size for evolutionary pattern mining)
Output: P (final population)
$P \leftarrow$ Randomly generate N solutions; `//main population`
$Q_{max} \leftarrow \emptyset$; `//max candidate sets of nonzero variables`
$Q_{min} \leftarrow \emptyset$; `//min candidate sets of nonzero variables`
while *termination criterion not fulfilled* **do**
  $Q_{max} \leftarrow MiningMaxSets(P, Q_{max}, N')$;
  $Q_{min} \leftarrow MiningMinSets(P, Q_{min}, N')$;
  $P' \leftarrow$ Select $2N$ parents via binary tournament selection based on the non-dominated front number of solutions in P;
  $P'' \leftarrow Variation(P', Q_{max}, Q_{min})$;
  $P \leftarrow EnvironmentalSelection(P \cup P'', N)$;
return P;

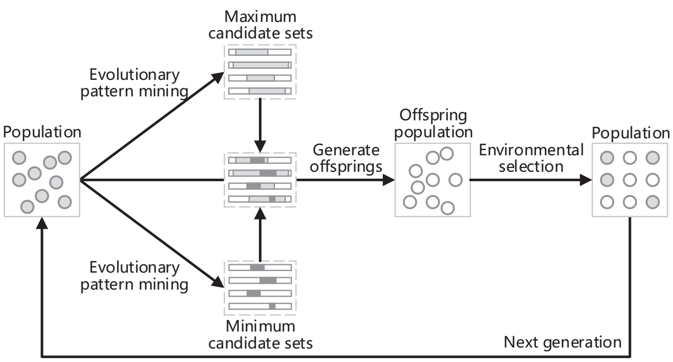

Fig. 1. Diagram of one generation of the proposed PM-MOEA.

algorithm) to establish operators [28] or tune the parameters of operators [29], while this work is the first time to develop an evolutionary pattern mining approach for dimensionality reduction in large-scale optimization. In the next section, the procedure of the proposed MOEA is described in detail.

III. The Proposed Algorithm

A. Framework of PM-MOEA

The framework of PM-MOEA is given in Algorithm 1, which starts with the random initialization of a population with size N. Then, the maximum candidate sets of nonzero variables and the minimum candidate sets of nonzero variables are set to empty. In each generation of PM-MOEA, the maximum and minimum candidate sets are mined according to the non-dominated solutions in the current population. Afterwards, a number of parents are selected via binary tournament selection and offspring solutions are generated in the dimensions determined by the maximum and minimum candidate sets. Lastly, the environmental selection is performed to select N solutions from the combination of the current population and the offspring population. The procedure of one generation of PM-MOEA is illuminated in Fig 1.

As shown in Algorithm 2, the environmental selection of PM-MOEA is similar to many other MOEAs such as SPEA2 [30] and Two_Arch2 [31]. Specifically, the combined population is first sorted by non-dominated sorting [32], and the solutions in the first k non-dominated fronts are selected, where k is the smallest value satisfying $|F_1 \cup F_2 \cup \cdots \cup F_k| \geq N$. For the solutions in the last front F_k, the solution having the minimum Euclidean distance to the other solutions in objective space is removed one by one, until $|F_1 \cup F_2 \cup \cdots \cup F_k| = N$. Thus, the population for next generation consists of all the solutions in $F_1 \cup F_2 \cup \cdots \cup F_k$, which contains N solutions in total.

Algorithm 2: $EnvironmentalSelection(P, N)$

Input: P (combined population), N (population size)
Output: P (population for next generation)
$[F_1, F_2, \cdots] \leftarrow$ Do non-dominated sorting on P; `//`F_i` is the solution set in the `i`-th non-dominated front`
$k \leftarrow \text{argmin}_i |F_1 \cup F_2 \cup \cdots \cup F_i| \geq N$;
while $|F_1 \cup F_2 \cup \cdots \cup F_k| > N$ **do**
  $\mathbf{p} \leftarrow \text{argmin}_{\mathbf{x} \in F_k} \min_{\mathbf{y} \in F_k \setminus \{\mathbf{x}\}} \|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\|$;
  $F_k \leftarrow F_k \setminus \{\mathbf{p}\}$;
$P \leftarrow F_1 \cup F_2 \cup \cdots \cup F_k$;
return P;

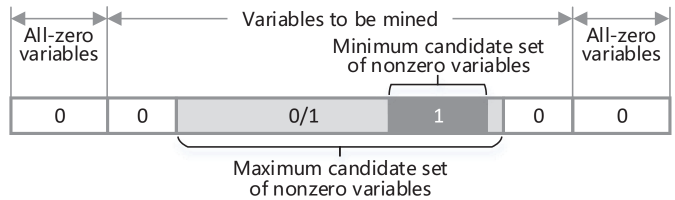

Fig. 2. Illustration of different parts of a decision vector. The variables in the maximum candidate set means that they are possibly nonzero variables, which should be searched by genetic operators. The variables in the minimum candidate set means that they should be nonzero variables, which are fixed to 1 and do not need to be searched. The remaining variables should be zero variables, which are fixed to 0 and also do not need to be searched.

It is noteworthy that some other effective selection strategies [33], [34] can also be adopted in PM-MOEA, since the environmental selection is independent of the core components of PM-MOEA, i.e., evolutionary pattern mining and genetic operators. In the next two subsections, the two components of PM-MOEA are described in detail.

B. The Proposed Evolutionary Pattern Mining Approach

The proposed PM-MOEA represents each solution $\mathbf{x}$ by a real vector dec and a binary vector $mask$ instead of the decision variables $x_1, \ldots, x_D$, and

$$x_i = dec_i \times mask_i, \quad i = 1, \ldots, D \tag{4}$$

where dec_i indicates the real value of the i-th decision variable and $mask_i$ indicates whether the i-th decision variable is zero. Based on this encoding scheme, the whole decision vector is divided into several parts as illustrated in Fig. 2. The all-zero variables indicate that these variables are zero in all the non-dominated solutions in the current population; that is, these variables will be fixed to 0 in the $mask$ of all the offspring solutions, which does not need to be mined by the proposed evolutionary pattern mining approach. For the other variables to be mined, the variables in the maximum candidate set indicate that these variables could be zero or nonzero, hence binary genetic operators will be performed on $mask$ to search for the best value (i.e., 0 or 1) for each of these variables. Besides, the variables in the minimum candidate set indicate that these variables should be nonzero, hence these variables will be fixed to 1 in the $mask$ of all the offspring solutions. In short, for the $mask$ of each offspring solution, the variables in the maximum candidate set are generated by binary genetic operators, the variables in the minimum candidate set are fixed to 1, and all the other variables are fixed to 0.

Now the core task is to mine the maximum and minimum candidate sets from the non-dominated solutions in the current population. Since the variables in a maximum candidate set could be zero or nonzero, the variables that are nonzero in *at least a few* non-dominated solutions should be considered in the maximum candidate set. By contrast, since the variables in a minimum candidate set should be nonzero, the variables that are nonzero in *most* non-dominated solutions can be considered in the minimum candidate set. For this aim, we define two objectives for mining the maximum candidate sets by the proposed evolutionary approach:

$$\begin{aligned} \text{Maximize } f_1(\mathbf{x}) &= \frac{|P'_{\mathbf{x}}|}{|P|} \\ f_2(\mathbf{x}) &= \frac{1}{|P'_{\mathbf{x}}|}\sum_{\mathbf{p}\in P'_{\mathbf{x}}}\frac{\sum \mathbf{p}.mask}{\sum \mathbf{x}}, \end{aligned} \tag{5}$$

where $\mathbf{x}$ is a binary vector denoting a maximum candidate set, P denotes the set of non-dominated solutions in the current population, and $P'_{\mathbf{x}}$ denotes the set of solutions whose nonzero variables in $mask$ is the subset of the nonzero variables in $\mathbf{x}$:

$$P'_{\mathbf{x}} = \{\mathbf{p} \in P \mid \mathbf{p}.mask_i \leq x_i,\ i = 1, 2, \cdots\}. \tag{6}$$

On the other hand, the two objectives for mining the minimum candidate sets by the proposed evolutionary approach are defined as

$$\begin{aligned} \text{Maximize } f_3(\mathbf{y}) &= \frac{|P_{\mathbf{y}}|}{|P|} \\ f_4(\mathbf{y}) &= \frac{1}{|P_{\mathbf{y}}|}\sum_{\mathbf{p}\in P_{\mathbf{y}}}\frac{\sum \mathbf{y}}{\sum \mathbf{p}.mask}, \end{aligned} \tag{7}$$

where $\mathbf{y}$ is a binary vector denoting a minimum candidate set and $P_{\mathbf{y}}$ denotes the set of solutions whose nonzero variables in $mask$ is the superset of the nonzero variables in $\mathbf{y}$:

$$P_{\mathbf{y}} = \{\mathbf{p} \in P \mid \mathbf{p}.mask_i \geq y_i,\ i = 1, 2, \cdots\}. \tag{8}$$

An example illustrating the calculation of the four objectives is given in Table II.

The definitions of f_3 and f_4 are the same to *support* and *occupancy* in (2), respectively, while the definitions of f_1 and f_2 are different from existing ones. As a consequence, the objectives for mining maximum candidate sets aim to find a set of nonzero variables covering more solutions, i.e., a maximum candidate set can approximately be a *union set* of the nonzero variables in all solutions. By contrast, the objectives for mining minimum candidate sets aim to find a set of nonzero variables existing in more solutions, i.e., a minimum candidate set can approximately be an *intersection set* of the nonzero variables in all solutions. That is, a maximum candidate set defines a *upper bound* of the nonzero variable set, since the variables in it could be zero or nonzero; while a minimum candidate set defines a *lower bound* of the nonzero variable set, since the variables in it are probably be nonzero. Obviously, mining the upper bound and lower bound of the nonzero variable set is much less greedy than directly mining the nonzero variable set, where the later is likely to make the population trapped in local optima. The experimental results in Fig. 6 can demonstrate the superiority of mining both maximum and minimum candidate sets over mining just a single one.

TABLE II
EXAMPLE OF THE CALCULATION OF f_1, f_2, f_3, AND f_4.

$P = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5, \mathbf{p}_6\}$
$\mathbf{p}_1.mask = \{1\ 1\ 1\ 1\ 1\ 1\ 0\}$
$\mathbf{p}_2.mask = \{1\ 1\ 1\ 0\ 0\ 0\ 1\}$
$\mathbf{p}_3.mask = \{0\ 1\ 0\ 0\ 0\ 0\ 1\}$
$\mathbf{p}_4.mask = \{1\ 1\ 1\ 0\ 0\ 0\ 0\}$
$\mathbf{p}_5.mask = \{1\ 1\ 0\ 1\ 1\ 1\ 0\}$
$\mathbf{p}_6.mask = \{1\ 1\ 1\ 0\ 0\ 0\ 0\}$
$\mathbf{x} = \{1\ 1\ 1\ 0\ 0\ 0\ 1\}$, $P'_{\mathbf{x}} = \{\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_6\}$
$\mathbf{y} = \{1\ 1\ 1\ 0\ 0\ 0\ 0\}$, $P_{\mathbf{y}} = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_4, \mathbf{p}_6\}$
$f_1(\mathbf{x}) = \frac{4}{6}$, $f_2(\mathbf{x}) = \frac{1}{4} \times (\frac{4}{4} + \frac{2}{4} + \frac{3}{4} + \frac{3}{4})$
$f_3(\mathbf{y}) = \frac{4}{6}$, $f_4(\mathbf{y}) = \frac{1}{4} \times (\frac{3}{6} + \frac{3}{4} + \frac{3}{3} + \frac{3}{3})$

Algorithm 3 details the procedure of using the proposed evolutionary approach to mine maximum candidate sets. To begin with, the dominated solutions are removed from the population (Line 1) and the all-zero variables are removed (Lines 2–5). Then, the maximum candidate sets from last generation are extended by several new ones, where each new candidate set is a union set of the nonzero variables in the $mask$ of several randomly selected non-dominated solutions (Lines 6–9). In each generation of the evolutionary approach, the maximum candidate sets are evolved by the same mating selection, genetic operators, and environmental selection as Algorithm 1. As shown in Algorithm 4, the procedure of using the proposed evolutionary approach to mine minimum candidate sets is very similar to Algorithm 3, where the main difference is that Algorithm 4 evaluates each minimum candidate set by (7) but not (5). Fig. 3 depicts the current population, three maximum

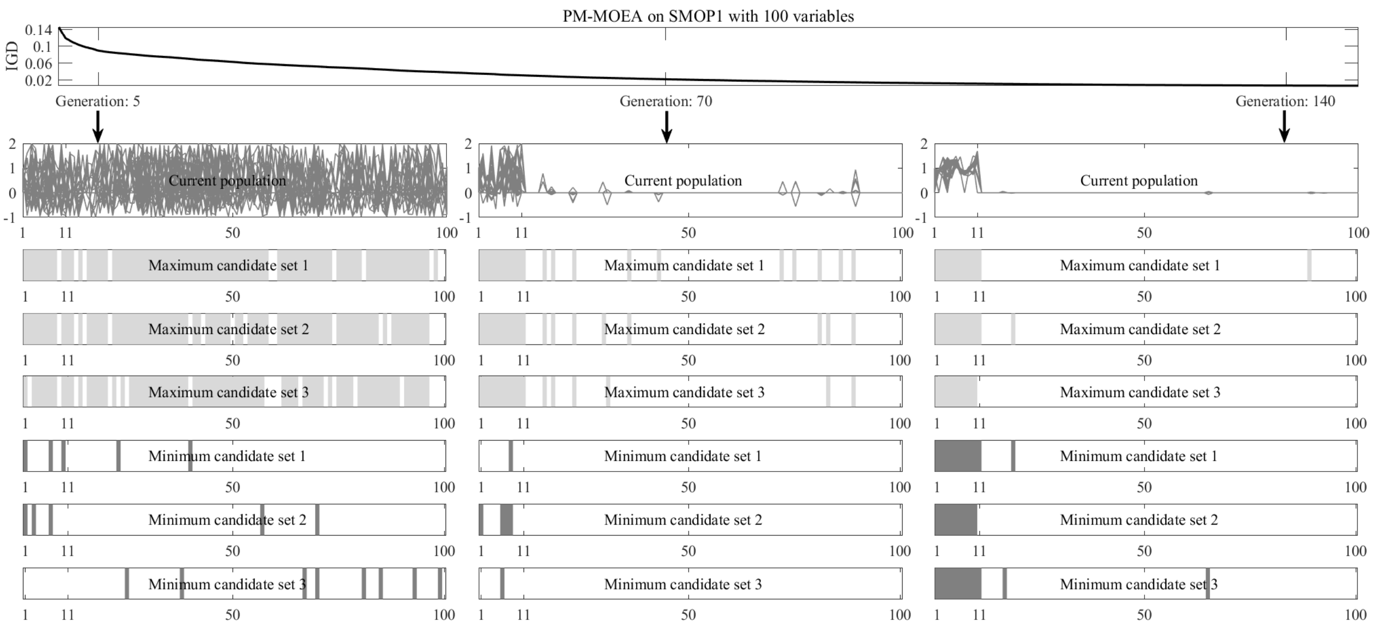

Fig. 3. The current population, three maximum candidate sets, and three minimum candidate sets at generation 5 (left column), 70 (middle column), and 140 (right column) in one run of PM-MOEA on SMOP1 with 100 variables. The 1st to 11th variables in Pareto optimal solutions are nonzero, and all the other variables in Pareto optimal solutions are zero.

Algorithm 3: $MingingMaxSets(P, Q_{max}, N')$

Input: P (current population), Q_{max} (maximum candidate sets of nonzero variables), N' (population size for evolutionary pattern mining)
Output: Q_{max} (new maximum candidate sets of nonzero variables)

$P \leftarrow$ Delete dominated solutions from P;
$Zero \leftarrow$ Set of variables that are zero in all the solutions in P;
$P \leftarrow$ Delete the dimensions that are in $Zero$ from the solutions in P;
if $Q_{max} \neq \emptyset$ **then**
  $Q_{max} \leftarrow$ Delete the dimensions that are in $Zero$ from the solutions in Q_{max};
for $i = 1$ *to* N' **do**
  $P_{sub} \leftarrow$ Randomly select half the solutions from P;
  $\mathbf{q} \leftarrow$ Set of variables that are nonzero in at least one solution in P_{sub};
  $Q_{max} \leftarrow Q_{max} \cup \{\mathbf{q}\}$;
Evaluate Q_{max} by (5) based on P;
for $generation = 1$ *to* *10* **do**
  $Q'_{max} \leftarrow$ Select $2N'$ parents via binary tournament selection based on the non-dominated front number of solutions in Q_{max};
  $Q''_{max} \leftarrow BinaryOperators(Q'_{max})$;
  Evaluate Q''_{max} by (5) based on P;
  $Q_{max} \leftarrow EnvironmentalSelection(Q_{max} \cup Q''_{max}, N')$;
return Q_{max};

Algorithm 4: $MingingMinSets(P, Q_{min}, N')$

Input: P (current population), Q_{min} (minimum candidate sets of nonzero variables), N' (population size for evolutionary pattern mining)
Output: Q_{min} (new minimum candidate sets of nonzero variables)

$P \leftarrow$ Delete dominated solutions from P;
$Zero \leftarrow$ Set of variables that are zero in all the solutions in P;
$P \leftarrow$ Delete the dimensions that are in $Zero$ from the solutions in P;
if $Q_{min} \neq \emptyset$ **then**
  $Q_{min} \leftarrow$ Delete the dimensions that are in $Zero$ from the solutions in Q_{min};
for $i = 1$ *to* N' **do**
  $P_{sub} \leftarrow$ Randomly select half the solutions from P;
  $\mathbf{q} \leftarrow$ Set of variables that are nonzero in all solutions in P_{sub};
  $Q_{min} \leftarrow Q_{min} \cup \{\mathbf{q}\}$;
Evaluate Q_{min} by (7) based on P;
for $generation = 1$ *to* *10* **do**
  $Q'_{min} \leftarrow$ Select $2N'$ parents via binary tournament selection based on the non-dominated front number of solutions in Q_{min};
  $Q''_{min} \leftarrow BinaryOperators(Q'_{min})$;
  Evaluate Q''_{min} by (7) based on P;
  $Q_{min} \leftarrow EnvironmentalSelection(Q_{min} \cup Q''_{min}, N')$;
return Q_{min};

candidate sets, and three minimum candidate sets at generation 5, 70, and 140 in one run of PM-MOEA on SMOP1 with 100 variables. It is clear that with the decrease of the IGD [35] value, the population approximates the Pareto optimal solutions with more and more variables becoming zero. Accordingly, the maximum and minimum candidate sets are consistent with the sparse distribution of the variables in the current population, hence the candidate sets can also approximate the nonzero variable set of the Pareto optimal solutions and thus accelerate the convergence of the population.

It is worth noting that the proposed evolutionary pattern mining approach should be performed twice in each generation of PM-MOEA, but it does not highly increase the computational complexity. On one hand, the maximum and minimum candidate sets from last generation are used as the initial candidate sets, hence the proposed evolutionary approach does not need to search from scratch. On the other hand, since the transac-

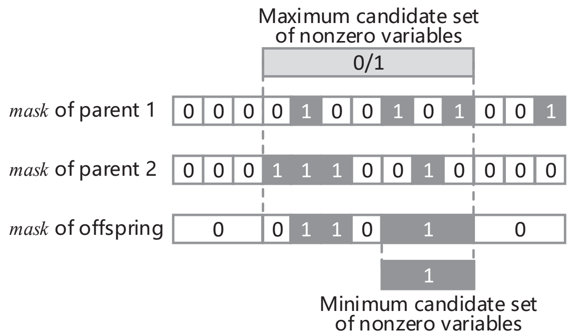

Fig. 4. Illustration of generating the $mask$ of an offspring solution based on a maximum candidate set and a minimum candidate set.

tion dataset is the $mask$ of the non-dominated solutions excluding all-zero variables, the search space is relatively small and the evaluation of each candidate set is very efficient. Besides, the proposed evolutionary approach does not introduce any evaluation of the original SMOP since the four objectives are pursued based on only the decision variables of the non-dominated solutions. In the experiments, the population size and the number of generations of the proposed evolutionary pattern mining approach are set to just 20 and 10, respectively, and the runtime of PM-MOEA is competitive to the other compared MOEAs as shown in the experimental results in Table VI.

C. The Proposed Genetic Operators

In general, the $mask$ of each offspring solution is generated by performing the proposed binary genetic operators on the $mask$ of two parents, and the dec of each offspring solution is generated by performing simulated binary crossover [36] and polynomial mutation [37] on the dec of the same parents. Besides, the generation of $mask$ is guided by maximum and minimum candidate sets as illustrated in Fig. 4. As presented in Algorithm 5, the proposed PM-MOEA selects two parents to generate an offspring solution each time (Lines 3–4). For generating the $mask$ of the offspring solution, the proposed binary genetic operators are performed on the dimensions of a randomly selected maximum candidate set (Lines 5–7), and the variables in the dimensions of a randomly selected minimum candidate set are set to 1 (Lines 8–9). For generating the dec of the offspring solution, the simulated binary crossover and polynomial mutation are directly performed on all the dimensions (Line 10).

Before we give the details of the proposed binary operators, the bit-flip mutation is analyzed as an example to illustrate the ineffectiveness of existing operators in solving SMOPs. Assuming a solution to be mutated contains D variables, where n variables are nonzero and $(D-n)$ variables are zero. After the mutation is performed with a probability of p, the expectation of the number of nonzero variables will be changed to

$$\begin{aligned} E^1_{nonzero} =& n + p(D-n) - pn \\ =& (1-2p)n + pD. \end{aligned} \tag{9}$$

Algorithm 5: $Variation(P', Q_{max}, Q_{min})$

Input: P' (parent population), Q_{max} (maximum candidate sets of nonzero variables), Q_{min} (minimum candidate sets of nonzero variables)
Output: P'' (offspring population)
1 $P'' \leftarrow \emptyset$;
2 **while** $P' \neq \emptyset$ **do**
//Select two parents to generate one offspring solution
3 $[\mathbf{p}_1, \mathbf{p}_2] \leftarrow$ Randomly select two parents from P';
4 $P' \leftarrow P' \setminus \{\mathbf{p}_1, \mathbf{p}_2\}$;
//Generate the $mask$ of the offspring solution on the dimensions of a maximum candidate set
5 $\mathbf{q}_{max} \leftarrow$ Randomly select a maximum candidate set from Q_{max};
6 $\mathbf{o}.mask \leftarrow$ a vector of zeros;
7 $\mathbf{o}.mask(\mathbf{q}_{max}) \leftarrow BinaryOperators(\{\mathbf{p}_1.mask(\mathbf{q}_{max}), \mathbf{p}_2.mask(\mathbf{q}_{max})\})$; //$\mathbf{o}.mask(\mathbf{q}_{max})$ denotes the variables in dimensions $\mathbf{q}_{max}$ of $\mathbf{o}.mask$
//Generate the $mask$ of the offspring solution on the dimensions of a minimum candidate set
8 $\mathbf{q}_{min} \leftarrow$ Randomly select a minimum candidate set from Q_{min};
9 $\mathbf{o}.mask(\mathbf{q}_{min}) \leftarrow 1$; //$\mathbf{o}.mask(\mathbf{q}_{min})$ denotes the variables in dimensions $\mathbf{q}_{min}$ of $\mathbf{o}.mask$
//Generate the dec of the offspring solution
10 $\mathbf{o}.dec \leftarrow$ Perform simulated binary crossover and polynomial mutation based on $\mathbf{p}_1.dec$ and $\mathbf{p}_2.dec$;
11 $P'' \leftarrow P'' \cup \{\mathbf{o}\}$;
12 **return** P'';

Hence, if the mutation is performed for i times, the expectation of the number of nonzero variables will be

$$\begin{aligned} E^i_{nonzero} =& (1-2p)^i n + pD + pD(1-2p) + \cdots + pD(1-2p)^{i-1} \\ =& (1-2p)^i n + pD\frac{1-(1-2p)^i}{1-(1-2p)} \end{aligned}. \tag{10}$$

When $i \to \infty$, the expectation of the number of nonzero variables will be $\frac{D}{2}$. That is, approximately half the variables will be nonzero after a sufficient number of generations, no matter what the initial solution is. Since only a few variables are nonzero in the Pareto optimal solutions of SMOPs, existing operators can hardly search for these optimal solutions. To address this issue, we make the expectation of the number of flipped nonzero variables equal to the expectation of the number of flipped zero variables, thus the sparsity of solutions can be ensured. In other words, we make $p_1 n = p_0(D-n)$, where p_1 and p_0 denote the probabilities to flip each nonzero variable and each zero variable, respectively. Since $p_1 n + p_0(D-n) = pD$, we have

$$\begin{aligned} p_1 =& \frac{pD}{2n} \\ p_0 =& \frac{pD}{2(D-n)} \end{aligned}. \tag{11}$$

Furthermore, a very sparse solution (i.e., $n \ll D$) may make $p_1 > 1$, hence the mutation probability p is set to $\frac{2n}{D}$ if it is larger than this value.

Algorithm 6 presents the procedure of the proposed binary operators. For each offspring solution, it is first set to the same to the first parent (Line 4). Then, the crossover operator flips each variable in the dimensions where the two parents are different with the probability

Algorithm 6: $BinaryOperators(P')$

Input: P' (parent population)
Output: P'' (offspring population)
1 $P'' \leftarrow \emptyset$;
2 **while** $P' \neq \emptyset$ **do**
  //The proposed binary crossover
3   $[\mathbf{p}_1, \mathbf{p}_2] \leftarrow$ Randomly select two parents from P';
4   $\mathbf{o} \leftarrow \mathbf{p}_1$;
5   $diff \leftarrow$ xor$(\mathbf{p}_1, \mathbf{p}_2)$; //The variables that are different in $\mathbf{p}_1$ and $\mathbf{p}_2$
6   $pro_{diff} \leftarrow$ Calculate the probability to flip each variable in dimensions $diff$ of $\mathbf{o}$ by (11);
7   Flip each variable in dimensions $diff$ of $\mathbf{o}$ with the probability determined by pro_{diff};
  //The proposed binary mutation
8   $pro \leftarrow$ Calculate the probability to flip each variable of $\mathbf{o}$ by (11);
9   Flip each variable of $\mathbf{o}$ with the probability determined by pro;
10   $P'' \leftarrow P'' \cup \{\mathbf{o}\}$;
11 **return** P'';

calculated by (11) (Line 7). Lastly, the mutation operator flips each variable with the probability calculated by (11) (Line 9).

IV. Experimental Studies

This section empirically verifies the effectiveness of the proposed PM-MOEA on eight benchmark SMOPs [1] and four SMOPs in real-world applications, namely, neural network training [6], instance selection [38], critical node detection [39], and portfolio optimization [3]. We select NSGA-II [40], WOF-SMPSO [15], LMOCSO [16], and SparseEA [1] as the baseline algorithms, where NSGA-II is one of the most classical multi-objective genetic algorithms, and WOF-SMPSO and LMOCSO are two state-of-the-art multi-objective particle swarm optimization algorithms tailored for LMOPs. Besides, SparseEA is currently the only MOEA for solving SMOPs, but it is ineffective for solving large-scale SMOPs since it does not use any dimensionality reduction technique. All the experiments are implemented on PlatEMO [41].

A. Settings of Problems

Table III lists the parameter settings of the involved benchmark SMOPs and real-world SMOPs, including the types of variables, the number of decision variables, the sparsity of the benchmark SMOPs, and the datasets used in the real-world SMOPs. The eight benchmark problems SMOP1–SMOP8 are with sparse Pareto optimal solutions and various difficulties including multi-modality, deception, epistasis, and low intrinsic dimensionality, which are challenging for existing MOEAs. The mathematical definitions of them are referred to [1]. Besides, the definitions of the four real-world problems are given in the following.

Firstly, the neural network training problem aims to find the optimal weights for the lowest classification error, which is defined as [6]

$$\min \begin{array}{l} f_1(\mathbf{x}) = \dfrac{\|\mathbf{x}\|_0}{D} \\ f_2(\mathbf{x}) = ErrorRate(\mathbf{x}) \end{array}, \tag{12}$$

where $\mathbf{x}$ is the decision vector denoting the weights of a three-layer feedforward neural network, $\|\mathbf{x}\|_0$ denotes the number of nonzero weights, D denotes the total number of weights, and $ErrorRate(\mathbf{x})$ denotes the error rate of the neural network. The objective f_1 indicates the ratio of nonzero weights, and the objective f_2 indicates the error rate of the neural network on training set. Secondly, the instance selection problem aims to select the fewest training samples for the lowest classification error, which is defined as [38]

$$\min \begin{array}{l} f_1(\mathbf{x}) = \dfrac{|\mathbf{x}|}{D} \\ f_2(\mathbf{x}) = ErrorRate(\mathbf{x}) \end{array}, \tag{13}$$

where $\mathbf{x}$ is the decision vector denoting a set of selected samples, D denotes the total number of samples, and $ErrorRate(\mathbf{x})$ denotes the error rate of the classifier. The objective f_1 indicates the ratio of selected samples, and the objective f_2 indicates the error rate of the classifier on validation set. Thirdly, the critical node detection problem aims to delete the fewest nodes for the largest destruction of a graph, which is defined as [39]

$$\min \begin{array}{l} f_1(\mathbf{x}) = \dfrac{|\mathbf{x}|}{D} \\ f_2(\mathbf{x}) = \dfrac{Connect(G[V \backslash \mathbf{x}])}{Connect(G)} \end{array}, \tag{14}$$

where $\mathbf{x}$ is the decision vector denoting a set of deleted nodes, D denotes the total number of nodes in the graph, and $Connect(G)$ denotes the pairwise connectivity of graph G:

$$Connect(G) = \sum_{i \in V} \sum_{\substack{j \in V \\ j \neq i}} Con(G, i, j), \tag{15}$$

where $Con(G, i, j)$ denotes whether node i is reachable from node j in G:

$$Con(G, i, j) = \begin{cases} 1, & \text{if } i \text{ is reachable from } j \\ 0, & \text{otherwise} \end{cases}. \tag{16}$$

The objective f_1 indicates the ratio of deleted nodes, and the objective f_2 indicates the ratio of the pairwise connectivity of the remaining graph to that of the original graph. Lastly, the portfolio optimization problem aims to find the portfolio of instruments with the largest expected return and the lowest risk, which is defined as [3]

$$\min \begin{array}{l} f_1(\mathbf{x}) = \displaystyle\sum_{i=1}^{D} \sum_{j=1}^{D} \mathbf{x}_i \sigma_{i,j} \mathbf{x}_j \\ f_2(\mathbf{x}) = 1 - \displaystyle\sum_{i=1}^{D} \mathbf{x}_i r_i \end{array}, \tag{17}$$

TABLE III
PARAMETER SETTINGS OF EIGHT BENCHMARK SMOPS AND FOUR REAL-WORLD SMOPS.

Benchmark problem	Types of variables	No. of variables (D)	No. of objectives	Sparsity of Pareto optimal solutions		
SMOP1 SMOP2 SMOP3 SMOP4 SMOP5 SMOP6 SMOP7 SMOP8	Real	100 500 1000 5000	2	0.1		
Neural network training problem	**Type of variables**	**No. of variables (D)**	**Dataset**	**Size of hidden layer**	**No. of features**	**No. of classes**
NN1	Real	101	Climate Model Simulation Crashes[1]	5	18	2
NN2		521	Statlog/German[2]	20	24	2
NN3		1241	Connectionist Bench Sonar[1]	20	60	2
NN4		6241	LSVT Voice Rehabilitation[1]	20	310	2
Instance selection problem	**No. of variables**	**No. of variables (D)**	**Dataset**	**No. of samples**	**No. of features**	**No. of classes**
IS1	Binary	195	Parkinsons[1]	195	22	2
IS2		540	Climate Model Simulation Crashes[1]	540	18	2
IS3		846	Vehicle[3]	846	18	2
IS4		5300	Banana[3]	5300	2	2
Critical node detection problem	**Type of variables**	**No. of variables (D)**	**Dataset**	**No. of nodes**	**No. of edges**	
CN1	Binary	235	ER235[4]	235	350	
CN2		466	ER466[4]	466	700	
CN3		941	ER941[4]	941	1400	
CN4		4941	US power grid [42]	4941	6594	
Portfolio optimization problem	**Type of variables**	**No. of variables (D)**	**Dataset**	**No. of instruments**	**Length of each instrument**	
PO1	Real	100	EURCHF[5]	100	50	
PO2		500	EURCHF[5]	500	50	
PO3		1000	EURCHF[5]	1000	50	
PO4		5000	EURCHF[5]	5000	50	

1. https://archive.ics.uci.edu/ml/datasets.php
2. https://www.csie.ntu.edu.tw/%7ecjlin/libsvmtools/datasets/binary.html
3. https://sci2s.ugr.es/keel/category.php?cat=clas&order=ins#sub2
4. http://individual.utoronto.ca/mventresca/cnd.html
5. https://www.metatrader5.com/en

where $\mathbf{x}$ is the decision vector denoting the ratio of each instrument in the portfolio, $\sigma_{i,j}$ denotes the covariance between the i-th and j-th instruments, and r_i denotes the expected return of the i-th instrument. The objective f_1 indicates the total risk of the portfolio, and f_2 indicates the negative value of the expected return.

B. Settings of Algorithms

1) Operators: For NSGA-II, SparseEA, and the proposed PM-MOEA, they use simulated binary crossover [36] and polynomial mutation [37] to generate the real variables of offspring solutions, where the crossover probability is set to 1, the mutation probability is set to $1/D$ (D denotes the number of variables), and the distribution index of both crossover and mutation is set to 20. Besides, NSGA-II uses single-point crossover and bit-flip mutation to generate the binary variables of offspring solutions, while SparseEA and PM-MOEA use their own binary genetic operators. For WOF-SMPSO and LMOCSO, they generate the real variables of offspring solutions by particle swarm optimization and competitive swarm optimizer, respectively. In order to generate the binary variables of offspring solutions, WOF-SMPSO and LMOCSO optimize real variables within $[0, 1]$ and round the variables before calculating objective values.

2) Population size: In all the compared MOEAs, the population size is set to 100 for solving benchmark SMOPs and 50 for solving real-world SMOPs.

3) Maximum number of function evaluations: In all the compared MOEAs, the maximum number of function evaluations is set to $150 \times D$ for solving SMOPs with real variables and $100 \times D$ for solving SMOPs with binary variables.

4) Other parameters: In WOF-SMPSO, the number of groups, the number of evaluations for original problem, the number of evaluations for transformed problem, the number of chosen solutions for weight optimization, and the fraction of evaluations for weight optimization are set to 4, 1000, 500, 3, 0.5, respectively. In PM-MOEA, the population size and the number of generations of the evolutionary pattern mining approach are set to 20 and 10, respectively.

C. Performance of PM-MOEA on Benchmark SMOPs

Table IV lists the mean and standard deviation of the IGD values [35] obtained by NSGA-II, WOF-SMPSO, LMOCSO, SparseEA, and the proposed PM-MOEA on SMOP1–SMOP8 with 100 to 5000 decision variables, averaged over 30 runs. Each IGD value is calculated based on one population with respect to approximately 10000 reference points uniformly sampled on the true Pareto front [43]. In addition, the Wilcoxon rank sum test [44] with a significance level of 0.05 is adopted to

TABLE IV
IGD VALUES OBTAINED BY NSGA-II, WOF-SMPSO, LMOCSO, SPARSEEA, AND THE PROPOSED PM-MOEA ON SMOP1–SMOP8 WITH 100 TO 5000 DECISION VARIABLES. THE BEST RESULT IN EACH ROW IS HIGHLIGHTED.

Problem	D	NSGA-II	WOF-SMPSO	LMOCSO	SparseEA	PM-MOEA
SMOP1	100	8.6956e-2 (1.54e-2) −	2.7201e-1 (3.22e-2) −	3.9278e-1 (2.91e-2) −	7.8464e-3 (1.50e-3) −	5.3914e-3 (1.50e-3)
	500	1.4275e-1 (6.74e-3) −	2.5109e-1 (2.12e-2) −	4.7584e-1 (2.12e-2) −	1.5518e-2 (2.63e-3) −	1.0332e-2 (2.60e-3)
	1000	1.8273e-1 (7.62e-3) −	2.2479e-1 (2.62e-2) −	4.9111e-1 (1.54e-2) −	2.4364e-2 (2.20e-3) −	1.5347e-2 (1.73e-3)
	5000	3.3307e-1 (1.07e-2) −	1.8214e-1 (1.67e-2) −	5.3713e-1 (1.63e-2) −	3.6494e-2 (1.36e-3) −	3.0367e-2 (1.00e-3)
SMOP2	100	4.8717e-1 (7.30e-2) −	8.7620e-1 (1.54e-1) −	1.5316e+0 (6.93e-2) −	2.6258e-2 (7.53e-3) −	1.5835e-2 (4.37e-3)
	500	7.7352e-1 (3.02e-2) −	3.5279e-1 (1.55e-1) −	1.6478e+0 (6.44e-2) −	4.2796e-2 (4.91e-3) −	3.1189e-2 (5.89e-3)
	1000	8.8971e-1 (2.57e-2) −	2.3584e-1 (9.36e-2) −	1.6755e+0 (6.94e-2) −	6.2346e-2 (6.59e-3) −	4.2868e-2 (5.62e-3)
	5000	1.1223e+0 (1.65e-2) −	1.7635e-1 (1.16e-2) −	1.7021e+0 (3.92e-2) −	9.4302e-2 (2.10e-3) −	7.7687e-2 (2.72e-3)
SMOP3	100	7.7752e-1 (2.67e-2) −	7.1767e-1 (1.27e-2) −	1.5303e+0 (9.34e-2) −	1.3303e-2 (2.49e-3) ≈	2.9654e-2 (4.13e-2)
	500	9.3078e-1 (3.57e-2) −	7.0295e-1 (2.90e-3) −	1.6276e+0 (3.89e-2) −	1.7596e-2 (3.52e-3) −	1.3618e-2 (2.56e-3)
	1000	1.0854e+0 (2.56e-2) −	7.0188e-1 (1.38e-3) −	1.6423e+0 (2.83e-2) −	2.5076e-2 (3.15e-3) +	5.5470e-2 (1.72e-1)
	5000	1.4919e+0 (1.98e-2) −	7.0172e-1 (1.52e-3) −	1.6945e+0 (1.17e-2) −	3.9696e-2 (1.67e-3) +	6.7110e-2 (1.70e-1)
SMOP4	100	1.3090e-1 (3.74e-2) −	3.9212e-1 (8.86e-2) −	7.4140e-1 (4.84e-2) −	4.6830e-3 (2.33e-4) −	4.0730e-3 (6.06e-5)
	500	3.2139e-1 (1.90e-2) −	7.4637e-2 (6.13e-2) −	8.2032e-1 (4.57e-2) −	4.7577e-3 (3.03e-4) −	4.0736e-3 (7.87e-5)
	1000	3.9383e-1 (1.17e-2) −	6.8366e-2 (5.97e-2) −	8.2778e-1 (2.76e-2) −	4.7020e-3 (2.66e-4) −	4.0698e-3 (7.84e-5)
	5000	5.3756e-1 (6.62e-3) −	9.4936e-3 (8.44e-3) −	8.7138e-1 (4.52e-2) −	4.7641e-3 (2.13e-4) −	4.0731e-3 (6.52e-5)
SMOP5	100	3.6399e-1 (1.77e-3) −	3.6590e-1 (6.17e-3) −	4.0619e-1 (5.47e-3) −	5.5153e-3 (2.53e-4) −	4.8756e-3 (3.37e-4)
	500	3.6651e-1 (9.32e-4) −	3.5787e-1 (1.53e-3) −	4.2110e-1 (4.28e-3) −	5.4773e-3 (2.72e-4) −	4.5610e-3 (1.92e-4)
	1000	3.7233e-1 (2.08e-3) −	3.5440e-1 (9.60e-4) −	4.2548e-1 (3.59e-3) −	5.4475e-3 (2.34e-4) −	4.4602e-3 (1.54e-4)
	5000	3.9886e-1 (2.24e-3) −	3.4918e-1 (5.09e-4) −	4.3274e-1 (2.41e-3) −	5.4380e-3 (1.95e-4) −	4.5937e-3 (7.92e-5)
SMOP6	100	3.4129e-2 (3.56e-3) −	7.8553e-2 (3.28e-3) −	1.3717e-1 (9.08e-3) −	6.6293e-3 (4.26e-4) −	6.1198e-3 (7.32e-4)
	500	4.1042e-2 (2.43e-3) −	7.2309e-2 (3.82e-3) −	1.7041e-1 (6.09e-3) −	6.4243e-3 (4.67e-4) −	4.6834e-3 (1.42e-4)
	1000	5.3202e-2 (2.32e-3) −	7.3342e-2 (8.66e-3) −	1.7144e-1 (4.38e-3) −	6.4011e-3 (2.36e-4) −	4.5470e-3 (8.98e-5)
	5000	9.8106e-2 (2.83e-3) −	3.2781e-2 (1.52e-2) −	1.8294e-1 (3.47e-3) −	6.7850e-3 (2.47e-4) −	4.7824e-3 (7.31e-5)
SMOP7	100	2.8344e-1 (3.08e-2) −	1.2963e-1 (1.47e-2) −	4.6431e-1 (4.67e-2) −	3.1206e-2 (9.23e-3) −	2.1092e-2 (7.31e-3)
	500	3.4063e-1 (1.94e-2) −	9.2099e-2 (5.42e-3) −	5.3564e-1 (3.83e-2) −	5.1675e-2 (6.92e-3) −	4.2147e-2 (1.01e-2)
	1000	3.9978e-1 (1.80e-2) −	8.7758e-2 (5.91e-3) −	5.5419e-1 (3.86e-2) −	7.8756e-2 (8.05e-3) −	6.0005e-2 (5.19e-3)
	5000	5.9142e-1 (1.41e-2) −	7.9518e-2 (5.62e-3) +	5.9117e-1 (5.26e-2) −	1.1773e-1 (4.85e-3) −	1.0088e-1 (5.38e-3)
SMOP8	100	1.3677e+0 (1.02e-1) −	7.3186e-1 (4.56e-2) −	2.1116e+0 (1.98e-1) −	1.5241e-1 (3.04e-2) ≈	1.3882e-1 (3.95e-2)
	500	1.6394e+0 (5.01e-2) −	5.6992e-1 (2.36e-2) −	2.0862e+0 (1.17e-1) −	2.0315e-1 (2.39e-2) −	1.5795e-1 (1.54e-2)
	1000	1.7909e+0 (4.12e-2) −	5.4238e-1 (7.80e-3) −	2.1822e+0 (1.22e-1) −	2.3000e-1 (1.48e-2) −	1.8579e-1 (1.51e-2)
	5000	2.2011e+0 (3.23e-2) −	5.3296e-1 (2.80e-3) −	2.2979e+0 (5.45e-2) −	3.0769e-1 (8.27e-3) −	2.6568e-1 (1.19e-2)
+/ − / ≈		0/32/0	1/31/0	0/32/0	2/28/2	

perform statistical analysis, where '+', '−' and '≈' indicate that the result obtained by an MOEA is significantly better, significantly worse, and statistically similar to that obtained by the proposed PM-MOEA, respectively.

As shown in Table IV, the proposed PM-MOEA exhibits obviously better performance than the other compared MOEAs on SMOP1–SMOP8, having achieved the best IGD values on 28 of 32 test instances. In terms of Wilcoxon rank sum test, PM-MOEA is significantly better than NSGA-II, WOF-SMPSO, LMOCSO, and SparseEA on 32, 31, 32, and 28 test instances, respectively. To illustrate the superiority of PM-MOEA visually, Fig. 5 shows the parallel coordinates plot of the decision variables of solutions obtained by the compared MOEAs on SMOP1, SMOP2, and SMOP7 with 1000 variables, where all the variables outside the gray region are zero in the Pareto optimal solutions. For NSGA-II and LMOCSO, it is obvious that most variables of the obtained solutions are far from zero. For WOF-SMPSO, most variables of the obtained solutions are close to zero. As for SparseEA and the proposed PM-MOEA, most variables of the obtained solutions are equal to zero, and the solutions obtained by PM-MOEA are sparser than those obtained by SparseEA. As a consequence, PM-MOEA is superior over existing MOEAs in solving benchmark SMOPs.

D. Effectiveness of the Components in PM-MOEA

The superiority of the proposed PM-MOEA is mainly attributed to its two components, i.e., the evolutionary pattern mining approach and the new genetic operators. To verify the effectiveness of these two components, we compare PM-MOEA to its three variants, where the first variant PM-MOEA′ mines only the maximum candidate sets of nonzero variables (i.e., without minimum candidate sets), the second variant PM-MOEA″ mines only the minimum candidate sets of nonzero variables (i.e., without maximum candidate sets), and the third variant PM-MOEA‴ uses single-point crossover and bit-flip mutation instead of the proposed ones (i.e., without the proposed genetic operators).

Fig. 6 depicts the convergence profiles of IGD values obtained by PM-MOEA and its three variants on SMOP1, SMOP2, and SMOP7 with 1000 variables, averaged over 30 runs. It can be observed from the figure that the original PM-MOEA converges faster than its variants on the three test instances, which indicates that both the evolutionary pattern mining approach and the new genetic operators are effective for solving SMOPs. Besides, the PM-MOEA′ without minimum candidate sets

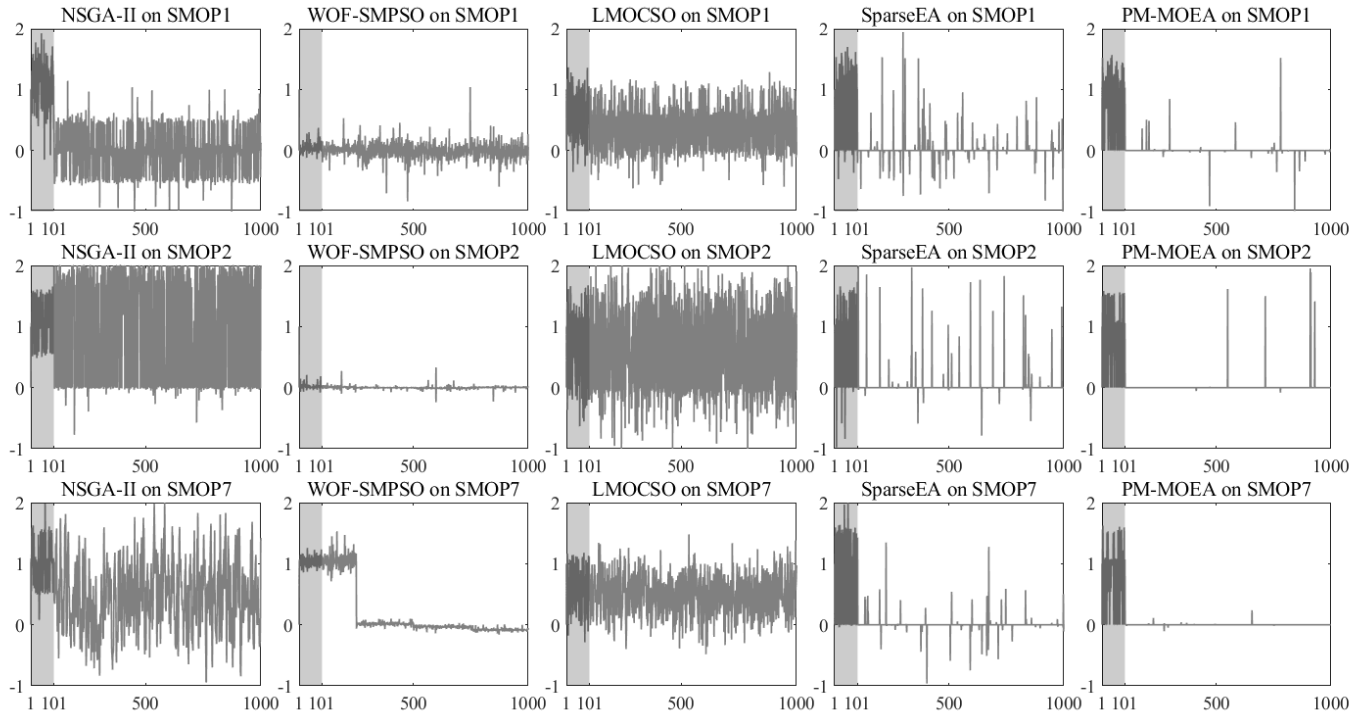

Fig. 5. Parallel coordinates plot of the decision variables of solutions obtained by NSGA-II, WOF-SMPSO, LMOCSO, SparseEA, and the proposed PM-MOEA on SMOP1, SMOP2, and SMOP7 with 1000 variables. All the variables outside the gray region are zero in the Pareto optimal solutions.

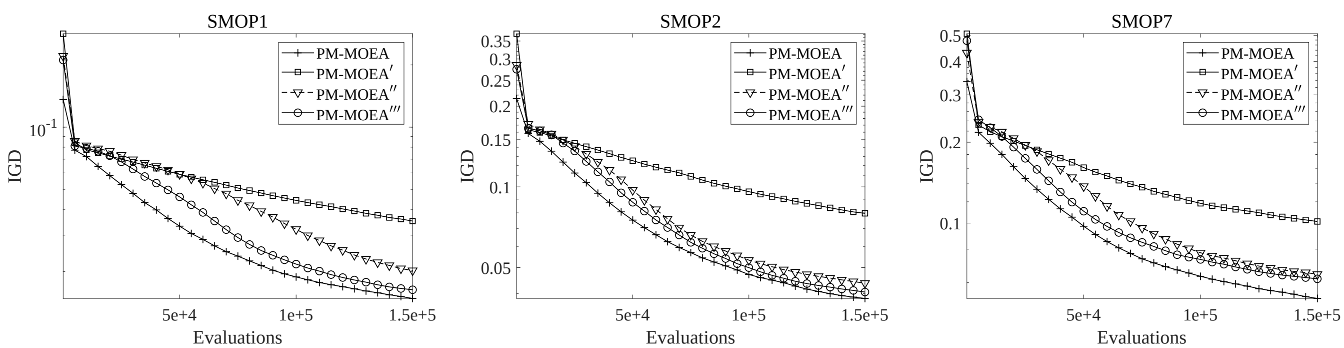

Fig. 6. Convergence profiles of IGD values obtained by PM-MOEA, PM-MOEA′ (without minimum candidate set), PM-MOEA′′ (without maximum candidate set), and PM-MOEA′′′ (without the proposed genetic operators) on SMOP1, SMOP2, and SMOP7 with 1000 variables.

and the PM-MOEA′′ without maximum candidate sets converge slower than the PM-MOEA′′′ without the proposed genetic operators, which means that the evolutionary pattern mining approach is more important than the new genetic operators for PM-MOEA.

E. Performance of PM-MOEA on Real-World SMOPs

In this subsection, the proposed PM-MOEA is compared to the other four MOEAs on real-world SMOPs. Table V lists the mean and standard deviation of the HV values [45] obtained by NSGA-II, WOF-SMPSO, LMOCSO, SparseEA, and PM-MOEA on the neural network training problem (NN1–NN4), the instance selection problem (IS1–IS4), the critical node detection problem (CN1–CN4), and the portfolio optimization problem (PO1–PO4). Since the true Pareto fronts of the real-world SMOPs are unknown, the HV value with respect to a reference point $(1, 1)$ is calculated instead of IGD. Besides, the Wilcoxon rank sum test is also performed.

According to Table V, it can be found that the performance of the proposed PM-MOEA is obviously better than the other compared MOEAs. Specifically, PM-MOEA obtains the best HV values on 14 test instances, WOF-SMPSO and SparseEA perform the best on 1 test instance, respectively, while NSGA-II and LMOCSO cannot gain any best result. Furthermore, Fig. 7 plots the objective values of solutions obtained by the compared MOEAs on NN3, IS3, CN3, and PO3. For the neural network training problem (NN3), the solutions obtained by SparseEA and PM-MOEA are much sparser than those obtained by NSGA-II, WOF-SMPSO, and

TABLE V
HV VALUES OBTAINED BY NSGA-II, WOF-SMPSO, LMOCSO, SPARSEEA, AND THE PROPOSED PM-MOEA ON NEURAL NETWORK TRAINING, INSTANCE SELECTION, CRITICAL NODE DETECTION, AND PORTFOLIO OPTIMIZATION. THE BEST RESULT IN EACH ROW IS HIGHLIGHTED.

Problem	D	NSGA-II	WOF-SMPSO	LMOCSO	SparseEA	PM-MOEA
NN1	101	3.4562e-1 (3.87e-2) −	3.4671e-1 (3.52e-2) −	3.2190e-1 (1.84e-2) −	9.2214e-1 (1.22e-16) ≈	9.2214e-1 (1.22e-16)
NN2	521	3.0252e-1 (1.79e-2) −	2.9173e-1 (1.34e-2) −	2.8454e-1 (1.65e-2) −	7.9868e-1 (4.17e-3) ≈	8.0090e-1 (3.01e-3)
NN3	1241	3.2708e-1 (1.61e-2) −	3.2876e-1 (1.58e-2) −	3.1794e-1 (1.14e-2) −	8.7106e-1 (1.64e-2) ≈	8.7479e-1 (7.94e-3)
NN4	6241	3.3821e-1 (1.46e-2) −	3.2939e-1 (2.44e-2) −	3.3937e-1 (1.78e-2) −	9.4882e-1 (1.47e-2) ≈	9.5181e-1 (7.34e-3)
IS1	195	8.8652e-1 (7.53e-3) −	9.0810e-1 (4.07e-3) ≈	7.1424e-1 (3.72e-2) −	9.0718e-1 (3.75e-3) ≈	9.1178e-1 (6.05e-3)
IS2	540	9.5563e-1 (4.36e-3) −	9.7135e-1 (1.69e-3) −	7.4010e-1 (3.48e-2) −	9.7356e-1 (1.78e-3) ≈	9.7464e-1 (1.23e-3)
IS3	846	7.9562e-1 (9.52e-3) −	8.1590e-1 (8.13e-3) ≈	4.8488e-1 (6.88e-2) −	8.0355e-1 (5.98e-3) −	8.1632e-1 (6.07e-3)
IS4	5300	6.0055e-1 (4.51e-2) −	6.8173e-1 (3.24e-4) +	3.4918e-1 (4.99e-2) −	6.5037e-1 (1.21e-4) ≈	6.5037e-1 (7.93e-5)
CN1	235	8.1952e-1 (2.03e-2) −	8.6417e-1 (6.25e-3) −	6.9135e-1 (3.75e-2) −	9.0390e-1 (3.38e-3) ≈	9.0453e-1 (5.89e-3)
CN2	466	8.7662e-1 (5.42e-3) −	8.5170e-1 (5.58e-3) −	6.9411e-1 (2.64e-2) −	9.0319e-1 (3.06e-3) −	9.1202e-1 (1.82e-3)
CN3	941	8.6440e-1 (7.37e-3) −	8.4038e-1 (4.16e-3) −	6.8505e-1 (1.37e-2) −	8.8349e-1 (6.71e-3) −	9.0483e-1 (3.09e-3)
CN4	4941	8.0403e-1 (7.92e-3) −	9.5899e-1 (5.85e-3) −	7.0578e-1 (3.69e-2) −	9.8496e-1 (8.02e-4) −	9.9100e-1 (7.96e-4)
PO1	100	9.9197e-2 (7.93e-4) −	1.0048e-1 (1.65e-3) −	1.1933e-1 (4.14e-4) −	1.2379e-1 (5.39e-4) −	1.2443e-1 (3.16e-4)
PO2	500	9.3289e-2 (1.85e-4) −	9.3992e-2 (3.87e-4) −	1.1430e-1 (4.39e-4) −	1.2286e-1 (1.41e-3) −	1.2449e-1 (4.11e-4)
PO3	1000	9.2310e-2 (9.63e-5) −	9.2983e-2 (3.80e-4) −	1.1095e-1 (5.32e-4) −	1.2325e-1 (2.56e-3) ≈	1.2491e-1 (1.06e-4)
PO4	5000	9.1280e-2 (1.72e-5) −	9.1543e-2 (9.88e-5) −	1.0236e-1 (3.68e-4) −	1.2494e-1 (6.12e-9) ≈	1.2494e-1 (3.86e-9)
+/ − / ≈		0/16/0	1/13/2	0/16/0	0/6/10	

LMOCSO, and the solutions obtained by PM-MOEA dominate those obtained by SparseEA. For the instance selection problem (IS3), the solutions obtained by PM-MOEA can dominate most solutions obtained by the other MOEAs. For the critical node detection problem (CN3), PM-MOEA exhibits competitive performance to NSGA-II, where the solutions obtained by PM-MOEA have better spread than those obtained by NSGA-II. As for the portfolio optimization problem (PO3), the solutions obtained by PM-MOEA have obviously better convergence and spread than those obtained by the other MOEAs. To summarize, the proposed PM-MOEA is also superior over existing MOEAs in solving real-world SMOPs.

F. Computational Efficiency of PM-MOEA

Lastly, the computational efficiency of PM-MOEA is compared to the other four MOEAs. Table VI lists the runtimes (in second) of the five MOEAs on the benchmark SMOPs and real-world SMOPs with approximately 5000 decision variables. It can be observed that the efficiency of PM-MOEA is worse than the other MOEAs on SMOP1–SMOP8, NN4, and CN4, competitive to the other MOEAs on PO4, and better than the other MOEAs on IS4. Since PM-MOEA needs to perform the proposed evolutionary pattern mining approach many times, its runtime is slightly longer than the other MOEAs. While for some real-world SMOPs such as IS4, PM-MOEA is more efficient than the other MOEAs, since PM-MOEA can obtain much sparser solutions and a sparser solution corresponds to a cheaper objective evaluation of these SMOPs. As a consequence, the computational efficiency of PM-MOEA is not obviously worse than the existing MOEAs.

TABLE VI
RUNTIMES (IN SECOND) OF NSGA-II, WOF-SMPSO, LMOCSO, SPARSEEA, AND THE PROPOSED PM-MOEA ON SMOP1–SMOP8, NEURAL NETWORK TRAINING, INSTANCE SELECTION, CRITICAL NODE DETECTION, AND PORTFOLIO OPTIMIZATION. LEAST RUNTIME IN EACH ROW IS HIGHLIGHTED.

Problem	D	NSGA-II	WOF-SMPSO	LMOCSO	SparseEA	PM-MOEA
SMOP1–SMOP8 (average)	5000	8.1795e+3	8.0752e+3	5.9150e+3	1.7970e+4	4.2445e+4
NN4	6241	3.8762e+3	5.9592e+3	3.7517e+3	5.8890e+3	7.0076e+3
IS4	5300	3.0129e+4	4.4549e+4	1.3506e+5	7.6770e+3	4.8045e+3
CN4	4941	6.4473e+4	4.3592e+4	3.1588e+4	8.7610e+4	1.5105e+5
PO4	5000	2.5306e+5	1.0287e+4	1.6584e+4	1.0142e+4	1.8444e+4

V. CONCLUSIONS AND FUTURE WORK

In order to address the curse of dimensionality in solving large-scale SMOPs, this paper has proposed a pattern mining based evolutionary algorithm to mine the sparse distribution of Pareto optimal solutions. To be specific, an evolutionary pattern mining approach has been proposed to mine the nonzero variables from the current population. In contrast to traditional approaches that reduce the decision space directly, the proposed approach finds a set of maximum candidate sets and a set of minimum candidate sets of the nonzero variables, where each offspring solution is generated inside the dimensions determined by a randomly selected maximum candidate set and a randomly selected minimum candidate set. The proposed approach can not only highly reduce the decision space, but also enhance the population diversity and decrease the probability of being trapped into local optimum. Moreover, a binary crossover operator and a binary mutation operator have

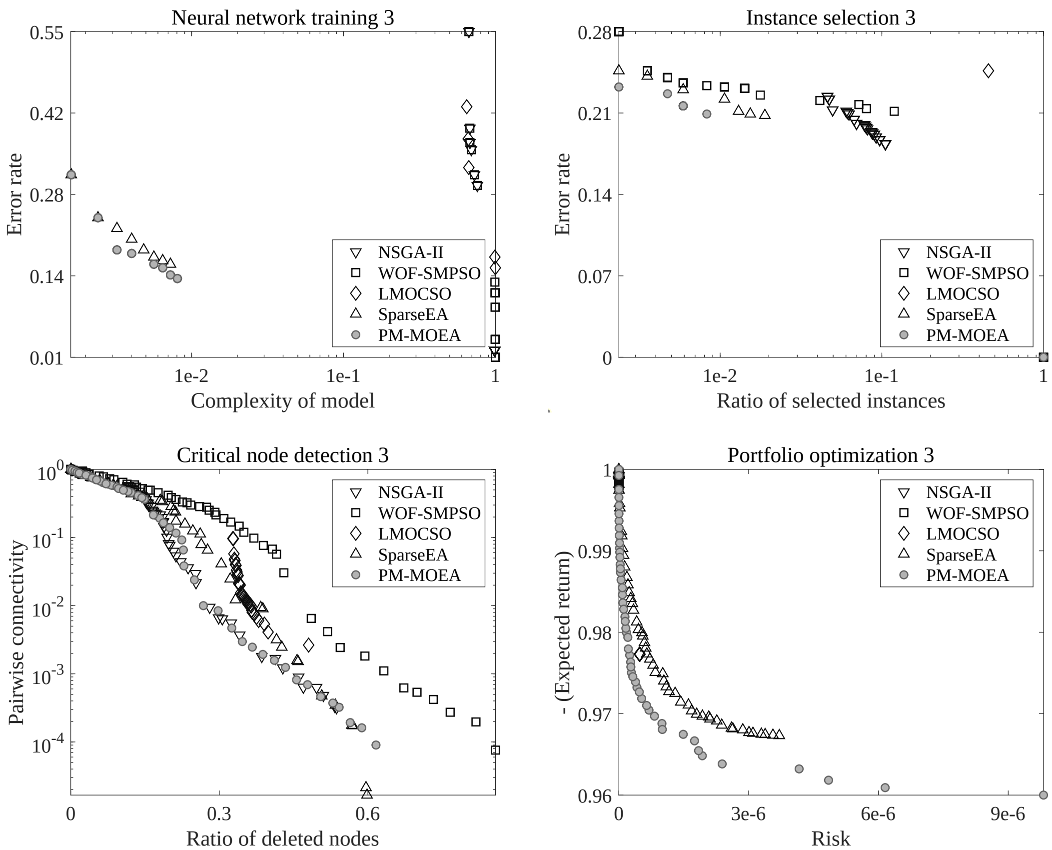

Fig. 7. Objective values of solutions obtained by NSGA-II, WOF-SMPSO, LMOCSO, SparseEA, and the proposed PM-MOEA on neural network training, instance selection, critical node detection, and portfolio optimization.

been proposed to ensure the sparsity of offspring solutions.

To verify the performance of the proposed MOEA, it has been compared to some state-of-the-art MOEAs on eight benchmark SMOPs and four real-world SMOPs, including neural network training, instance selection, critical node detection, and portfolio optimization. Experimental results have demonstrated that the proposed MOEA can effectively ensure the sparsity of solutions on the tested problems, resulting in populations with better convergence and diversity than those obtained by existing MOEAs.

This work has shown the promising prospect of pattern mining approach in solving large-scale SMOPs, and further exploration on the potential of this approach is highly desirable. On one hand, the proposed MOEA mines useful information from the binary variables of solutions due to the restriction of the objective functions of pattern mining, and new objective functions can be designed to mine useful information from the real variables of solutions. On the other hand, the performance of the proposed MOEA has been verified on SMOPs with up to 6000 variables, and it is reasonable to combine metaheuristics with heuristics to develop more efficient pattern mining approach to solve SMOPs with much more variables.

References

[1] Y. Tian, X. Zhang, C. Wang, and Y. Jin, "An evolutionary algorithm for large-scale sparse multi-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, 2019, in press.

[2] Y. Jin, T. Okabe, and B. Sendhoff, "Neural network regularization and ensembling using multi-objective evolutionary algorithms," in *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, 2004, pp. 1–8.

[3] A. Ponsich, A. L. Jaimes, and C. A. C. Coello, "A survey on multi-objective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 3, pp. 321–344, 2013.

[4] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[5] J. Branke, B. Scheckenbach, M. Stein, K. Deb, and H. Schmeck, "Portfolio optimization with an envelope-based multi-objective evolutionary algorithm," *European Journal of Operational Research*, vol. 199, no. 3, pp. 684–693, 2009.

[6] Y. Jin and B. Sendhoff, "Pareto-based multiobjective machine learning: An overview and case studies," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 3, pp. 397–415, 2008.

[7] Y. Tian, S. Yang, L. Zhang, F. Duan, and X. Zhang, "A surrogate-assisted multi-objective evolutionary algorithm for large-scale task-oriented pattern mining," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 3, no. 2, pp. 106–116, 2019.

[8] Y. Xiang, Y. Zhou, Z. Zheng, and M. Li, "Configuring software product lines by combining many-objective optimization and SAT solvers," *ACM Transactions on Software Engineering and Methodology*, vol. 26, no. 4, p. 14, 2018.

[9] L. Zhang, H. Pan, Y. Su, X. Zhang, and Y. Niu, "A mixed representation-based multiobjective evolutionary algorithm for overlapping community detection," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2703–2716, 2017.

[10] H. Li, Q. Zhang, J. Deng, and Z.-B. Xu, "A preference-based multiobjective evolutionary approach for sparse optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1716–1731, 2018.

[11] K. Lwin, R. Qu, and G. Kendall, "A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization," *Applied Soft Computing*, vol. 24, pp. 757–772, 2014.

[12] S. Wang and D. Zhao, "Power grid fault diagnosis based on immune clonal constrained multi-objective optimization method," *Power System Technology*, vol. 41, no. 12, pp. 4061–4068, 2017.

[13] X. Ma, F. Liu, Y. Qi, X. Wang, L. Li, L. Jiao, M. Yin, and M. Gong, "A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 2, pp. 275–298, 2016.

[14] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 97–112, 2018.

[15] H. Zille, H. Ishibuchi, S. Mostaghim, and Y. Nojima, "A framework for large-scale multi-objective optimization based on problem transformation," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 260–275, 2018.

[16] Y. Tian, X. Zheng, X. Zhang, and Y. Jin, "Efficient large-scale multi-objective optimization based on a competitive swarm optimizer," *IEEE Transactions on Cybernetics*, vol. 2019, in press.

[17] C. He, L. Li, Y. Tian, X. Zhang, R. Cheng, Y. Jin, and X. Yao, "Accelerating large-scale multi-objective optimization via problem reformulation," *IEEE Transactions on Evolutionary Computation*, 2019, in press.

[18] L. Tang, L. Zhang, P. Luo, and M. Wang, "Incorporating occupancy into frequent pattern mining for high quality pattern recommendation," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 2012, pp. 75–84.

[19] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *ACM Sigmod Record*, vol. 22, no. 2, pp. 207–216, 1993.

[20] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: Current status and future directions," *Data Mining and Knowledge Discovery*, vol. 15, no. 1, pp. 55–86, 2007.

[21] X. Zhang, F. Duan, L. Zhang, F. Cheng, Y. Jin, and K. Tang, "Pattern recommendation in task-oriented applications: A multi-objective perspective [application notes]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 3, pp. 43–53, 2017.

[22] H. Yao, H. J. Hamilton, and C. J. Butz, "A foundational approach to mining itemset utilities from databases," in *Proceedings of the 2004 SIAM International Conference on Data Mining*, 2004, pp. 482–486.

[23] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceedings of the 20th International Conference on Very Large Data Bases*, 1994, pp. 487–499.

[24] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," *Data Mining and Knowledge Discovery*, vol. 8, no. 1, pp. 53–87, 2004.

[25] R. Agrawal and R. Srikant, "MAFIA: A maximal frequent itemset algorithm for transactional databases," in *Proceedings of the 17th International Conference on Data Engineering*, 2001, pp. 443–452.

[26] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.

[27] H. Tang, V. A. Shim, K. C. Tan, and J. Y. Chia, "Restricted Boltzmann machine based algorithm for multi-objective optimization," in *Proceedings of the 2010 IEEE Congress on Evolutionary Computation*, 2010.

[28] Y. T. Wu, Y. J. An, J. Geller, and Y. T. Wu, "A data mining based genetic algorithm," in *Proceedings of the Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, and the Second International Workshop on Collaborative Computing, Integration, and Assurance*, 2006.

[29] S. Lessmann, M. Caserta, and I. M. Arango, "Tuning metaheuristics: A data mining based approach for particle swarm optimization," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12 826–12 838, 2011.

[30] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proceedings of the Fifth Conference on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, 2001, pp. 95–100.

[31] H. Wang, L. Jiao, and X. Yao, "Two_Arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 524–541, 2015.

[32] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "An efficient approach to non-dominated sorting for evolutionary multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 201–213, 2015.

[33] M. Li, S. Yang, and X. Liu, "Pareto or non-Pareto: Bi-criterion evolution in multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 645–665, 2015.

[34] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[35] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[36] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 4, pp. 115–148, 1995.

[37] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Computer Science and Informatics*, vol. 26, no. 4, pp. 30–45, 1996.

[38] N. Verbiest, J. Derrac, C. Cornelis, S. García, and F. Herrera, "Evolutionary wrapper approaches for training set selection as preprocessing mechanism for support vector machines: Experimental evaluation and support vector analysis," *Applied Soft Computing*, vol. 38, pp. 10–22, 2016.

[39] M. Lalou, M. A. Tahraoui, and H. Kheddouci, "The critical node detection problem in networks: A survey," *Computer Science Review*, vol. 28, pp. 92–117, 2018.

[40] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[41] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[42] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, p. 440, 1998.

[43] Y. Tian, X. Xiang, X. Zhang, R. Cheng, and Y. Jin, "Sampling reference points on the Pareto fronts of benchmark multi-objective optimization problems," in *Proceedings of the 2018 IEEE Congress on Evolutionary Computation*, 2018, in press.

[44] J. Derrac, S. Garcia, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[45] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 29–38, 2006.