This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

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

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

***Abstract*—In real-world applications, there exist a lot of multiobjective optimization problems whose Pareto-optimal solutions are sparse, that is, most variables of these solutions are 0. Generally, many sparse multiobjective 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 article proposes an evolutionary algorithm for solving large-scale SMOPs, which aims to mine the sparse distribution of the 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 the Pareto-optimal solutions, and uses 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 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*—Evolutionary algorithm, genetic operator, pattern mining, sparse multiobjective optimization.**

Manuscript received March 15, 2020; revised July 10, 2020 and September 25, 2020; accepted November 25, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0100100; in part by the National Natural Science Foundation of China under Grant 61590922, Grant 61672033, Grant 61822301, Grant 61876123, Grant 61906001, Grant 62076001, 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 the Royal Society International Exchanges Program under Grant IEC\NSFC\170279. This article was recommended by Associate Editor H. Ishibuchi. *(Corresponding author: Xingyi Zhang.)*

Ye 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 (e-mail: field910921@gmail.com).

Chang Lu, Xingyi Zhang, and Fan 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 (e-mail: lucy_lc@outlook.com; xyzhanghust@gmail.com; chengfan@mail.ustc.edu.cn).

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

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2020.3041325.

Digital Object Identifier 10.1109/TCYB.2020.3041325

I. INTRODUCTION

MANY optimization problems in scientific and engineering areas are characterized by multiple objectives and sparse optimal solutions, which are known as sparse multiobjective optimization problems (SMOPs) [1]. The objectives of SMOPs conflict with each other to some extent; hence, there does not exist a single solution, making all the objectives optimal; instead, a set of tradeoff solutions, called the 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, that is, most variables of these solutions are 0. In neural-network training, sparse network structures are expected to alleviate overfitting [4], which means that most weights to be optimized should be 0; in portfolio optimization, only a small number of instruments can be selected to construct the portfolio [5]. SMOPs also widely exist in the field of cybernetics, such as power grid fault diagnosis [6] and ratio error estimation of voltage transformers [7]. Since most real-world SMOPs are pursued based on a large dataset, they are also large-scale multiobjective optimization problems (LMOPs).

In the last decade, a number of multiobjective evolutionary algorithms (MOEAs) have been tailored for solving LMOPs, which can be roughly divided into three categories. The first category divides the large number of decision variables by using control variable analysis [8], [9] or variable clustering [10], then optimizes them by the divide-and-conquer strategy. The second category reduces the decision space by transforming the LMOP into a small-scale problem [11]–[13] or using dimensionality reduction techniques [14], [15]. The third category develops delicate reproduction operators [16]–[20] or probability models [21], [22] for solving LMOPs more efficiently. Some of these algorithms (e.g., LSMOF [12]) dramatically improve the convergence speed of evolutionary algorithms in solving LMOPs. In contrast, their performance considerably deteriorates on large-scale SMOPs. On the one hand, the decision variable division-based MOEAs consume a large number of function evaluations for determining the interactions between decision variables [10], making them impractical for solving SMOPs with relatively expensive objective evaluation. On the other hand, for the problem transformation-based MOEAs and delicate reproduction operator-based MOEAs, they may get trapped in local optima [12] and cannot solve SMOPs with binary variables.

To address this issue, this article aims to reduce the difficulties of large-scale SMOPs by factoring their sparse nature into the optimization process. Following this idea, a novel pattern mining approach to assist MOEAs in solving large-scale SMOPs is proposed. The core idea of the proposed approach is to estimate the sparse distribution of the 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 article 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 the 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 0. Therefore, the decision space searched by genetic operators is highly 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 the 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) An MOEA is developed based on the proposed pattern mining approach and genetic operators, 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 remainder of this article is organized as follows. In Section II, some basic concepts related to sparse multiobjective 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 Multiobjective Optimization

An unconstrained multiobjective optimization problem can be mathematically defined as

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

where $\mathbf{x} = (x_1, \ldots, x_D) \in \Omega$ is a solution consisting of D decision variables, $\Omega \subseteq \mathbb{R}^D$ is the decision space, $\mathbf{f} : \Omega \to \Lambda \subseteq \mathbb{R}^M$ 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 $f_i(\mathbf{x}) \le f_i(\mathbf{y})$ for every $i \in \{1, \ldots, M\}$ and $f_j(\mathbf{x}) < f_j(\mathbf{y})$ for at least one $j \in \{1, \ldots, M\}$. 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 0; 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 need much more function evaluations for solving a problem with more decision variables, known as the curse of dimensionality [17]. 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 promise in many scenarios [23], 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 variables in the population will become 0, and it is desirable to detect the remaining nonzero variables by performing pattern mining on the current population. By doing

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

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

so, many zero variables can be ignored and the efficiency of finding the correct nonzero variables is significantly improved.

In the next section, 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 [24], much work has been dedicated to suggesting new objectives to evaluate the qualities of patterns, including *support* [25], *occupancy* [23], *area* [26], and *utility* [27], just to name a few. The *support* and *occupancy* are the two most widely used objectives, which can be calculated by

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

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 $\mathbf{x}$ that occur in

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

The first objective $f_{\text{supp}}(\mathbf{x})$ is the *support* of $\mathbf{x}$ and the second objective $f_{\text{occu}}(\mathbf{x})$ 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 $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_6$ consisting of seven items $A, B, \ldots, G$, and the solution $\mathbf{x}$ consists of three items A, B, and C. In short, *support* indicates the frequency $\mathbf{x}$ that 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 an 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 [26]. However, *support* and *occupancy* intuitively conflict with each other since a pattern containing few items can appear in many shopping lists but occupies 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 [28], the FP-Growth algorithm [29], the maximal frequent itemset algorithm [30], and the dominant and frequent itemset mining algorithm [23]. In recent years, evolutionary multiobjective optimization techniques have also been employed to solve the frequent pattern mining problem [26], [31]. 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 the Pareto-optimal solutions. Although some other dimensionality reduction techniques in machine learning have been adopted in MOEAs, such as principal component analysis [32] and restricted Boltzmann machine [15], evolutionary pattern mining has the following three advantages.

1) 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 unknown, a direct approximation of the nonzero variables based on the current nondominated 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.
2) 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.
3) 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. In 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 determine in advance since the Pareto-optimal solutions are unknown.

As a consequence, the proposed evolutionary pattern mining approach is parameterless and quite flexible. To the best of our knowledge, only a few single-objective evolutionary algorithms have adopted the traditional pattern mining approach (which is not based on the evolutionary algorithm) to establish operators [33] or tune the parameters of operators [34], 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. 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

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 nondominated front number of solutions in P;
  $P'' \leftarrow Variation(P', Q_{max}, Q_{min})$;
  $P \leftarrow EnvironmentalSelection(P \cup P'', N)$;
return P;

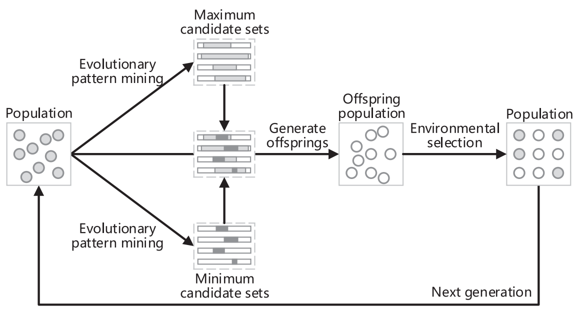

Fig. 1. Diagram of one generation of the proposed PM-MOEA.

Algorithm 2: *EnvironmentalSelection*(P, N)

Input: P (combined population), N (population size)
Output: P (population for next generation)

$[F_1, F_2, \cdots] \leftarrow$ Do nondominated 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;

are set to empty. In each generation of PM-MOEA, the maximum and minimum candidate sets are mined according to the nondominated solutions in the current population. Afterward, 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. Finally, 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 [35] and Two_Arch2 [36]. Specifically, the combined population is first sorted by nondominated sorting [37], and the solutions in the first k nondominated 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 the objective space is removed one by one, until $|F_1 \cup F_2 \cup \cdots \cup F_k| = N$. Thus, the population for the next generation consists of all the

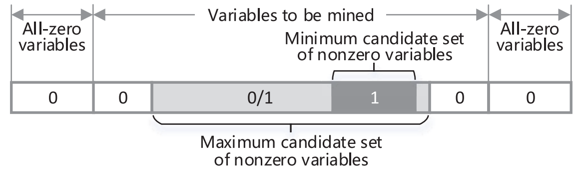

Fig. 2. Illustration of different parts of a decision vector. The variables in the maximum candidate set mean that they are possibly nonzero variables, which should be searched by genetic operators. The variables in the minimum candidate set mean that they should be nonzero variables, which are fixed 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.

solutions in $F_1 \cup F_2 \cup \cdots \cup F_k$, which contains N solutions in total.

Note that some other effective selection strategies [38], [39] can also be adopted in PM-MOEA, since the environmental selection is independent of the core components of PM-MOEA, that is, evolutionary pattern mining and genetic operators. In the next two sections, the two components of PM-MOEA are described in detail.

B. 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

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

where dec_i indicates the real value of the ith decision variable and mask_i indicates whether the ith decision variable is 0. Based on this encoding scheme, the entire decision vector is divided into several parts as illustrated in Fig. 2. The all-zero variables indicate that these variables are 0 in all the nondominated 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 nondominated 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* nondominated solutions should be considered in the maximum candidate set. In contrast, since the variables in a minimum candidate set should be nonzero, the variables that are nonzero in *most* nondominated solutions can be considered in the minimum candidate set. For this aim, we define two objectives for mining the maximum candidate sets by the

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

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

proposed evolutionary approach

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

where $\mathbf{x}$ is a binary vector denoting a maximum candidate set, P denotes the set of nondominated 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 $\mathbf{x}$

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

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

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

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 $\mathbf{y}$

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

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 the existing ones. The objectives for mining maximum candidate sets aim to find a set of nonzero variables covering more solutions, that is, a maximum candidate set can approximately be a *union set* of the nonzero variables in all solutions. Since all the variables will be selected if we only consider f_1, both f_1 and f_2 should be considered to find compact union sets. In contrast, the objectives for mining minimum candidate sets aim to find a set of nonzero variables existing in more solutions, that is, a minimum candidate set can approximately be an *intersection set* of the nonzero variables in all solutions. Since a single variable will be selected if we only consider f_3, both f_3 and f_4 should be considered to find compact intersection sets. Besides, mining both the maximum and minimum sets of the nonzero variable set is much less greedy than directly mining the nonzero variable set, where the latter is likely to make the population trapped in local optima. The results in Fig. 7 can demonstrate the superiority of mining both the maximum and minimum sets over mining just a single one.

Algorithm 3: *MiningMaxSets*(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)
1 $P \leftarrow$ Delete dominated solutions from P;
2 $Zero \leftarrow$ Set of variables that are zero in all the solutions in P;
3 $P \leftarrow$ Delete the dimensions that are in $Zero$ from the solutions in P;
4 **if** $Q_{max} \neq \emptyset$ **then**
5 $\quad Q_{max} \leftarrow$ Delete the dimensions that are in $Zero$ from the solutions in Q_{max};
6 **for** $i = 1$ *to* N' **do**
7 $\quad P_{sub} \leftarrow$ Randomly select half the solutions from P;
8 $\quad \mathbf{q} \leftarrow$ Set of variables that are nonzero in at least one solution in P_{sub};
9 $\quad Q_{max} \leftarrow Q_{max} \cup \{\mathbf{q}\}$;
10 Evaluate Q_{max} by (5) based on P;
11 **for** *generation* $= 1$ *to 10* **do**
12 $\quad Q'_{max} \leftarrow$ Select $2N'$ parents via binary tournament selection based on the nondominated front number of solutions in Q_{max};
13 $\quad Q''_{max} \leftarrow$ *BinaryOperators*(Q'_{max});
14 $\quad$ Evaluate Q''_{max} by (5) based on P;
15 $\quad Q_{max} \leftarrow$ *EnvironmentalSelection*$(Q_{max} \cup Q''_{max}, N')$;
16 **return** Q_{max};

Algorithm 4: *MiningMinSets*(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)
1 $P \leftarrow$ Delete dominated solutions from P;
2 $Zero \leftarrow$ Set of variables that are zero in all the solutions in P;
3 $P \leftarrow$ Delete the dimensions that are in $Zero$ from the solutions in P;
4 **if** $Q_{min} \neq \emptyset$ **then**
5 $\quad Q_{min} \leftarrow$ Delete the dimensions that are in $Zero$ from the solutions in Q_{min};
6 **for** $i = 1$ *to* N' **do**
7 $\quad P_{sub} \leftarrow$ Randomly select half the solutions from P;
8 $\quad \mathbf{q} \leftarrow$ Set of variables that are nonzero in all solutions in P_{sub};
9 $\quad Q_{min} \leftarrow Q_{min} \cup \{\mathbf{q}\}$;
10 Evaluate Q_{min} by (7) based on P;
11 **for** *generation* $= 1$ *to 10* **do**
12 $\quad Q'_{min} \leftarrow$ Select $2N'$ parents via binary tournament selection based on the nondominated front number of solutions in Q_{min};
13 $\quad Q''_{min} \leftarrow$ *BinaryOperators*(Q'_{min});
14 $\quad$ Evaluate Q''_{min} by (7) based on P;
15 $\quad Q_{min} \leftarrow$ *EnvironmentalSelection*$(Q_{min} \cup Q''_{min}, N')$;
16 **return** Q_{min};

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 nondominated solutions (lines 6–9). In each generation of the evolutionary approach, the maximum candidate sets evolve by the same mating selection, genetic operators, and environmental selection as Algorithm 1. As

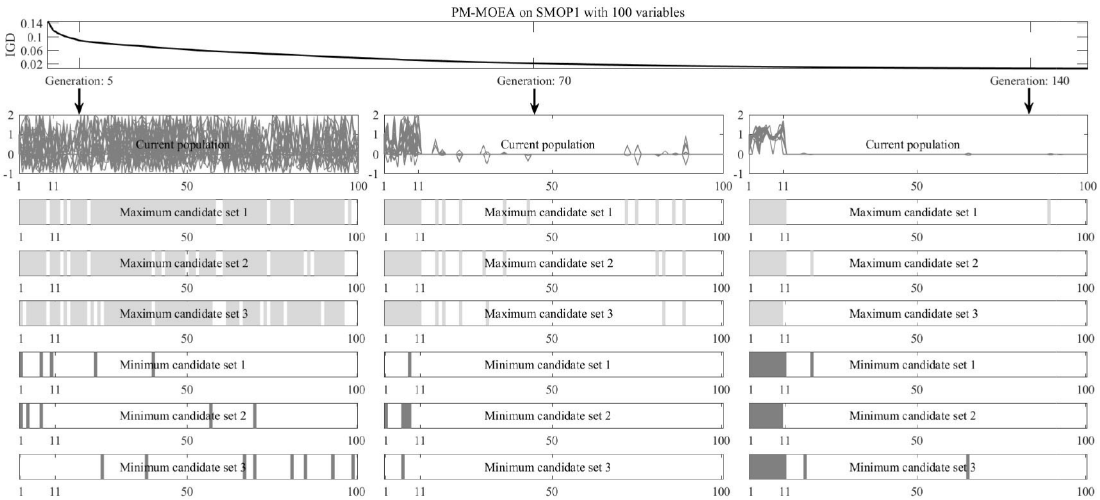

Fig. 3. Current population, three maximum candidate sets, and three minimum candidate sets at generation 5 (left column), 70 (middle column), and 140 (right column) in one run of PM-MOEA on SMOP1 with 100 variables. The 1st to 11th variables in Pareto-optimal solutions are nonzero, and all the other variables in Pareto-optimal solutions are 0.

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 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 [40] value, the population approximates the Pareto-optimal solutions with more and more variables becoming 0. 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 the one hand, the maximum and minimum candidate sets from the 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 transaction dataset is the mask of the nondominated 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 nondominated 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 Table VI.

C. 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 dec of each offspring solution is generated by performing simulated binary crossover [41] and polynomial mutation [42] on 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 first randomly selects a maximum and a minimum candidate sets, where the minimum candidate set should be a subset of the maximum candidate set (lines 7–11); while if all the minimum candidate sets are not the subset of any maximum candidate set, an empty minimum candidate set will be used (lines 3–5). Then, two parents are selected to generate an offspring solution each time (line 12). For generating the mask of the offspring solution, the proposed binary genetic operators are performed on the dimensions of the selected maximum candidate set (line 15), and the variables in the dimensions of the selected minimum candidate set are set to 1 (line 16). For generating the dec of the offspring solution, the simulated binary crossover and polynomial mutation are directly performed on all the dimensions (line 17).

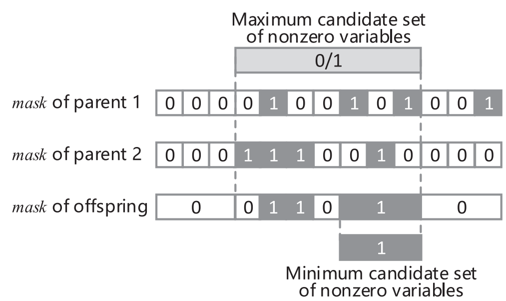

Fig. 4. Illustration of generating the mask of an offspring solution based on a maximum candidate set and a minimum candidate set.

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. Assume a solution to be mutated contains D variables, where n variables are nonzero and $(D-n)$ variables are 0. After the

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)

$P'' \leftarrow \emptyset$;
while $P' \neq \emptyset$ **do**
  //Randomly select a maximum and a minimum sets
  if *no set in* Q_{min} *is the subset of any set in* Q_{max} **then**
    $\mathbf{q}_{max} \leftarrow$ Randomly select a set from Q_{max};
    $\mathbf{q}_{min} \leftarrow \emptyset$;
  else
    $Q'_{min} = \emptyset$;
    while $Q'_{min} = \emptyset$ **do**
      $\mathbf{q}_{max} \leftarrow$ Randomly select a set from Q_{max};
      $Q'_{min} \leftarrow \{\mathbf{q} \in Q_{min} | \mathbf{q} \subseteq \mathbf{q}_{max}\}$;
    $\mathbf{q}_{min} \leftarrow$ Randomly select a set from Q'_{min};
  //Select two parents to generate one offspring
  $[\mathbf{p}_1, \mathbf{p}_2] \leftarrow$ Randomly select two parents from P';
  $P' \leftarrow P' \setminus \{\mathbf{p}_1, \mathbf{p}_2\}$;
  //Generate the mask of the offspring on the dimensions of the maximum candidate set
  $\mathbf{o}.mask \leftarrow$ A vector of zeros;
  $\mathbf{o}.mask(\mathbf{q}_{max}) \leftarrow BinaryOperators(\{\mathbf{p}_1.mask(\mathbf{q}_{max}), \mathbf{p}_2.mask(\mathbf{q}_{max})\})$; //$\mathbf{o}.mask(\mathbf{q}_{max})$ denotes the variables in dimensions $\mathbf{q}_{max}$ of $\mathbf{o}.mask$
  //Generate the mask of the offspring on the dimensions of the minimum candidate set
  $\mathbf{o}.mask(\mathbf{q}_{min}) \leftarrow 1$; //$\mathbf{o}.mask(\mathbf{q}_{min})$ denotes the variables in dimensions $\mathbf{q}_{min}$ of $\mathbf{o}.mask$
  //Generate the dec of the offspring solution
  $\mathbf{o}.dec \leftarrow$ Perform simulated binary crossover and polynomial mutation based on $\mathbf{p}_1.dec$ and $\mathbf{p}_2.dec$;
  $P'' \leftarrow P'' \cup \{\mathbf{o}\}$;
return P'';

mutation is performed with a probability of p, the expectation of the number of nonzero variables will be changed to

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

Analogously, if the mutation is performed for i times, the expectation of the number of nonzero variables $E^i_{\text{nonzero}} = (1-2p)E^{i-1}_{\text{nonzero}} + pD$, hence

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

When $i \to \infty$, the expectation of the number of nonzero variables will be $(D/2)$. 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.

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   $diff \leftarrow \text{xor}(\mathbf{p}_1, \mathbf{p}_2)$; //The variables that are different in $\mathbf{p}_1$ and $\mathbf{p}_2$
6   $prob_{diff} \leftarrow$ Calculate the prob. to flip each variable in dimensions $diff$ of $\mathbf{o}$ by (11);
7   Flip each variable in dimensions $diff$ of $\mathbf{o}$ with the prob. determined by $prob_{diff}$;
  //The proposed binary mutation
8   $prob \leftarrow$ Calculate the prob. to flip each variable of $\mathbf{o}$ by (11);
9   Flip each variable of $\mathbf{o}$ with the pro. determined by $prob$;
10   $P'' \leftarrow P'' \cup \{\mathbf{o}\}$;
11 **return** P'';

Since $p_1 n + p_0(D-n) = pD$ should be met, 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 $(2n/D)$ if it is larger than this value.

Algorithm 6 presents the procedure of the proposed binary operators. Each offspring solution is first set to the same as the first parent (line 4). Then, the crossover operator flips each variable in the dimensions where the two parents are different with the probability calculated by (11) (line 7). Finally, 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: 1) neural-network training [44]; 2) feature selection [45]; 3) critical node detection [46]; and 4) portfolio optimization [3]. We select LMEA [10], WOF-SMPSO [11], LMOCSO [17], and SparseEA [1] as the baseline algorithms, where LMEA, WOF-SMPSO, and LMOCSO are state-of-the-art MOEAs tailored for LMOPs and based on decision variable division, problem transformation, and delicate reproduction operator, respectively. Besides, SparseEA is an efficient 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 [47].

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

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 (ratio of nonzero variables in optimal solutions)		
SMOP1–SMOP8	Real	100 500 1000 5000 10000	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
Feature selection problem	**No. of variables**	**No. of variables (D)**	**Dataset**	**No. of samples**	**No. of features**	**No. of classes**
FS1	Binary	166	MUSK1[1]	476	166	2
FS2		500	Madelon[1]	2600	500	2
FS3		800	Gse72526[3]	61	800	4
FS4		5966	Prostate GE[4]	102	5966	2
Critical node detection problem	**Type of variables**	**No. of variables (D)**	**Dataset**	**No. of nodes**	**No. of edges**	
CN1	Binary	235	ER235[5]	235	350	
CN2		466	ER466[5]	466	700	
CN3		941	ER941[5]	941	1400	
CN4		4941	US power grid [43]	4941	6594	
Portfolio optimization problem	**Type of variables**	**No. of variables (D)**	**Dataset**	**No. of instruments**	**Length of each instrument**	
PO1	Real	100	EURCHF[6]	100	50	
PO2		500	EURCHF[6]	500	50	
PO3		1000	EURCHF[6]	1000	50	
PO4		5000	EURCHF[6]	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://www.ncbi.nlm.nih.gov/geo/query/acc.cgi 4. http://featureselection.asu.edu/datasets.php
5. http://individual.utoronto.ca/mventresca/cnd.html 6. https://www.metatrader5.com/en

with sparse Pareto-optimal solutions and various difficulties, including multimodality, 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 supplementary materials I.

B. Settings of Algorithms

1) Operators: For LMEA, SparseEA, and PM-MOEA, they use simulated binary crossover [41] and polynomial mutation [42] 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. 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 solve SMOPs with binary variables, LMEA, WOF-SMPSO, and LMOCSO optimize real variables within [0, 1] and round them 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 SMOPs with real variables and $100 \times D$ for SMOPs with binary variables.

4) Other Parameters: In LMEA, the number of selected solutions for variable clustering is set to 2, the number of perturbations on each solution for variable clustering is set to 4, and the number of selected solutions for variable interaction analysis is set to 5. In WOF-SMPSO, the number of groups, the number of evaluations for the original problem, the number of evaluations for the 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, and 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 [40] obtained by LMEA, WOF-SMPSO, LMOCSO, SparseEA, and the proposed PM-MOEA on SMOP1–SMOP8 with 100 to 10 000 decision variables, averaged over 30 runs. Each IGD value is calculated based on one population with respect to approximately 10 000 reference points uniformly sampled on the true Pareto front [48]. In addition, the Wilcoxon rank-sum test [49] with a significance level of 0.05 is adopted to 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 35 of 40 test instances. In terms of the Wilcoxon rank-sum test, PM-MOEA is significantly better than LMEA, WOF-SMPSO, LMOCSO, and SparseEA on 40, 38, 40, and 36 test instances, respectively. Besides, the HV values obtained by the compared MOEAs on the same test instances are presented in supplementary materials II, where the statistical results are the same

TABLE IV
IGD VALUES OBTAINED BY LMEA, WOF-SMPSO, LMOCSO, SPARSEEA, AND THE PROPOSED PM-MOEA ON SMOP1–SMOP8 WITH 100 TO 10 000 DECISION VARIABLES. THE BEST RESULT IN EACH ROW IS HIGHLIGHTED

Problem	D	LMEA	WOF-SMPSO	LMOCSO	SparseEA	PM-MOEA
SMOP1	100	5.7082e-1 (1.67e-2) −	2.7201e-1 (3.22e-2) −	3.8628e-1 (3.87e-2) −	7.8464e-3 (1.50e-3) −	5.3914e-3 (1.50e-3)
	500	6.6271e-1 (9.37e-3) −	2.5109e-1 (2.12e-2) −	4.8602e-1 (2.22e-2) −	1.5518e-2 (2.63e-3) −	1.0332e-2 (2.60e-3)
	1000	6.8976e-1 (5.27e-3) −	2.2479e-1 (2.62e-2) −	4.9768e-1 (2.37e-2) −	2.4364e-2 (2.20e-3) −	1.5347e-2 (1.73e-3)
	5000	7.1809e-1 (1.72e-3) −	1.8214e-1 (1.67e-2) −	5.3715e-1 (1.83e-2) −	3.6494e-2 (1.36e-3) −	3.0367e-2 (1.00e-3)
	10000	7.2322e-1 (1.45e-3) −	1.5555e-1 (1.78e-2) −	5.4600e-1 (1.55e-2) −	3.6999e-2 (1.05e-3) −	3.4337e-2 (1.43e-3)
SMOP2	100	1.5137e+0 (2.35e-2) −	8.7620e-1 (1.54e-1) −	1.5205e+0 (6.24e-2) −	2.6258e-2 (7.53e-3) −	1.5835e-2 (4.37e-3)
	500	1.5952e+0 (1.02e-2) −	3.5279e-1 (1.55e-1) −	1.6803e+0 (6.51e-2) −	4.2796e-2 (4.91e-3) −	3.1189e-2 (5.89e-3)
	1000	1.6161e+0 (5.99e-3) −	2.3584e-1 (9.36e-2) −	1.6723e+0 (6.62e-2) −	6.2346e-2 (6.59e-3) −	4.2868e-2 (5.62e-3)
	5000	1.6404e+0 (2.37e-3) −	1.7635e-1 (1.16e-2) −	1.6987e+0 (4.74e-2) −	9.4302e-2 (2.10e-3) −	7.7687e-2 (2.72e-3)
	10000	1.6449e+0 (1.75e-3) −	1.7194e-1 (9.46e-3) −	1.7124e+0 (5.14e-2) −	9.8680e-2 (1.13e-3) −	8.8310e-2 (6.15e-4)
SMOP3	100	1.8867e+0 (1.79e-2) −	7.1767e-1 (1.27e-2) −	1.5550e+0 (7.73e-2) −	1.3303e-2 (2.49e-3) ≈	2.9654e-2 (4.13e-2)
	500	1.9797e+0 (1.84e-2) −	7.0295e-1 (2.90e-3) −	1.6235e+0 (2.56e-2) −	1.7596e-2 (3.52e-3) −	1.3618e-2 (2.56e-3)
	1000	2.0012e+0 (8.84e-3) −	7.0188e-1 (1.38e-3) −	1.6491e+0 (2.39e-2) −	2.5076e-2 (3.15e-3) +	5.5470e-2 (1.72e-1)
	5000	2.0308e+0 (3.47e-3) −	7.0172e-1 (1.52e-3) −	1.6928e+0 (1.25e-2) −	3.9696e-2 (1.67e-3) +	6.7110e-2 (1.70e-1)
	10000	2.0357e+0 (8.29e-4) −	7.0181e-1 (1.20e-3) −	1.7034e+0 (7.76e-3) −	4.2396e-2 (3.03e-4) −	3.3909e-2 (7.13e-4)
SMOP4	100	7.6353e-1 (1.78e-2) −	3.9212e-1 (8.86e-2) −	7.6069e-1 (4.05e-2) −	4.6830e-3 (2.33e-4) −	4.0730e-3 (6.06e-5)
	500	8.0467e-1 (6.48e-3) −	7.4637e-2 (6.13e-2) −	8.2115e-1 (3.67e-2) −	4.7577e-3 (3.03e-4) −	4.0736e-3 (7.87e-5)
	1000	8.1201e-1 (3.75e-3) −	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	8.2314e-1 (1.99e-3) −	9.4936e-3 (8.44e-3) −	8.6032e-1 (1.49e-2) −	4.7641e-3 (2.13e-4) −	4.0731e-3 (6.52e-5)
	10000	8.2537e-1 (1.01e-3) −	6.0595e-3 (9.23e-4) −	9.3837e-1 (7.29e-2) −	4.9447e-3 (2.71e-4) −	4.0475e-3 (5.25e-5)
SMOP5	100	4.8157e-1 (1.19e-2) −	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	5.2588e-1 (6.02e-3) −	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	5.3068e-1 (4.48e-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	5.4436e-1 (1.94e-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)
	10000	5.4712e-1 (1.14e-3) −	3.4846e-1 (6.69e-5) −	4.3278e-1 (1.27e-3) −	5.3232e-3 (9.12e-5) −	4.6567e-3 (6.46e-5)
SMOP6	100	1.7761e-1 (4.58e-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	1.9956e-1 (2.28e-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	2.0361e-1 (1.66e-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	2.1010e-1 (9.16e-4) −	3.2781e-2 (1.52e-2) −	1.8294e-1 (3.47e-3) −	6.7850e-3 (2.47e-4) −	4.7824e-3 (7.31e-5)
	10000	2.1084e-1 (5.14e-4) −	2.4512e-2 (6.53e-3) −	1.8993e-1 (4.96e-3) −	6.8933e-3 (1.49e-4) −	5.1611e-3 (3.25e-5)
SMOP7	100	1.0404e+0 (3.90e-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	1.2019e+0 (2.29e-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	1.2474e+0 (8.37e-3) −	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	1.3001e+0 (5.66e-3) −	7.9518e-2 (5.62e-3) +	5.9117e-1 (5.26e-2) −	1.1773e-1 (4.85e-3) −	1.0088e-1 (5.38e-3)
	10000	1.3118e+0 (5.11e-3) −	7.4387e-2 (2.69e-3) +	5.8160e-1 (3.52e-2) −	1.2456e-1 (1.59e-3) −	1.1515e-1 (4.20e-3)
SMOP8	100	2.8651e+0 (3.54e-2) −	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	3.0511e+0 (9.63e-3) −	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	3.0936e+0 (7.69e-3) −	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	3.1392e+0 (4.61e-3) −	5.3296e-1 (2.80e-3) −	2.2979e+0 (5.45e-2) −	3.0769e-1 (8.27e-3) −	2.6568e-1 (1.19e-2)
	10000	3.1494e+0 (1.80e-3) −	5.3069e-1 (1.00e-3) −	2.3078e+0 (3.74e-2) −	3.2570e-1 (5.69e-3) −	2.9453e-1 (7.97e-3)
$+/-/\approx$		0/40/0	2/38/0	0/40/0	2/36/2	

to those in Table IV. To illustrate the superiority of PM-MOEA visually, Fig. 5 shows the parallel coordinates plot [50] 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 LMEA and LMOCSO, it is obvious that most variables of the obtained solutions are far from 0. For WOF-SMPSO, most variables of the obtained solutions are close to 0. As for SparseEA and the proposed PM-MOEA, most variables of the obtained solutions are equal to 0, 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. Furthermore, Fig. 6 plots the average IGD values obtained by the compared MOEAs on SMOP5 with 500 variables and different levels of sparsity. It can be observed that SparseEA and PM-MOEA are more effective when the problem becomes sparser, since these two MOEAs are tailored for SMOPs. In contrast, the other three MOEAs tailored for LMOPs are effective for nonsparse problems, due to the fact that the nonzero variables are easier to be optimized than the zero variables in SMOP5.

D. *Effectiveness of the Components in PM-MOEA*

The superiority of the proposed PM-MOEA is mainly attributed to its two components, that is: 1) the evolutionary pattern mining approach and 2) 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. 7 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 and the PM-MOEA″ without maximum

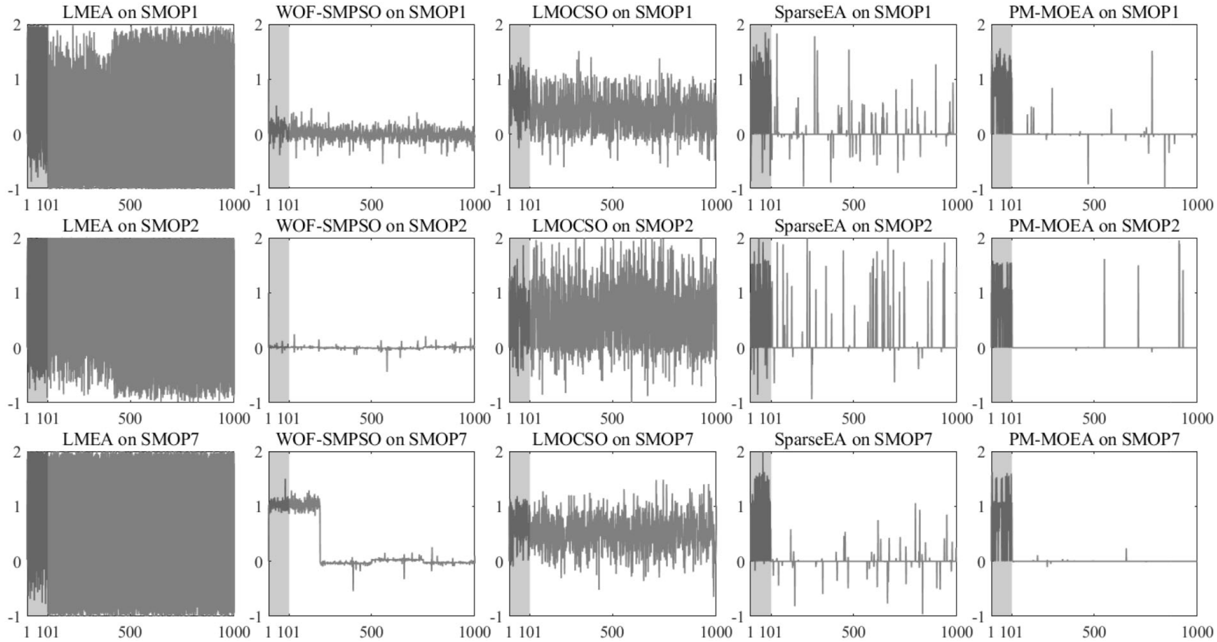

Fig. 5. Parallel coordinates plot of the decision variables of solutions obtained by LMEA, WOF-SMPSO, LMOCSO, SparseEA, and the proposed PM-MOEA on SMOP1, SMOP2, and SMOP7 with 1000 variables. All the variables outside the gray region are zero in the Pareto-optimal solutions.

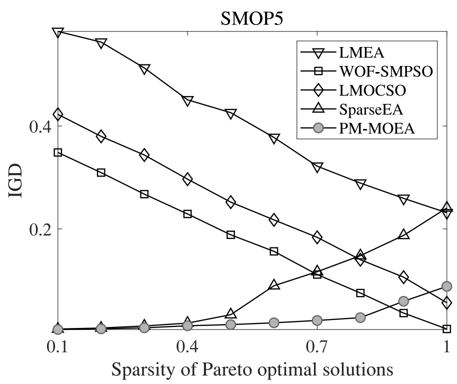

Fig. 6. IGD values obtained by LMEA, WOF-SMPSO, LMOCSO, SparseEA, and the proposed PM-MOEA on SMOP5 with 500 variables, where the sparsity of the Pareto-optimal solutions is ranged from 0.1 to 1.

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 section, 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 [51] obtained by LMEA, WOF-SMPSO, LMOCSO, SparseEA, and PM-MOEA on the neural network training problem (NN1–NN4), the feature selection problem (FS1–FS4), 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, SparseEA performs the best on two test instances, while LMEA, WOF-SMPSO, and LMOCSO cannot gain any best result. Besides, the IGD values obtained by the compared MOEAs on the same test instances are presented in supplementary materials II, where the statistical results are similar to those in Table V. Furthermore, Fig. 8 plots the objective values of solutions obtained by the compared MOEAs on NN4, FS4, CN4, and PO4. For the neural-network training problem (NN4), the solutions obtained by SparseEA and PM-MOEA are much sparser than those obtained by LMEA, WOF-SMPSO, and LMOCSO, and the solutions obtained by PM-MOEA dominate those obtained by SparseEA. For the feature selection problem (FS4), the solutions obtained by PM-MOEA and SparseEA are also better than those obtained by the others. For the critical node detection problem (CN4), the solutions obtained by PM-MOEA have better convergence and spread than those obtained by the other MOEAs. As for the portfolio optimization problem (PO4), the solutions obtained by PM-MOEA also dominate those obtained by the other. In short, SparseEA and the proposed PM-MOEA outperform the other MOEAs in solving real-world SMOPs, and the performance of PM-MOEA is similar to or better than SparseEA on most test instances. In addition, supplementary materials III presents the results of SparseEA and PM-MOEA with fewer function evaluations, where the superiority of PM-MOEA over SparseEA becomes larger.

F. Computational Efficiency of PM-MOEA

Finally, the computational efficiency of PM-MOEA is compared to the other four MOEAs. Table VI lists the runtimes

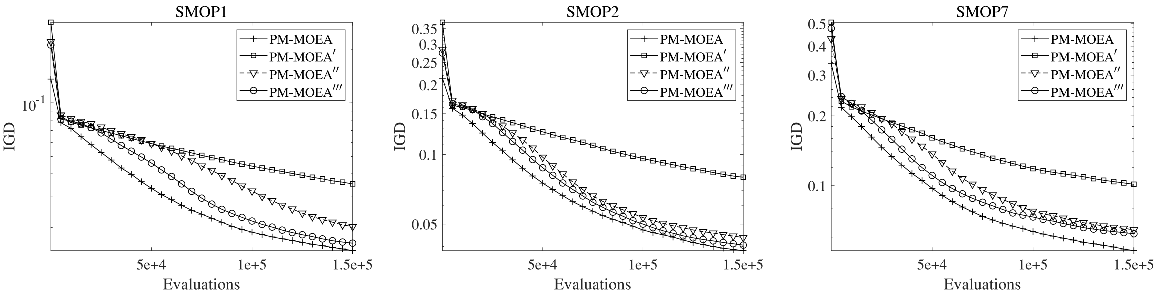

Fig. 7. Convergence profiles of IGD values obtained by PM-MOEA, PM-MOEA′ (without minimum candidate set), PM-MOEA″ (without maximum candidate set), and PM-MOEA‴ (without the proposed genetic operators) on SMOP1, SMOP2, and SMOP7 with 1000 variables.

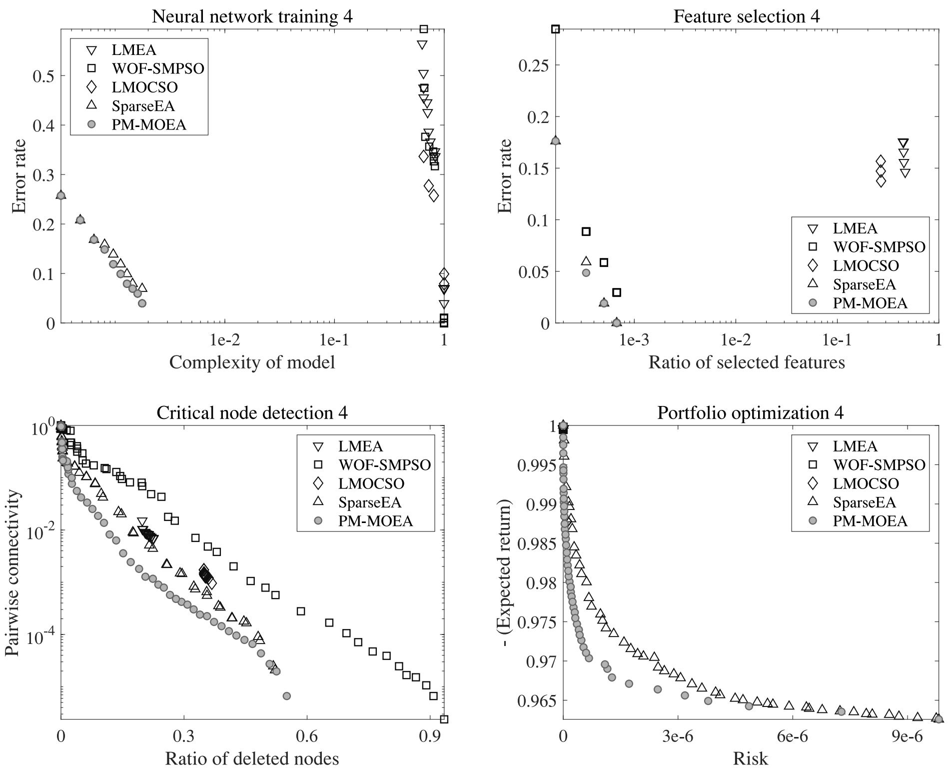

Fig. 8. Objective values of solutions obtained by LMEA, WOF-SMPSO, LMOCSO, SparseEA, and the proposed PM-MOEA on neural-network training, feature selection, critical node detection, and portfolio optimization.

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

Problem	D	LMEA	WOF-SMPSO	LMOCSO	SparseEA	PM-MOEA
NN1	101	3.5492e-1 (2.65e-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	2.8351e-1 (8.81e-3) −	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	3.1748e-1 (1.67e-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	3.2210e-1 (1.57e-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)
FS1	166	6.9370e-1 (1.24e-2) −	8.9339e-1 (3.88e-3) −	6.7275e-1 (2.37e-2) −	9.3149e-1 (2.65e-3) −	9.3424e-1 (4.50e-6)
FS2	500	4.6373e-1 (9.75e-3) −	7.6560e-1 (4.28e-2) −	4.7383e-1 (8.79e-3) −	9.1710e-1 (1.27e-6) −	9.1941e-1 (2.50e-6)
FS3	800	9.4504e-1 (1.63e-2) −	9.1044e-1 (1.50e-6) −	2.5649e-1 (1.37e-3) −	9.8477e-1 (7.95e-6) +	9.6961e-1 (4.00e-6)
FS4	5966	5.1955e-1 (1.49e-3) −	9.7305e-1 (1.39e-6) −	6.6321e-1 (1.00e-3) −	9.9990e-1 (7.26e-7) ≈	9.9990e-1 (5.05e-7)
CN1	235	6.9111e-1 (3.38e-3) −	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	7.2937e-1 (1.62e-2) −	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	7.3616e-1 (1.29e-2) −	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	8.1912e-1 (6.91e-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	9.6593e-2 (7.48e-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	9.2334e-2 (1.07e-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	9.1885e-2 (1.34e-4) −	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	9.1398e-2 (8.22e-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	0/16/0	0/16/0	1/7/8	

(in second) of the five MOEAs on the benchmark SMOPs and real-world SMOPs. 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 FS4 and PO4. Since PM-MOEA needs to perform the proposed evolutionary pattern mining approach many

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

Problem	D	LMEA	WOF-SMPSO	LMOCSO	SparseEA	PM-MOEA
SMOP1–SMOP8 (average)	5000	1.3737e+3	1.8940e+3	6.4435e+2	2.1515e+3	9.8683e+3
SMOP1–SMOP8 (average)	10000	6.1171e+3	9.1643e+3	2.3030e+3	1.3287e+4	4.7289e+4
NN4	6241	3.8762e+3	5.9592e+3	3.7517e+3	5.8890e+3	7.0076e+3
FS4	5966	1.9890e+4	8.7081e+3	1.6930e+4	1.2198e+4	9.9377e+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

times, its runtime is slightly longer than the other MOEAs. While for some real-world SMOPs, such as FS4 and PO4, PM-MOEA becomes more efficient since it can obtain very sparse solutions and a sparse solution corresponds to a cheap objective evaluation of these SMOPs. In short, the computational efficiency of PM-MOEA is not obviously worse than the existing MOEAs.

V. CONCLUSION

To address the curse of dimensionality in solving large-scale SMOPs, this article has proposed a pattern mining-based evolutionary algorithm to mine the sparse distribution of the 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 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, feature 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 the pattern mining approach in solving large-scale SMOPs, and further exploration on the potential of this approach is highly desirable. On the 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 unconstrained large-scale SMOPs, and it is interesting to combine the proposed MOEA with constraint handling techniques to solve large-scale SMOPs with constraints [7].

REFERENCES

[1] Y. Tian, X. Zhang, C. Wang, and Y. Jin, "An evolutionary algorithm for large-scale sparse multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 380–393, Apr. 2020.

[2] Y. Jin, T. Okabe, and B. Sendhoff, "Neural network regularization and ensembling using multi-objective evolutionary algorithms," in *Proc. IEEE Congr. Evol. Comput.*, Portland, OR, USA, 2004, pp. 1–8.

[3] A. Ponsich, A. L. Jaimes, and C. A. C. Coello, "A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 321–344, Jun. 2013.

[4] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, 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," *Eur. J. Oper. Res.*, vol. 199, no. 3, pp. 684–693, 2009.

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

[7] C. He, R. Cheng, C. Zhang, Y. Tian, H. Li, and X. Yao, "Evolutionary large-scale multiobjective optimization for ratio error estimation of voltage transformers," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 868–881, Oct. 2020.

[8] X. Ma *et al.*, "A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 275–298, Apr. 2016.

[9] B. Cao, J. Zhao, Z. Lv, and X. Liu, "A distributed parallel cooperative coevolutionary multiobjective evolutionary algorithm for large-scale optimization," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 2030–2038, Aug. 2017.

[10] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 97–112, Feb. 2018.

[11] H. Zille, H. Ishibuchi, S. Mostaghim, and Y. Nojima, "A framework for large-scale multiobjective optimization based on problem transformation," *IEEE Trans. Evol. Comput.*, vol. 22, no. 2, pp. 260–275, Apr. 2018.

[12] C. He *et al.*, "Accelerating large-scale multiobjective optimization via problem reformulation," *IEEE Trans. Evol. Comput.*, vol. 23, no. 6, pp. 949–961, Dec. 2019.

[13] H. Zille and S. Mostaghim, "Linear search mechanism for multi- and many-objective optimisation," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.*, 2019, pp. 399–410.

[14] R. Liu, R. Ren, J. Liu, and J. Liu, "A clustering and dimensionality reduction based evolutionary algorithm for large-scale multi-objective problems," *Appl. Soft Comput.*, vol. 89, Apr. 2020, Art. no. 106120.

[15] Y. Tian, C. Lu, X. Zhang, K. C. Tan, and Y. Jin, "Solving large-scale multiobjective optimization problems with sparse optimal solutions via unsupervised neural networks," *IEEE Trans. Cybern.*, early access, Mar. 26, 2020, doi: 10.1109/TCYB.2020.2979930.

[16] Y. Qi, L. Bao, X. Ma, Q. Miao, and X. Li, "Self-adaptive multi-objective evolutionary algorithm based on decomposition for large-scale problems: A case study on reservoir flood control operation," *Inf. Sci.*, vols. 367-368, pp. 529–549, Nov. 2016.

[17] Y. Tian, X. Zheng, X. Zhang, and Y. Jin, "Efficient large-scale multiobjective optimization based on a competitive swarm optimizer," *IEEE Trans. Cybern.*, vol. 50, no. 8, pp. 3696–3708, Aug. 2020.

[18] H. Hiba, A. A. Bidgoli, A. Ibrahim, and S. Rahnamayan, "CGDE3: An efficient center-based algorithm for solving large-scale multi-objective optimization problems," in *Proc. IEEE Congr. Evol. Comput.*, 2019, pp. 350–358.

[19] Y. Abdi and M. Feizi-Derakhshi, "Hybrid multi-objective evolutionary algorithm based on search manager framework for big data optimization problems," *Appl. Soft Comput.*, vol. 87, Feb. 2020, Art. no. 105991.

[20] C. He, R. Cheng, and D. Yazdani, "Adaptive offspring generation for evolutionary large-scale multiobjective optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Jul. 10, 2020, doi: 10.1109/TSMC.2020.3003926.

[21] W. Hong, K. Tang, A. Zhou, H. Ishibuchi, and X. Yao, "A scalable indicator-based evolutionary algorithm for large-scale multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 525–537, Jun. 2019.

[22] H. Chen, R. Cheng, J. Wen, H. Li, and J. Weng, "Solving large-scale many-objective optimization problems by covariance matrix adaptation evolution strategy with scalable small subpopulations," *Inf. Sci.*, vol. 509, pp. 457–469, Jan. 2020.

[23] L. Tang, L. Zhang, P. Luo, and M. Wang, "Incorporating occupancy into frequent pattern mining for high quality pattern recommendation," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manag.*, 2012, pp. 75–84.

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

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

[26] 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 Comput. Intell. Mag.*, vol. 12, no. 3, pp. 43–53, Aug. 2017.

[27] H. Yao, H. J. Hamilton, and C. J. Butz, "A foundational approach to mining itemset utilities from databases," in *Proc. SIAM Int. Conf. Data Mining*, 2004, pp. 482–486.

[28] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proc. 20th Int. Conf. Very Large Data Bases*, 1994, pp. 487–499.

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

[30] R. Agrawal and R. Srikant, "MAFIA: A maximal frequent itemset algorithm for transactional databases," in *Proc. 17th Int. Conf. Data Eng.*, Heidelberg, Germany, 2001, pp. 443–452.

[31] Y. Tian, S. Yang, L. Zhang, F. Duan, and X. Zhang, "A surrogate-assisted multiobjective evolutionary algorithm for large-scale task-oriented pattern mining," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3, no. 2, pp. 106–116, Apr. 2019.

[32] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 41–63, Feb. 2008.

[33] Y. T. Wu, Y. J. An, J. Geller, and Y. T. Wu, "A data mining based genetic algorithm," in *Proc. 4th IEEE Workshop Softw. Technol. Future Embedded Ubiquitous Syst. 2nd Int. Workshop Collab. Comput. Integr. Assurance*, 2006, p. 6.

[34] S. Lessmann, M. Caserta, and I. M. Arango, "Tuning metaheuristics: A data mining based approach for particle swarm optimization," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 12826–12838, 2011.

[35] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. 5th Conf. Evol. Methods Design Optim. Control Appl. Ind. Problems*, 2001, pp. 95–100.

[36] H. Wang, L. Jiao, and X. Yao, "Two_arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 524–541, Aug. 2015.

[37] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "An efficient approach to nondominated sorting for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 201–213, Apr. 2015.

[38] M. Li, S. Yang, and X. Liu, "Pareto or non-Pareto: Bi-criterion evolution in multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 645–665, Oct. 2016.

[39] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.

[40] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

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

[42] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Comput. Sci. Informat.*, vol. 26, no. 4, pp. 30–45, 1996.

[43] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[44] Y. Jin and B. Sendhoff, "Pareto-based multiobjective machine learning: An overview and case studies," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 3, pp. 397–415, May 2008.

[45] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1656–1671, Dec. 2013.

[46] M. Lalou, M. A. Tahraoui, and H. Kheddouci, "The critical node detection problem in networks: A survey," *Comput. Sci. Rev.*, vol. 28, pp. 92–117, May 2018.

[47] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 73–87, Nov. 2017.

[48] 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 *Proc. IEEE Congr. Evol. Comput.*, 2018, pp. 1–6.

[49] 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 Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011.

[50] M. Li, L. Zhen, and X. Yao, "How to read many-objective solution sets in parallel coordinates [educational forum]," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 88–100, Nov. 2017.

[51] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 29–38, Feb. 2006.

Ye Tian (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from Anhui University, Hefei, China, in 2012, 2015, and 2018, respectively.

He is currently an Associate Professor with the Institutes of Physical Science and Information Technology, Anhui University, and also a Postdoctoral Research Fellow with the Department of Computer Science, City University of Hong Kong, Hong Kong. His current research interests include multiobjective optimization methods and their applications.

Dr. Tian is a recipient of the 2018 and 2021 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award and the 2020 *IEEE Computational Intelligence Magazine* Outstanding Paper Award.

Chang Lu received the B.Sc. degree from Anhui University, Hefei, China, in 2017, where he is currently pursuing the M.Sc. degree with the School of Computer Science and Technology.

His current research interests include multiobjective optimization and machine learning.

Xingyi Zhang (Senior Member, IEEE) received the B.Sc. degree from Fuyang Normal College, Fuyang, China, in 2003, and the M.Sc. and Ph.D. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 2006 and 2009, respectively.

He is currently a Professor with the School of Computer Science and Technology, Anhui University, Hefei, China. His current research interests include unconventional models and algorithms of computation, multiobjective optimization, and membrane computing.

Prof. Zhang is a recipient of the 2018 and 2021 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award and the 2020 *IEEE Computational Intelligence Magazine* Outstanding Paper Award.

Fan Cheng received the B.Sc. and M.Sc. degrees from the Hefei University of Technology, Hefei, China, in 2000 and 2003, respectively, and the Ph.D. degree from the University of Science and Technology of China, Hefei, in 2012.

He is currently an Associate Professor with the School of Computer Science and Technology, Anhui University, Hefei. His main research interests include machine learning, multiobjective optimization, and complex networks.

Yaochu Jin (Fellow, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1988, 1991, and 1996, respectively, and the Dr.-Ing. degree from Ruhr University Bochum, Bochum, Germany, in 2001.

He is currently a Distinguished Chair Professor of Computational Intelligence with the Department of Computer Science, University of Surrey, Guildford, U.K., where he heads the Nature Inspired Computing and Engineering Group. He was a Finland Distinguished Professor and a Changjiang Distinguished Visiting Professor. He has (co)authored over 300 peer-reviewed journal and conference papers and been granted eight patents on evolutionary optimization.

Dr. Jin is a recipient of the 2014, 2016, and 2020 *IEEE Computational Intelligence Magazine* Outstanding Paper Award, the 2018 and 2021 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award, and the Best Paper Award of the 2010 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology. He has been named a Highly Cited Researcher for 2019 and 2020 by the Web of Science Group. He is the Co-Editor-in-Chief of the IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS and *Complex & Intelligent Systems*. He is also an Associate Editor or an Editorial Board Member of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON CYBERNETICS, *Evolutionary Computation*, and *Soft Computing*. He was an IEEE Distinguished Lecturer from 2013 to 2017.