A Surrogate-Assisted Multi-Objective Evolutionary Algorithm for Large-Scale Task-Oriented Pattern Mining

Ye Tian, Shangshang Yang, Lei Zhang, Fuchen Duan, and Xingyi Zhang

***Abstract*—As a branch of frequent pattern mining, task-oriented pattern mining has received increasing attention due to its broad application scenarios. So far, the lexicographic subset tree based algorithm and the multi-objective evolutionary algorithm are two effective approaches for finding the most frequent and complete pattern in task-oriented applications. However, both of them could suffer from heavy computational cost since their runtime increases rapidly as the transaction dataset is scaled up. To address this issue, this work regards the task-oriented pattern mining as a data-driven optimization problem, and solves it by a surrogate-assisted multi-objective evolutionary algorithm. Based on the framework of our previous multi-objective evolutionary algorithm for task-oriented pattern mining, the proposed algorithm estimates the objective values of most solutions using an ensemble of surrogates instead of the real objective functions, thereby highly improving the efficiency of the algorithm. Experimental results on three task-oriented applications indicate that the proposed algorithm has better efficiency than state-of-the-art algorithms.**

***Index Terms*—Task-oriented pattern mining, multi-objective optimization, data-driven optimization, surrogate.**

I. Introduction

A transaction dataset refers to a set of transactions representing the tasks that users try to achieve, where each transaction consists of many items (i.e., sub-tasks) [1]. In general, task-oriented pattern mining aims to mine the most frequent and complete pattern (i.e., set of items) in a transaction dataset for recommendation. For example when a customer is purchasing one item (e.g., hat) on an e-commerce Website, the task-oriented pattern mining is to recommend the matching items (e.g., scarf and glove) to the customer, by mining the most frequent and complete item set from all the historical shopping lists containing a hat. Due to the bright prospect of task-oriented pattern mining in many scenarios such as goods match recommendation [2], investment portfolio recommendation [3], and print area recommendation [4], it has attracted increasing research interests over the last decade.

Manuscript received –. This work was supported in part by the National Natural Science Foundation of China (Grant No. 61502001, 61502004, and 61672033), and by the Academic and Technology Leader Imported Project of Anhui University (No. J01006057). *(Corresponding author: Xingyi Zhang)*

Y. Tian is with the Institute of Physical Science and Information Technology, Anhui University, Hefei 230039, China (email: field910921@gmail.com).

S. Yang, L. Zhang, F. Duan, and X. Zhang are with the Institute of Bio-inspired Intelligence and Mining Knowledge, School of Computer Science and Technology, Anhui University, Hefei 230039, China (email: yangshang0308@gmail.com; zl@ahu.edu.cn; slow295185031@163.com; xyzhanghust@gmail.com).

In 2012, Tang *et al.* [5] proposed an algorithm called DOFIA (i.e., dominant and frequent itemset mining algorithm) to find the most frequent and complete pattern in a transaction dataset, where each pattern is measured by two measures *support* and *occupancy* [6], [7], [8]. The *support* of a pattern indicates the ratio of transactions it appears in, and the *occupancy* of a pattern indicates its occupancy rate in the transactions it appears in. As a result, *support* and *occupancy* can measure the frequency and completeness of a pattern, respectively. The DOFIA adopts a lexicographic subset tree to represent the search space, and explores the upper bound properties on *occupancy* for pruning the search space. In 2015, this algorithm was extended for mining task-oriented sequential patterns (i.e., items in each pattern have sequential order) by Zhang *et al.* [9].

It is worth noting that the performance of DOFIA is sensitive to the settings of parameters, i.e., the minimum threshold of *support*, the minimum threshold of *occupancy*, and the relative importance preference between *support* and *occupancy*. Due to the conflicting nature between these two measures, task-oriented pattern mining can be regarded as a multi-objective optimization problem (MOP). Inspired by the success of evolutionary algorithm on many real-world MOPs [10], Zhang *et al.* [2] proposed an evolutionary algorithm called MOPM (i.e., multi-objective pattern mining algorithm) to solve the task-oriented pattern mining problem from a multi-objective perspective, which can steer clear of the difficulty in striking a balance between *support* and *occupancy*. By following a similar framework to NSGA-II [11], MOPM is able to obtain a set of trade-off patterns in a single run. In addition, since there often exist many candidate patterns having the same values of *support* and *occupancy*, a new measure *area* was suggested in MOPM to break the tie. Experimental results demonstrated that MOPM outperforms DOFIA in terms of both efficiency and effectiveness.

Despite the high improvement of MOPM shown in task-oriented pattern mining, it could suffer from heavy computational cost when mining large-scale transaction datasets. This is because the evaluation of each pattern in MOPM (i.e., calculating *support*, *occupancy*, and *area*) should traverse all the transactions in the dataset, and thus the computational complexity of MOPM is mainly determined by the scale of the dataset.

As a matter of fact, task-oriented pattern mining is naturally a data-driven optimization problem [12], since it has no analytic objective function and is pursued solely based on the transaction dataset. For data-driven optimization problems, the evaluation of a solution is often computationally or financially expensive. Therefore, the surrogate-assisted evolutionary algorithms (SAEAs) [13] are employed to reduce the needed number of expensive evaluations for finding the optimal solutions. SAEAs adopt computationally efficient surrogates like Kriging model [14], artificial neural network [15], and support vector machine [16] to approximate the objective or fitness functions of solutions instead of visiting the dataset and calculating the real objective values, hence the efficiency of search process can be highly improved. In the last two decades, numerous effective SAEAs have been proposed for solving data-driven MOPs, such as ParEGO [17], GS-MOMA [18], K-RVEA [19], GCS-MOE [20], cK-RVEA [21], and MOCS-RBF [22]. A recently published survey on existing multi-objective SAEAs can be found in [23].

Inspired by data-driven optimization, this work aims to use multi-objective SAEA to improve the efficiency in solving large-scale task-oriented pattern mining problem. It is worth noting that the evaluation of a solution for task-oriented pattern mining can take around ten seconds or more on a large-scale dataset, which is much more efficient than some traditional applications of SAEAs that take hours or even days for each function evaluation [13]. Nevertheless, it is still desirable to solve the task-oriented pattern mining problem by SAEAs, since task-oriented pattern mining is applied to many online scenarios (e.g., e-commerce Website), which require the algorithm to find a good solution for users as quickly as possible. This work is not only the first attempt to tackle the task-oriented pattern mining problem via surrogate-assisted algorithm, but also the exploitation of data-driven optimization in a new application area. However, it should be noted that most existing multi-objective SAEAs are designed for solving continuous problems, whereas the task-oriented pattern mining problem is based on binary decision variables. Therefore, a novel surrogate is adopted in the proposed algorithm to address this issue. The main new contributions of this work are summarized as follows.

1) A surrogate-assisted multi-objective pattern mining algorithm, called SA-MOPM, is proposed for large-scale task-oriented pattern mining. The proposed SA-MOPM is based on the general framework of MOPM, while the objective values of most solutions are estimated by an ensemble of surrogates. In each generation, all the solutions in the current population are evaluated using real objective functions, and a radial basis function (RBF) network [24] with binary inputs is trained on the current population. Then an ensemble model is generated by combining the historical RBF networks according to their accuracies. Afterwards, the population is evolved by MOPM for some iterations, where the objective values of all the newly generated solutions are estimated by the ensemble model instead of being evaluated using real objective functions, thereby saving much runtime in traversing the transaction dataset.
2) In order to verify the efficiency of the proposed SA-MOPM, it is compared to MOPM and two state-of-the-art algorithms for task-oriented pattern mining. The experimental results on three task-oriented applications indicate that the proposed SA-MOPM exhibits better efficiency in comparison to the algorithms without surrogate.

The rest of the paper is organized as follows. In Section II, we provide a detailed introduction to multi-objective task-oriented pattern mining, then recall the surrogates used in existing SAEAs especially the RBF network. The proposed SA-MOPM is elaborated in Section III, followed by the empirical evaluations presented in Section IV. Finally, conclusions are drawn in Section V.

II. Background

A. Multi-Objective Task-Oriented Pattern Mining

A multi-objective task-oriented pattern mining problem can be mathematically defined as [2]:

$$\begin{aligned} \text{Maximize } \mathbf{f}(\mathbf{x}) =&\ \ (supp(\mathbf{x}), occu(\mathbf{x}), area(\mathbf{x})) \\ supp(\mathbf{x}) =&\ \ \frac{|\mathcal{T}_{\mathbf{x}}|}{|\mathcal{T}|} \\ occu(\mathbf{x}) =&\ \ \frac{1}{|\mathcal{T}_{\mathbf{x}}|}\sum_{\mathbf{t}\in\mathcal{T}_{\mathbf{x}}}\frac{|\mathbf{x}|}{|\mathbf{t}|} \\ area(\mathbf{x}) =&\ \ \frac{|\mathcal{T}_{\mathbf{x}}|\cdot|\mathbf{x}|}{\sum_{\mathbf{t}\in\mathcal{T}}|\mathbf{t}|} \end{aligned}, \quad (1)$$

where $\mathbf{x}$ denotes a pattern (i.e., decision variables), $\mathcal{T}$ denotes the transaction dataset to be mined, and $\mathcal{T}_{\mathbf{x}}$ denotes the set of transactions $\mathbf{x}$ appears in, i.e.,

$$\mathcal{T}_{\mathbf{x}} = \{\mathbf{t} \in \mathcal{T} | \mathbf{x} \subseteq \mathbf{t}\}. \quad (2)$$

Note that the binary encoding is adopted to fix the length of decision variables, where the length equals to the number of distinct items and each decision variable indicates whether one item is in the pattern or not. Therefore, for the MOP given in (1), the decision space is $\{0, 1\}^d$ and the objective space is $[0, 1]^3$, where d denotes the number of distinct items in the transaction dataset.

In short, multi-objective task-oriented pattern mining is to find a set of patterns for maximizing three objectives, namely, *support*, *occupancy*, and *area*. The *support* of a pattern $\mathbf{x}$ indicates the frequency it appears in a transaction dataset $\mathcal{T}$, the *occupancy* of a pattern $\mathbf{x}$ indicates its occupancy rate in $\mathcal{T}_{\mathbf{x}}$, and the *area* of a pattern $\mathbf{x}$ indicates its occupancy rate in the whole transaction dataset $\mathcal{T}$. Intuitively, the objectives *support* and *occupancy* are conflicting with each other, since a pattern containing few items can appear in many transactions but occupy a small portion of these transactions, and vice versa. As a result, task-oriented pattern mining can be regarded as an MOP.

In the last two decades, evolutionary algorithms have shown to be very promising in solving various types of MOPs [25], such as those with many objectives [26], [27], large-scale decision variables [28], [29], discrete search spaces [30], [31], and computationally expensive objectives [17], [19]. Therefore, an evolutionary algorithm for solving multi-objective task-oriented pattern mining problem was proposed by Zhang *et al.* [2], called MOPM. The procedure of MOPM is the same to NSGA-II [11], while the difference between MOPM and NSGA-II is that MOPM suggests a novel population initialization strategy to improve the search efficiency.

It is known that both the mating selection and genetic operator of NSGA-II or MOPM have a quite low computational complexity. Due to the use of efficient non-dominated sort (ENS) [32] in non-dominated sorting, the environmental selection is also very fast. In fact, the time complexities of mating selection, genetic operator, and environmental selection within one generation of MOPM are $O(n)$, $O(nd)$ and $O(n^2)$, respectively, while the time complexity of objective evaluations is $O(nd|\mathcal{T}|)$, where n denotes the population size, d denotes the number of decision variables, $|\mathcal{T}|$ denotes the number of transactions in the dataset, and $n, d \ll |\mathcal{T}|$. Therefore, the overall runtime of MOPM is mainly determined by the evaluation of solutions, and it is reasonable to improve the efficiency by saving many objective evaluations via surrogates.

B. Surrogates used in Existing Evolutionary Algorithms

Since SAEAs are proposed for reducing the computational complexity by approximating the objective or fitness functions, the development of SAEAs mainly focuses on the use of approximation (e.g., choice of surrogates and update of surrogates). By contrast, most existing SAEAs employ an existing framework (e.g., NSGA-II [11] and MOEA/D [33]) and generate offsprings by general genetic operators (e.g., simulated binary crossover [34] and polynomial mutation [35]).

In terms of the surrogates used in existing SAEAs, Kriging (also known as Gaussian process regression) [14] is the most popular technique when compared to others. Kriging has a strong mathematical basis, which has shown effectiveness in many existing SAEAs, such as SMS-EGO [36], ParEGO [17], MOEA/D-EGO [37], and K-RVEA [19]. As reported in [23], Kriging is probably one of the most powerful interpolation algorithms currently available. However, the Kriging based surrogate can hardly be used in solving the task-oriented pattern mining problem because of two reasons. First, Kriging assumes a multivariate normal distribution, which is hard to be applied to combinatorial problems; second, the number of decision variables is limited when using Kriging, whereas the task-oriented pattern mining problem usually contains many decision variables (for example in our experiments, the number of decision variables is up to 3,000).

On the other hand, there exist many SAEAs using surrogates other than Kriging, for instance, quadratic polynomial approximation [38], feedforward neural network [15], modular neural network [39], support vector machine [16], extreme learning [40], radial basis function network [41], k-nearest neighbor [42], and random forest [43], where some of them are effective in approximating the functions of combinatorial problems. For example, Zhuang *et al.* [44] adopted an RBF network to assist the mixed-integer evolution strategy [45] in solving mixed-integer optimization problems; Herrera *et al.* [46] combined SVM with NSGA-II to solve a real-world application, which has five continuous and five categorical variables; Wang and Jin [43] proposed a random forest assisted evolutionary algorithm for a real-world application, which is a constrained multi-objective combinatorial optimization problem.

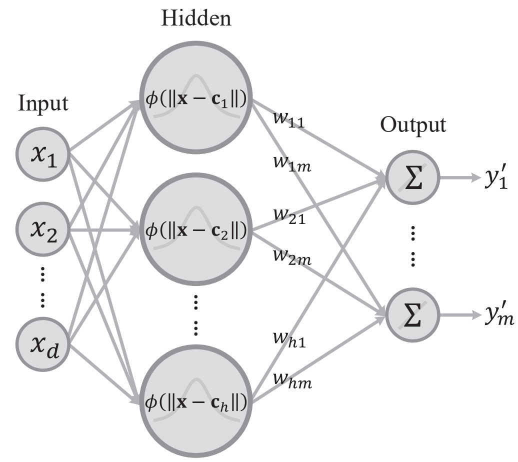


Fig. 1. Structure of an RBF network.

In the proposed SA-MOPM, the RBF network is selected as the surrogate due to its high flexibility and efficiency. A detailed introduction to RBF network is given in the next subsection.

C. Radial Basis Function (RBF) Network

RBF network is a popular type of artificial neural network, in which the RBF takes on the role of the activation function of hidden layer [24]. In comparison to general feedforward neural network, RBF network can deal with discrete inputs and be trained by the method of least squares, which is more efficient than gradient descent. Fig. 1 illustrates the structure of an RBF network. The j-th output of the network can be calculated by

$$y_j' = \sum_{i=1}^{h} w_{ij}\phi(\|\mathbf{x}-\mathbf{c}_i\|), \tag{3}$$

where w_{ij} denotes the weight between the i-th hidden neuron and the j-th output neuron, $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ denotes the inputs, $\mathbf{c}_i$ denotes the i-th center point, and $\phi(\|\mathbf{x}-\mathbf{c}_i\|)$ denotes an RBF depending only on the distance between $\mathbf{x}$ and $\mathbf{c}_i$. In particular, the Gaussian RBF is

$$\phi(\|\mathbf{x}-\mathbf{c}_i\|) = e^{-\frac{\|\mathbf{x}-\mathbf{c}_i\|^2}{2\delta_i^2}}, \tag{4}$$

where δ_i is a parameter determining the amplitude of the function for the i-th center point. In terms of a training set $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)$, the optimal weights of the RBF network can be obtained by the linear algebraic least squares method, i.e.,

$$\mathbf{w} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{y}, \tag{5}$$

where $\mathbf{\Phi}$ consists of the values of the hidden neurons.

On the other hand, the center points can be determined by the k-means clustering method based on the Euclidean distances between each pair of inputs. However, neither the Euclidean distances nor the k-means clustering can be applied

Algorithm 1: General framework of SA-MOPM

Input: n (population size)
Output: P (final population)
$P \leftarrow Initialization(n)$;
$P \leftarrow Evaluation(P)$;
$RBF \leftarrow \emptyset$;
while *termination criterion not fulfilled* **do**
  $RBF \leftarrow RBF \cup \{RBFtraining(P)\}$;
  $Surrogate \leftarrow Ensemble(RBF)$;
  $Q \leftarrow MOPM(P, Surrogate)$;
  $Q \leftarrow Q \setminus (Q \cap P)$;
  $Q \leftarrow Evaluation(Q)$;
  $P \leftarrow P \bigcup Q$;
  $[F_1, F_2, \ldots] \leftarrow NondominatedSorting(P)$;
  $CrowdDis \leftarrow CrowdingDistance(F_1, F_2, \ldots)$;
  $k \leftarrow$ Minimum value s.t. $|F_1 \bigcup \ldots \bigcup F_k| >= n$;
  Delete $|F_1 \bigcup \ldots \bigcup F_k| - n$ solutions from F_k with the worst crowding distance values;
  $P \leftarrow F_1 \bigcup \ldots \bigcup F_k$;
return Non-dominated solutions in P;

to approximate the functions of the task-oriented pattern mining problem, since the input variables in (1) are binary numbers. As a consequence, we adopt another method to calculate the distances between inputs as well as determine the center points for binary encoding based inputs, the details of which will be elaborated in Section III-B.

III. Surrogate-Assisted Multi-Objective Pattern Mining Algorithm

A. Framework of SA-MOPM

The general framework of the proposed SA-MOPM is presented in Algorithm 1, which starts with the initialization of a population P containing n solutions. As suggested in [2], the search space of the task-oriented pattern mining problem is very sparse, where randomly generated solutions are usually useless (i.e., the corresponding patterns do not appear in any transaction). To address this issue, SA-MOPM follows the heuristic initialization strategy used in MOPM, where half the initial population is made up of randomly generated solutions having only one item (i.e., only one decision variable is set to 1), and the other half consists of randomly generated solutions having all the items in one transaction.

In each generation of SA-MOPM, an RBF network is trained on the solutions in the current population P. Note that all the solutions in P have been evaluated using real objective functions, hence they can be a training set for the surrogate to approximate the MOP being solved. After training the RBF network, it is combined with the historical RBF networks to generate an ensemble model. Then, the MOPM is employed to evolve the population for several iterations, where all the newly generated offsprings are evaluated using the ensemble model instead of the real objective functions. Afterwards, the evolved population Q is evaluated using real objective functions; note that the solutions in Q which have already existed in P do not need to be evaluated again. Finally, Q is combined with the original population P, and half the combined population is retained for the next generation by non-dominated sorting and crowding distance. The above steps will repeat until a termination criterion is fulfilled, and all the non-dominated solutions in the final population are returned.

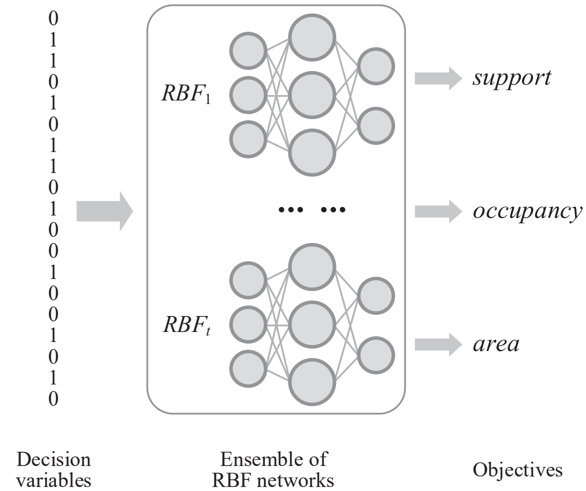


Fig. 2. The ensemble model used in SA-MOPM, where RBF_t denotes the RBF network trained at the t-th generation.

In the next two subsections, the strategies of generating an ensemble of surrogates and evolving the population with the ensemble model will be detailed separately.

B. Building the Surrogates

Fig. 2 depicts the ensemble model used in SA-MOPM. In general, the ensemble model consists of the RBF networks trained in each generation, where the inputs and outputs of each RBF network are the decision variables and objective values of a solution, respectively. When training the RBF network of the current generation, SA-MOPM first determines the center points in the hidden layer by clustering all the solutions in the current population. In this work, the Hamming distance is used to measure the distance between every two solutions in decision space, i.e., counting the number of different values between the decision variables of two solutions. In addition, the center points are determined by the k-prototype clustering method [47], which is tailored for clustering the datasets with discrete attributes. In k-prototype, several center points are randomly selected from the population, and each center point is iteratively updated by the most frequent values in the solutions close to it. The only parameter should be predefined in k-prototype is the number of center points, which is set to 10% of the population size. Afterwards, SA-MOPM calculates the parameter δ_i in the Gaussian RBF of the i-th center point according to the Hamming distances between the center point and others:

$$\delta_i = \min_{j \neq i} \|\mathbf{c}_i - \mathbf{c}_j\|, \tag{6}$$

where $\|\cdot\|$ denotes the Hamming distance. Finally, the optimal weights of the RBF network can be obtained by (5). To sum

Algorithm 2: $RBFtraining(P)$

Input: P (training set)
Output: $Model$ (RBF network)
$[\mathbf{c}_1, \mathbf{c}_2, \ldots] \leftarrow$ Randomly select several solutions from P;
for $epoch = 1$ *to* 100 **do**
  for $i = 1$ *to* $|\mathbf{c}|$ **do**
    $G_i \leftarrow \{\mathbf{x} \in P \,|\, \|\mathbf{x} - \mathbf{c}_i\| = \min_{j=1,\ldots,|\mathbf{c}|} \|\mathbf{x} - \mathbf{c}_j\|\}$;
  for $i = 1$ *to* $|\mathbf{c}|$ **do**
    for $j = 1$ *to* $|\mathbf{c}_i|$ **do**
      if *the j-th decision variable of more than half the solutions in G_i is 1* **then**
        $c_{ij} \leftarrow 1$;
      else
        $c_{ij} \leftarrow 0$;
$[\delta_1, \delta_2, \ldots] \leftarrow$ Calculate the parameters in the Gaussian RBF by (6);
$Model \leftarrow$ Calculate the weights in the network by (5);
return $Model$;

up, the procedure of training the RBF network in SA-MOPM is concluded in Algorithm 2.

As suggested in many SAEAs [18], [48], [49], an ensemble of surrogates can further improve the effectiveness of the algorithms. Therefore in the proposed SA-MOPM, an ensemble model is established by taking the historical RBF networks into account. Specifically, the output of the ensemble model is

$$\mathbf{y}'_{ens} = \sum_{i=1}^{t} \lambda_i \mathbf{y}'_i, \tag{7}$$

where $\mathbf{y}'_i$ denotes the output of the RBF network trained at the i-th generation and t is current generation number. Besides, λ_i denotes the weight of the i-th RBF network in the ensemble model, and is calculated by

$$\lambda_i = \frac{\sum_{j=1, j\neq i}^{t} \epsilon_j}{(t-1)\sum_{j=1}^{t} \epsilon_j}, \tag{8}$$

where ϵ_j measures the accuracy of the j-th RBF network in terms of the current population. In this work, the root mean square error (RMSE) is adopted as ϵ, i.e.,

$$\epsilon = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{m}(y_{ij} - y'_{ij})^2}{nm}}, \tag{9}$$

where n denotes the population size, m denotes the number of objectives, y denotes the real objective value, and y' denotes the objective value estimated by RBF network.

It is worth noting that some algorithms cluster the solutions into several groups and build one local model for each group of solutions [50], [51], the motivation of which is completely different from the proposed SA-MOPM. In SA-MOPM, the clustering of solutions is used for training an RBF network, while one RBF network is trained based on all the solutions in each generation. Therefore, the proposed SA-MOPM uses an ensemble model rather than local models.

C. Surrogate-Assisted Search

After obtaining the RBF network, SA-MOPM evolves the population for several iterations, where the objective values of all the newly generated offsprings are calculated using (3) and (7) instead of the real objective functions in (1). This procedure will terminate after 50 iterations, or 90% solutions in the population have not been changed for 3 consecutive iterations. In each iteration, SA-MOPM evolves the population by the same procedure to MOPM, i.e., selecting parents by binary tournament selection, generating offsprings by one-point crossover and bitwise mutation, and truncating the combined population by non-dominated sorting and crowding distance based environmental selection.

To summarize, the difference between MOPM and SA-MOPM lies in the generation of offsprings Q. In MOPM, the offsprings are directly generated by genetic operators, while in SA-MOPM, they are obtained by evolving the population for several iterations. So the offsprings evaluated in one generation of SA-MOPM are more than those evaluated in one generation of MOPM. However, it should be noted that the time complexity of evaluating one solution using the model is $O(dh)$, while the time complexity of evaluating one solution using the real objective functions is $O(d|\mathcal{T}|)$, where d, h and $|\mathcal{T}|$ denote the number of decision variables, the number of center points, and the number of transactions, respectively. Due to the fact that $h \ll |\mathcal{T}|$ (for example in our experiments, h is set to 10 while $|\mathcal{T}|$ is up to 500,000), the computational cost of the surrogate-assisted search in SA-MOPM is insignificant in comparison to the whole algorithm.

IV. Experimental Studies

In this section, we first verify the efficiency of the proposed SA-MOPM by comparing it with MOPM on a batch of large-scale synthetic transaction datasets. Then SA-MOPM is compared with MOPM, DOFIA, and MAFIA on two real-world transaction datasets. Lastly, the sensitivity analysis of the number of center points in RBF network is studied. All the experiments are conducted on a server with two 2.00 GHz Intel Xeon E7-4830 CPUs and the Windows Server 2008 operating system.

A. Transaction Datasets

Three types of transaction datasets are involved in the experiments. The first type is the datasets generated by the IBM synthetic data generator [52], where one can vary the scale of the generated transaction dataset to investigate the variation of the efficiency of algorithms. To create a transaction dataset, a number of patterns with predefined size are generated by randomly picking up items, then the dataset is created by generating a number of transactions with predefined size, where each transaction consists of several randomly selected patterns. The detailed procedure for creating a transaction dataset can be found in [53].

The second type is the real-world Taobao shopping dataset from Tianchi Data Lab [54]. This dataset consists of a customers' historical behavior dataset containing 1,103,702 transactions, and a human-labeled goods match ground truth dataset containing 4,009 transactions. In the experiments, each time we select one transaction from the ground truth dataset as the query and let the first item in the query be the preference of the user, then the transaction dataset to be mined consists of all the transactions in the historical behavior dataset that contain this item. In this way, the performance of an algorithm can be evaluated according to the difference between the transaction in the ground truth dataset and the pattern obtained by the algorithm. The overall performance of an algorithm can be obtained by averaging its performance over all the 4,009 transactions in the ground truth dataset.

The last type is the real-world print area dataset in Smart Print [5], which is the human-labeled dataset of selected print areas on 2,000 Web pages from 100 major print-worthy Websites. Similarly, each time we select one Web page (i.e., transaction) as the query, and the transaction dataset to be mined consists of all the Web pages from the same Website. Then the performance of an algorithm is evaluated according to the difference between the print areas in the Web page and the pattern obtained by the algorithm, and the overall performance is obtained by averaging the performance over all the 2,000 Web pages in the dataset.

B. Parameter Settings

For DOFIA, both the minimum thresholds of *support* and *occupancy* are set to 0; and in order to approximate the best performance of DOFIA, the relative importance preference λ between *support* and *occupancy* is varied from 0 to 1 with an increment of 0.2. Similarly, the minimum threshold of *support* in MAFIA is varied from 0 to 1 with an increment of 0.2

For MOPM and SA-MOPM, the population size is set to 100, and the probabilities of crossover and mutation are set to 1.0 and $1/d$, respectively, where d denotes the number of decision variables. The number of center points in the RBF network in SA-MOPM is set to 10% of the population size, i.e., 10. Besides, since MOPM and SA-MOPM are stochastic algorithms, both of them are executed for 20 times on each dataset and the average performance metric value is recorded.

C. Comparison on Synthetic Transaction Datasets

In order to verify the performance of the surrogates in SA-MOPM, it is compared with MOPM on a batch of large-scale synthetic transaction datasets. In this experiment, the number of real function evaluations is adopted as the termination criterion for the two algorithms, which is set to 500. Table I presents the hypervolume (HV) [55] values obtained by MOPM and SA-MOPM. For each obtained population, we calculate its exact HV value by the code provided in PlatEMO [56]. The reference point for HV calculation is set to the origin, since all the objectives of the task-oriented pattern mining problem are to be maximized. In addition, Table I also presents the Wilcoxon rank sum test result with a significance level of 0.05, where the symbols '$+$', '$-$' and '$\approx$' indicate that the result obtained by MOPM is significantly better, significantly worse and statistically similar to that obtained by SA-MOPM, respectively.

TABLE I
AVERAGE HV VALUES OF MOPM AND SA-MOPM ON LARGE-SCALE SYNTHETIC TRANSACTION DATASETS. THE BEST RESULT IN EACH ROW IS HIGHLIGHTED.

Number of transactions	Length of transactions	MOPM	SA-MOPM
100,000	100	1.0482e-2 $-$	**1.1334e-2**
200,000		1.0566e-2 $-$	**1.1641e-2**
300,000		1.0918e-2 $-$	**1.1689e-2**
400,000		1.1204e-2 $-$	**1.1738e-2**
500,000		1.1536e-2 $-$	**1.1713e-2**
500,000	200	**7.0601e-3** $+$	6.8091e-3
	300	**5.4339e-3** $\approx$	5.1822e-3
	400	4.9394e-3 $\approx$	**4.9733e-3**
	500	4.1043e-3 $-$	**5.9275e-3**
$+/-/\approx$		1/6/2	

'$+$', '$-$' and '$\approx$' indicate that the result obtained by MOPM is significantly better, significantly worse and statistically similar to that obtained by SA-MOPM, respectively.

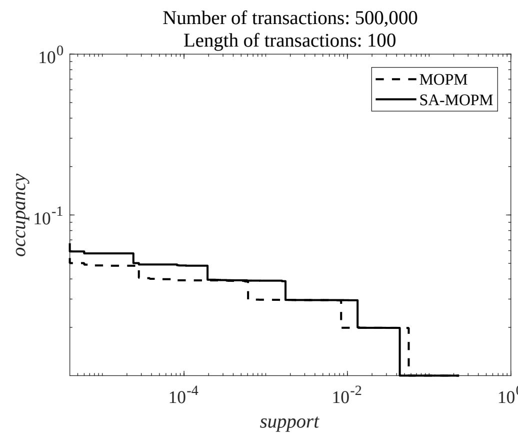


Fig. 3. Median attainment surface of MOPM and SA-MOPM when the number of transactions is 500,000 and the average length of transactions is 100. Note that both $support$ and $occupancy$ are to be maximized.

According to the statistical results in Table I, it can be seen that SA-MOPM obtains significantly better results than MOPM on most of the large-scale transaction datasets, hence the efficiency of the proposed SA-MOPM can be verified. In addition, Fig. 3 plots the median attainment surface [57] of MOPM and SA-MOPM in terms of $support$ and $occupancy$, when the number of transactions is 500,000 and the average length of transactions is 100. It is obvious from the figure that SA-MOPM outperforms MOPM, which is attributed to the ensemble model. For example, for the experiment shown in Fig. 3, the mean squared errors (MSEs) of the ensemble model obtained at the last generation of SA-MOPM are 1.4070e-2, 2.1714e-2, and 1.3998e-2 in terms of the three objective functions, respectively, hence the ensemble model can give good estimates of the objective functions and save many real-world function evaluations.

For further observation, Fig. 4 depicts the convergence profile of HV obtained by MOPM, SA-MOPM, and SA-MOPM* when the number of transactions is 500,000 and the average length of transactions is 100. Note that SA-

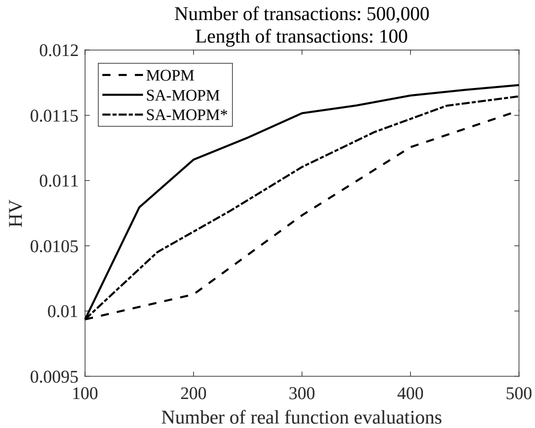


Fig. 4. Convergence profile of HV obtained by MOPM, SA-MOPM, and SA-MOPM* (i.e., with single surrogate) when the number of transactions is 500,000 and the average length of transactions is 100.

TABLE II
AVERAGE PRECISION, RECALL, AND F_1 VALUES (IN PERCENTAGE) OF DOFIA, MAFIA, MOPM, AND SA-MOPM ON THE TAOBAO SHOPPING DATASET. THE BEST RESULT IN EACH COLUMN IS HIGHLIGHTED.

	Pre	Rec	F_1
DOFIA ($\lambda = 0.0$)	21.13	07.59	10.72
DOFIA ($\lambda = 0.2$)	21.13	07.59	10.72
DOFIA ($\lambda = 0.4$)	05.79	91.01	08.15
DOFIA ($\lambda = 0.6$)	03.94	99.48	07.52
DOFIA ($\lambda = 0.8$)	03.94	99.48	07.52
DOFIA ($\lambda = 1.0$)	03.92	**99.50**	07.50
MAFIA (min *support* $= 0.0$)	21.13	07.59	10.72
MAFIA (min *support* $= 0.2$)	21.13	07.59	10.72
MAFIA (min *support* $= 0.4$)	21.13	07.59	10.72
MAFIA (min *support* $= 0.6$)	02.62	01.15	01.51
MAFIA (min *support* $= 0.8$)	0	0	0
MAFIA (min *support* $= 1.0$)	0	0	0
MOPM	28.74	57.61	**26.27**
SA-MOPM	**32.26**	55.72	25.87

MOPM* is a variant of SA-MOPM, in which only the RBF network trained at the current generation is used to estimate the objective values of offsprings. It can be seen from Fig. 4 that SA-MOPM has the fastest convergence speed, which is followed by SA-MOPM* and MOPM. As a consequence, it can be confirmed that the surrogates can improve the efficiency in solving the task-oriented pattern mining problem, and the ensemble model is more efficient than a single surrogate.

D. Comparison on Real-World Transaction Datasets

In this subsection, SA-MOPM is compared with MOPM, DOFIA, and MAFIA on the Taobao shopping dataset and the print area dataset. Three metrics are considered in this experiment, namely, precision Pre, recall Rec, and F_1, where

$$\begin{aligned} Pre &= \frac{|P \bigcap G|}{|P|} \\ Rec &= \frac{|P \bigcap G|}{|G|} \\ F_1 &= 2 \times \frac{Pre \times Rec}{Pre + Rec} \end{aligned} \tag{10}$$

with P denoting a pattern obtained by algorithms and G denoting the ground truth. Since DOFIA and MAFIA can only obtain a single solution in each run, we compare the patterns found by them to the ground truth. For MOPM and SA-MOPM, the one having the farthest Euclidean distance to the origin in the final population is used in the assessment. In addition, for a better performance of MOPM and SA-MOPM, they will terminate after 50 generations, or 90% solutions in the population have not been changed for 3 consecutive generations.

TABLE III
AVERAGE PRECISION, RECALL, AND F_1 VALUES (IN PERCENTAGE) OF DOFIA, MAFIA, MOPM, AND SA-MOPM ON THE PRINT AREA DATASET. THE BEST RESULT IN EACH COLUMN IS HIGHLIGHTED.

	Pre	Rec	F_1
DOFIA ($\lambda = 0.0$)	93.14	93.65	92.14
DOFIA ($\lambda = 0.2$)	93.20	**95.25**	93.22
DOFIA ($\lambda = 0.4$)	92.42	94.56	92.29
DOFIA ($\lambda = 0.6$)	91.96	93.98	91.63
DOFIA ($\lambda = 0.8$)	91.88	92.76	90.71
DOFIA ($\lambda = 1.0$)	91.21	91.50	89.55
MAFIA (min *support* $= 0.0$)	91.50	93.16	90.72
MAFIA (min *support* $= 0.2$)	90.77	90.78	89.52
MAFIA (min *support* $= 0.4$)	87.80	87.56	86.06
MAFIA (min *support* $= 0.6$)	78.76	77.15	75.86
MAFIA (min *support* $= 0.8$)	67.19	63.30	62.11
MAFIA (min *support* $= 1.0$)	40.77	33.42	33.55
MOPM	**93.88**	94.85	**93.40**
SA-MOPM	93.59	94.70	93.07

Table II lists the precision, recall, and F_1 values (in percentage) obtained by DOFIA, MAFIA, MOPM, and SA-MOPM on the Taobao shopping dataset. It can be observed from the table that SA-MOPM obtains the best Pre value, DOFIA obtains the best Rec value when $\lambda = 1.0$, and MOPM obtains the best F_1 value. Besides, the F_1 value of SA-MOPM is slightly less than that of MOPM, but significantly larger than those of DOFIA and MAFIA. On the other hand, Table III presents the results obtained by the four algorithms on the print area dataset. It can be found that MOPM obtains the best Pre value and F_1 value, while all the three metric values of SA-MOPM are slightly less than those of MOPM.

It worth noting that SA-MOPM is not significantly better than MOPM on these two datasets, which is mainly due to the fact that both of them are executed for sufficient

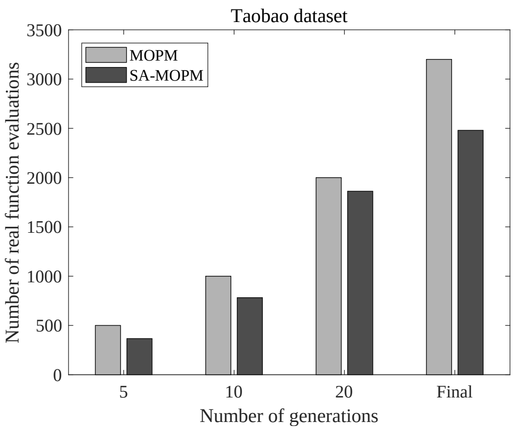


Fig. 5. Number of real function evaluations consumed by MOPM and SA-MOPM on the Taobao shopping dataset.

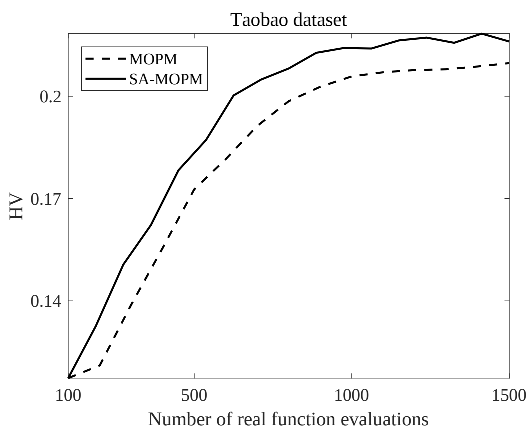


Fig. 6. Convergence profile of HV obtained by MOPM and SA-MOPM on the Taobao shopping dataset.

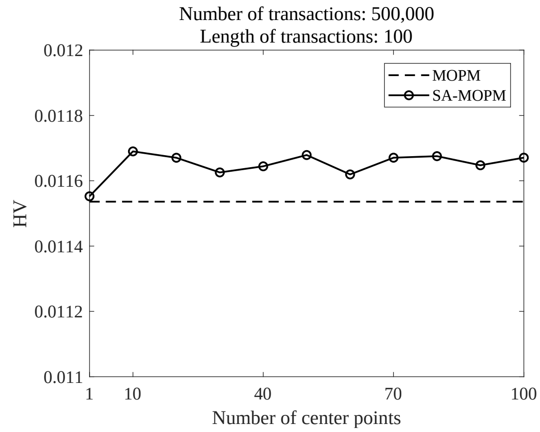


Fig. 7. Average HV value of MOPM and SA-MOPM when the number of transactions is 500,000 and the average length of transactions is 100, where the number of center points in SA-MOPM is varied from 1 to 100.

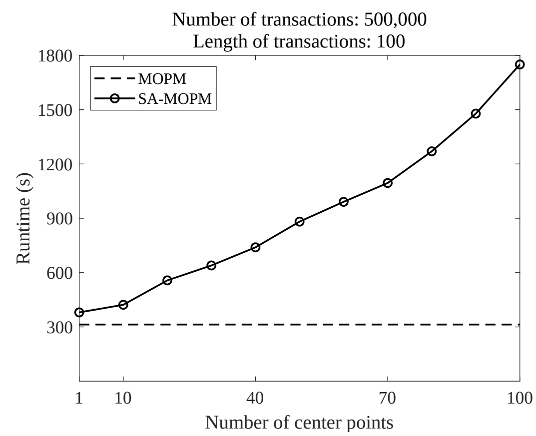


Fig. 8. Average runtime (in second) of MOPM and SA-MOPM when the number of transactions is 500,000 and the average length of transactions is 100, where the number of center points in SA-MOPM is varied from 1 to 100.

generations. In this case, SA-MOPM may not be able to obtain better solutions than MOPM by the surrogates. Nevertheless, according to the number of real function evaluations consumed by MOPM and SA-MOPM on the Taobao shopping dataset shown in Fig. 5, the real function evaluations consumed by SA-MOPM is in fact less than that consumed by MOPM. Therefore, SA-MOPM has competitive effectiveness and better efficiency than MOPM. Moreover, Fig. 6 plots the convergence profile of HV obtained by MOPM and SA-MOPM on the Taobao shopping dataset, where each algorithm is run for only 1500 real function evaluations. It is obvious that SA-MOPM converges faster than MOPM, hence SA-MOPM can be more effective than MOPM if the number of real function evaluations is limited.

E. Sensitivity Analysis of the Number of Center Points

As discussed in Section IV-B, the number of center points in the RBF network in SA-MOPM is set to 10. Here, we investigate the influence of the number of center points on the efficiency of SA-MOPM.

Fig. 7 shows the average HV value of MOPM and SA-MOPM when the number of transactions is 500,000 and the average length of transactions is 100, where the number of center points in SA-MOPM is set to $1, 10, 20, \dots, 100$. It turns out that the HV value fluctuates slightly when the number of center points is equal to or larger than 10, while the HV value decreases significantly when the number of center points is less than 10. In addition, Fig. 8 shows the average runtime (in second) of MOPM and SA-MOPM with the same setting, from which it is obvious that the runtime of SA-MOPM is relatively low when the number of center points is equal to or less than 10, and the runtime increases steadily when the number of center points is varied from 10 to 100. This is because more center points can take longer time for training RBF networks and evaluating solutions. As a consequence, SA-MOPM can have the best performance and relatively low runtime when the number of center points is set to 10.

V. CONCLUSIONS

Multi-objective evolutionary algorithm has shown to be promising in solving the task-oriented pattern mining problem, but the objective evaluation of each solution should traverse the transaction dataset, therefore it could suffer from heavy computational cost when the transaction dataset is large.

To address this issue, this work proposes a multi-objective surrogate-assisted evolutionary algorithm for improving the efficiency in solving large-scale task-oriented pattern mining problem, called SA-MOPM. The proposed SA-MOPM estimates the objective values of most solutions by an ensemble of surrogates instead of calculating their real objective values, and thus saves much runtime in traversing the transaction dataset. In comparison to state-of-the-art algorithms, the proposed SA-MOPM can improve the efficiency in solving the task-oriented pattern mining problem.

In SA-MOPM, the inputs and outputs of the RBF network are the decision variables and objective values, respectively, and the objective functions are totally estimated by the surrogates based on the solutions evaluated using real objective functions. In other words, SA-MOPM regards the task-oriented pattern mining problem as a black-box problem. On the contrary, we would like to further improve the performance by extracting useful information from the dataset and taking advantage of it in the search process. For instance, we can eliminate the items that rarely appear in the transactions, so that the decision space is highly reduced and the efficiency can be improved. Besides, it is also desirable to design customized genetic operators [58] for the task-oriented pattern mining problem.

References

[1] R. Agrawal, T. Imielinski, and A. N. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of International Conference on Management of Data*, 1993, pp. 207–216.

[2] 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.

[3] M. Fasanghari and G. A. Montazer, "Design and implementation of fuzzy expert system for tehran stock exchange portfolio recommendation," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6138–6147, 2010.

[4] L. Zhang, L. Tang, P. Luo, E. Chen, L. Jiao, M. Wang, and G. Liu, "Harnessing the wisdom of the crowds for accurate web page clipping," in *Proceedings of International Conference On Knowledge Discovery and Data Mining*, 2012, pp. 570–578.

[5] 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.

[6] K. Gade, J. Wang, and G. Karypis, "Efficient closed pattern mining in the presence of tough block constraints," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 138–147.

[7] F. Geerts, B. Goethals, and T. Mielikainen, "Tiling databases," in *Proceedings of 7th International Conference Discovery Science*, 2004, pp. 278–289.

[8] H. Xiong, P.-N. Tan, and V. Kumar, "Hyperclique pattern discovery," *Data Mining and Knowledge Discovery*, vol. 13, no. 2, pp. 219–242, 2006.

[9] L. Zhang, P. Luo, L. Tang, E. Chen, Q. Liu, M. Wang, and H. Xiong, "Occupancy-based frequent pattern mining," *ACM Transactions on Knowledge Discovery from Data*, vol. 10, no. 2, pp. 14:1–14:33, 2015.

[10] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

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

[12] H. Wang, Y. Jin, and J. O. Jansen, "Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 939–952, 2016.

[13] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm & Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.

[14] M. T. M. Emmerich, K. C. Giannakoglou, and B. Naujoks, "Single- and multi-objective evolutionary optimization assisted by Gaussian random field metamodels," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 421–439, 2006.

[15] N. Azzouz, S. Bechikh, and L. B. Said, "Steady state IBEA assisted by MLP neural networks for expensive multi-objective optimization problems," in *Proceedings of the Conference on Genetic and Evolutionary Computation*, 2014, pp. 581–588.

[16] I. Loshchilov and M. Schoenauer, "Comparison-based optimizers need comparison-based surrogates," in *Proceedings of the International Conference on Parallel Problem Solving From Nature*, 2010, pp. 364–373.

[17] J. Knowles, "ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50–66, 2006.

[18] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 329–355, 2010.

[19] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya, "A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 129–142, 2018.

[20] J. Luo, A. Gupta, Y. S. Ong, and Z. Wang, "Evolutionary optimization of expensive multiobjective problems with co-sub-Pareto front Gaussian process surrogates," *IEEE Transactions on Cybernetics*, 2018, in press.

[21] T. Chugh, K. Sindhya, K. Miettinen, J. Hakanen, and Y. Jin, "On constraint handling in surrogate-assisted evolutionary many-objective optimization," in *Proceedings of the 14th parallel problem solving from nature*, 2016, pp. 214–224.

[22] R. G. Regis, "Multi-objective constrained black-box optimization using radial basis function surrogates," *Journal of Computational Science*, vol. 16, pp. 140–155, 2016.

[23] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, "A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms," *Soft Computing*, 2017, in press.

[24] D. S. Broomhead and D. Lowe, "Radial basis functions, multi-variable functional interpolation and adaptive networks," Royal Signals & Radar Establishment, Tech. Rep., 1988.

[25] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[26] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[27] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, "An indicator based multi-objective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 609–622, 2018.

[28] 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.

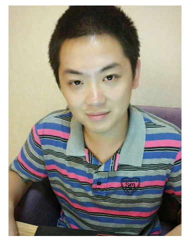
[29] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 97–112, 2018.

[30] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.

[31] X. Cai, Y. Li, Z. Fan, and Q. Zhang, "An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 508–523, 2015.

[32] Y. Tian, H. Wang, X. Zhang, and Y. Jin, "Effectiveness and efficiency of non-dominated sorting for evolutionary multi- and many-objective optimization," *Complex & Intelligent Systems*, vol. 3, no. 4, pp. 247–263, 2017.

[33] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
[34] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York: Wiley, 2001.
[35] 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.
[36] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze, "Multiobjective optimization on a limited budget of evaluations using model-assisted S-metric selection," in *Proceedings of the 2008 International Conference on Parallel Problem Solving from Nature*, 2008, pp. 784–794.
[37] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by moea/d with Gaussian process model," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 456–474, 2010.
[38] G. P. Liu, X. Han, and C. Jiang, "A novel multi-objective optimization method based on an approximation model management technique," *Computer Methods in Applied Mechanics & Engineering*, vol. 197, no. 33, pp. 2719–2731, 2008.
[39] G. Kourakos and A. Mantoglou, "Development of a multi-objective optimization algorithm using surrogate models for coastal aquifer management," *Journal of Hydrology*, vol. 479, no. 1, pp. 13–23, 2013.
[40] L. M. Pavelski, M. R. Delgado, C. P. D. Almeida, R. A. Goncalves, and S. M. Venske, "ELMOEA/D-DE: Extreme learning surrogate models in multi-objective optimization based on decomposition and differential evolution," in *Proceedings of the Brazilian Conference on Intelligent Systems*, 2014, pp. 318–323.
[41] R. Datta and R. G. Regis, "A surrogate-assisted evolution strategy for constrained multi-objective optimization," *Expert Systems with Applications*, vol. 57, pp. 270–284, 2016.
[42] J. Zhang, A. Zhou, and G. Zhang, "A classification and Pareto domination based multiobjective evolutionary algorithm," in *Proceedings of the 2015 IEEE Congress on Evolutionary Computation*, 2015, pp. 2883–2890.
[43] H. Wang and Y. Jin, "A random forest assisted evolutionary algorithm for data-driven constrained multi-objective combinatorial optimization of trauma systems," *IEEE Transactions on Cybernetics*, 2018, in press.
[44] L. Zhuang, K. Tang, and Y. Jin, "Metamodel assisted mixed-integer evolution strategies based on Kendall rank correlation coefficient," in *Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning*, 2013, pp. 366–375.
[45] R. Li, M. Emmerich, J. Eggermont, and E. G. Bovenkamp, "Mixed-integer optimization of coronary vessel image analysis using evolution strategies," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, 2006, pp. 1645–1652.
[46] M. Herrera, A. Guglielmetti, M. Xiao, and R. F. Coelho, "Metamodel-assisted optimization based on multiple kernel regression for mixed variables," *Structural and multidisciplinary optimization*, vol. 49, no. 6, pp. 979–991, 2014.
[47] Z. Huang, "Clustering large data sets with mixed numeric and categorical values," in *Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 1997, pp. 21–34.
[48] H. K. Singh, T. Ray, and W. Smith, "Surrogate assisted simulated annealing (SASA) for constrained multi-objective optimization," in *Proceedings of the 2010 IEEE Congress on Evolutionary Computation*, 2010, pp. 1–8.
[49] S. Zapotecas Martínez and C. A. Coello Coello, "MOEA/D assisted by rbf networks for expensive multi-objective optimization problems," in *Proceedings of the 15th annual conference on Genetic and evolutionary computation*. ACM, 2013, pp. 1405–1412.
[50] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multiobjective evolutionary algorithm using Gaussian process based inverse modeling," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 838–856, 2015.
[51] H. Li, Q. Zhang, and J. Deng, "Biased multiobjective optimization and decomposition algorithm," *IEEE Transactions on Cybernetics*, vol. 47, no. 1, pp. 52–66, 2017.
[52] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo, "Fast discovery of association rules," in *Advances in knowledge discovery and data mining*, 1996, pp. 307–328.
[53] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceedings of the International Conference on Very Large Data Bases*, 1994, pp. 487–499.
[54] https://tianchi.aliyun.com/datalab/dataSet.htm?id=13.
[55] 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.
[56] 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.
[57] C. M. Fonseca and P. J. Fleming, "On the performance assessment and comparison of stochastic multiobjective optimizers," in *International Conference on Parallel Problem Solving from Nature*, 1996, pp. 584–593.
[58] K. Deb and C. Myburgh, "Breaking the billion-variable barrier in real-world optimization using a customized evolutionary algorithm," in *Proceedings of the 2016 Genetic and Evolutionary Computation Conference*, 2016, pp. 653–660.



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

He is currently a Lecturer with the Institute of Physical Science and Information Technology, Anhui University, Hefei, China. His current research interests include multi-objective optimization methods and their application. He is the recipient of the 2017 IEEE Transactions on Evolutionary Computation Outstanding Paper Award.

Shangshang Yang 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, Anhui University, Hefei, China.

His current research interests include multi-objective optimization, community detection, and data mining.

Lei Zhang received the B.Sc. degree from Anhui Agriculture University, Hefei, China, in 2007, and the Ph.D. degree from University of Science and Technology of China, Hefei, China, in 2014.

He is currently a Lecturer with the School of Computer Science and Technology, Anhui University, Hefei, China. His current research interests include multi-objective optimization and applications, data mining, social network analysis, and pattern recommendation. He is the recipient of the ACM CIKM'12 Best Student Paper Award.

Fuchen Duan received the B.Sc. degree from Anhui University, Hefei, China, in 2014, where he is currently pursing the M.Sc. degree with the School of Computer Science and Technology, Anhui University, Hefei, China.

His current research interests include multi-objective optimization and data mining.

Xingyi Zhang received the B.Sc. degree from Fuyang Normal College, Fuyang, China, in 2003, and the M.Sc. and Ph.D. degrees from 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, multi-objective optimization, and membrane computing. He is the recipient of the 2017 IEEE Transactions on Evolutionary Computation Outstanding Paper Award.