

## Research Article

# Restricted Boltzmann Machine-Assisted Estimation of Distribution Algorithm for Complex Problems

Lin Bao,<sup>1,2</sup> Xiaoyan Sun ,<sup>1</sup> Yang Chen,<sup>1</sup> Guangyi Man,<sup>1</sup> and Hui Shao<sup>1</sup>

<sup>1</sup>School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, Jiangsu, China

<sup>2</sup>School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu, China

Correspondence should be addressed to Xiaoyan Sun; xysun78@126.com

Received 24 May 2018; Revised 17 August 2018; Accepted 19 August 2018; Published 1 November 2018

Academic Editor: Diyi Chen

Copyright © 2018 Lin Bao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A novel algorithm, called restricted Boltzmann machine-assisted estimation of distribution algorithm, is proposed for solving computationally expensive optimization problems with discrete variables. First, the individuals are evaluated using expensive fitness functions of the complex problems, and some dominant solutions are selected to construct the surrogate model. The restricted Boltzmann machine (RBM) is built and trained with the dominant solutions to implicitly extract the distributed representative information of the decision variables in the promising subset. The visible layer's probability of the RBM is designed as the sampling probability model of the estimation of distribution algorithm (EDA) and is updated dynamically along with the update of the dominant subsets. Second, according to the energy function of the RBM, a fitness surrogate is developed to approximate the expensive individual fitness evaluations and participates in the evolutionary process to reduce the computational cost. Finally, model management is developed to train and update the RBM model with newly dominant solutions. A comparison of the proposed algorithm with several state-of-the-art surrogate-assisted evolutionary algorithms demonstrates that the proposed algorithm effectively and efficiently solves complex optimization problems with smaller computational cost.

## 1. Introduction

Evolutionary computation (EC) has attracted considerable research attention in recent decades because of its ability to handle optimization problems [1]. EC methods, e.g., genetic algorithms (GAs), estimation of distribution algorithm (EDA), particle swarm optimization (PSO), ant colony optimization (ACO), and differential evolution (DE), have been empirically shown to perform well for a wide variety of real-world applications including load scheduling [2], energy management systems [3, 4], robotics [5], parameter control [6, 7], classification [8], and community detection [9]. These optimization problems usually have different types of decision variables, e.g., binary, integer, real, and mixed integer, and do not assume any convexity or differentiability of the objective functions and/or constraints involved. Despite the advantages, EC methods are often criticized because they have a relative slow convergence and a large amount of function evaluations (FEs), which present a serious challenge when

applying EC for computationally expensive optimization problems, e.g., fluid dynamic optimization, aerodynamic optimization, hybrid car controller optimization, or structural optimization. In solving such computationally expensive optimization problems, the heavy computational cost has a major impact on the effectiveness and efficiency of traditional ECs. To address this challenge, surrogate-assisted evolutionary algorithms (SAEAs) [10–12], such as surrogate-assisted GA [13, 14], surrogate-assisted PSO [15–17], and surrogate-assisted DE [18, 19], are receiving increasing attention in the EC community.

SAEAs have been developed to solve computationally expensive optimization problems, and they reduce the computational cost to a relatively low budget by using the computationally inexpensive surrogate models to replace the part of the expensive FEs. In the literature, common surrogate models include polynomial regression (PR) models [20], support vector machines (SVMs) [21], radial basis functions (RBFs) [16, 22, 23], artificial neural networks (ANNs) [24],

and kriging [25, 26]. Sun et al. [16] proposed a surrogate-assisted cooperative swarm optimization algorithm for high-dimensional expensive optimization problems, in which an RBF network is employed as the surrogate model. Akhtar and Shoemaker [23] proposed a parallel response surface-assisted evolutionary algorithm approach for multiobjective optimization, using RBFs to calculate surrogate response surfaces as an approximation of the computationally expensive objective function. Liu et al. [18] proposed a Gaussian process surrogate model to assist DE in solving computationally expensive optimization problems, named the surrogate model-aware evolutionary search (SMAS). However, the SMAS are trapped in local optima for computationally expensive optimization problems with discrete variables. Furthermore, Liu et al. [19] proposed a SAEA to solve expensive design optimization problems with discrete variables. Chugh et al. [25] proposed a kriging-based surrogate-assisted reference vector-guided evolutionary algorithm to approximate the computationally expensive objective function to reduce the computational cost. Min et al. [27] presented an adaptive knowledge reuse framework based on the novel idea of multiproblem surrogates and proposed a transfer evolutionary multiobjective optimization to solve multiobjective expensive optimization problems. Yang et al. [28] proposed a self-evaluation evolution algorithm to address high-dimensional computationally expensive optimization problems with the aid of metamodels.

Most existing SAEAs focus on constructing the surrogate model to accurately estimate the individual fitness by approximating the exact expensive function as closely as possible. Accordingly, these methods are limited in that they only apply the surrogate as a fitness estimation and do not adequately extract other relevant information or latent knowledge to further guide the evolution. The performance of surrogate-assisted ECs in solving complex problems will be greatly enhanced if the surrogate is designed to provide fitness estimation and evolutionary knowledge. Motivated by this, we propose a surrogate based on restricted Boltzmann machine (RBM) that can learn the distribution of the input data to implicitly describe the interactions among the variables and present an energy function to represent the relationships between the dependent and independent variables. The distribution or interactions of the variables are used to guide the evolution by using it as a probability model for EDA, and the energy function is applied to estimate the values of the complicated optimization objectives.

An RBM-assisted EDA (RBMAEDA) is proposed to solve computationally expensive optimization problems with discrete variables. First, the EDA is the evolutionary frame of the proposed RBMAEDA. An RBM model is constructed and trained by the distributed representative information of promising solutions. The probability model based on the RBM is then designed as the probability model in the EDA. Second, the fitness evaluation strategy based on the RBM energy function is developed as the surrogate model to estimate the individual fitness and reduce the computational cost. Third, model management is conducted to further improve the effectiveness of the RBMAEDA. Finally, the proposed RBMAEDA is validated by a series of benchmark

problems, and the experimental results demonstrate that the RBMAEDA can achieve satisfying performance with fewer FEs for complex optimization problems.

The main contributions of this paper are as follows:

- (1) An improved EDA based on an RBM is designed to generate new potential better individuals with discrete variables for guiding the evolutionary progress in the search space
- (2) The surrogate model based on an RBM is proposed to partly replace FEs to estimate the individual fitness and reduce the computational cost
- (3) The model management is presented to further enhance the effectiveness of the RBMAEDA by considering the relative rank of the promising individuals

The remainder of this paper is organized as follows. Section 2 briefly reviews the related techniques, including EDA and RBM. Section 3 describes the proposed RBMAEDA in detail. Section 4 demonstrates the comparative experimental results. Finally, Section 5 summarizes the paper and presents conclusions along with the scope of future extensions of this work.

## 2. Related Work

**2.1. EDA.** EDA [29–31] is a stochastic optimization algorithm based on statistical theory, which establishes the probability model from the macro perspective and describes the distributed information of the candidate solutions in the search space. Then, it predicts the promising region by statistical learning and produces new individuals by random sampling of the probability model. Meanwhile, the probability model is gradually updated with the increasing information on the better solutions. This process continues to realize the evolutionary progress and achieve excellent solutions until the termination conditions are met. Compared with GA methods, EDA makes full use of the global information of the solution space and the historical information of the promising region to explore and exploit better solutions in the evolutionary process, which effectively and efficiently improves the searching ability to solve the nonlinear optimization and the variable coupling problems. It has become a hot topic and has been successfully applied in many engineering fields.

EDA forms an effective parallel search framework based on building and sampling the probability model. According to the structure of the probability model and the relationship between the variables, probabilistic modelling techniques can be classified into univariate, bivariate, and multivariate methods. Univariate modelling methods are simple and easy to implement but do not fully utilize the linkage information of the decision variables to guide the evolutionary process. Bivariate and multivariate modelling methods can make use of the linkage information in the decision space to improve the searching ability of EDA but are generally more complex and difficult. Generally, the difficulty arises from the solution space of a problem expanding exponentially

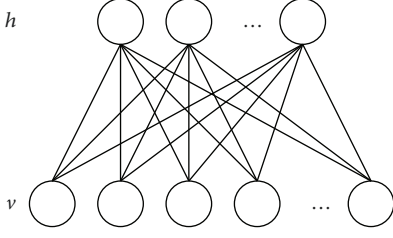


FIGURE 1: Structure of RBM.

with dimensionality so that the expanded solution space quickly exceeds the searching ability of existing EDA methods. In addition, EDA can easily overfit the distribution of the candidate solutions in the evolutionary process, which leads to an inaccurate representation of the promising region and failure of the searching process. Meanwhile, the population diversity will be gradually weakened to cause premature convergence in the EDA evolutionary process. Therefore, it is necessary to design appropriate probability models to describe the relationship between variables and the distributed information of the promising region for complex optimization problems. Researchers have proposed novel EDAs based on machine learning methods [32] and other techniques. Most EDAs are used for continuous optimization problems, whereas relatively few studies have focused on discrete combinatorial optimization problems. In addition, researches on surrogate-assisted EDA are relevant less mainly because of the difficulties in designing an appropriate probability model and surrogate model in EDA [33].

**2.2. RBM.** The RBM is an energy-based stochastic neural network with unsupervised learning, which has a two-layer network structure with symmetric connections and no self-feedback. The structure of an RBM is presented in Figure 1.

In the network structure,  $v$  is the visible layer with  $N$  visible units, which indicates the input data, and  $h$  is the hidden layer with  $M$  hidden units, which is the feature extractor. The RBM can learn the multivariate dependencies between the decision variables. Assuming that all the neurons in the RBM are binary variables, the energy function of the state  $\{v, h\}$  is used as the measure of the whole network state and is formulated as follows:

$$E_{\theta}(v, h) = - \sum_{i=1}^N \sum_{j=1}^M v_i W_{ij} h_j - \sum_{i=1}^N a_i v_i - \sum_{j=1}^M b_j h_j, \quad (1)$$

where  $v_i$  is the state of the  $i$  visible unit;  $h_j$  is the state of the  $j$  hidden unit;  $\theta = \{W_{ij}, a_i, b_j\}$  represents the model parameters;  $W_{ij}$  is the symmetric interaction weight between the visible unit  $i$  and hidden unit  $j$ ;  $a_i$  is the bias of the visible unit  $i$ ; and  $b_j$  is the bias of the hidden unit  $j$ .

The stability of the RBM network is measured through the energy function. Given the state of the visible unit, the activation state of each hidden unit is independent, and the activation probability of the  $j$  hidden unit is as follows:

$$P_{\theta}(h_j = 1|v) = \sigma \left( b_j + \sum_i v_i W_{ij} \right), \quad (2)$$

where  $\sigma(x) = 1/(1 + \exp(-x))$  is the logistic function.

Given the state of the hidden unit, the activation state of each visible unit is also independent, and the activation probability of the  $i$  visible unit is as follows:

$$P_{\theta}(v_i = 1|h) = \sigma \left( a_i + \sum_j W_{ij} h_j \right). \quad (3)$$

As for the training of RBM, Hinton [34] proposed a fast learning algorithm for RBM, i.e., the contrastive divergence (CD) algorithm, which greatly improves the learning efficiency of RBM. Subsequently, researches on RBM have been boomed, and RBM has been widely used in speech recognition [35], signal processing [36], imagery classification [37], high-dimensional time series modelling [38], etc.

### 3. RBM-Assisted EDA

The general framework of the RBM-assisted EDA is shown in Figure 2.

**3.1. Construction of Softmax RBM Based on Dominant Solutions.** RBM is an effective feature extraction technique that has self-organization, self-learning, nonlinear approximation ability, and better fault tolerance. In this paper, the visible units of RBM adopt softmax units [39] to further improve the feature extraction ability, which can increase the sparsity of input data so that the hidden units can only be activated in limited cases. The RBM model has a two-layer network structure. The visible layer  $V$  has  $N$  visible softmax units, which indicate  $N$  decision variables. Each visible softmax unit consists of  $K$  binary units. The hidden layer  $h$  has  $F$  hidden units (binary units) and indicates feature information. The architecture diagram of the softmax RBM network model is illustrated in Figure 3.

The visible softmax units of the input data are  $V = \{v_1^k, v_2^k, \dots, v_N^k\}$ ,  $k \in \{1, 2, \dots, K\}$ , which form a  $K \times N$  matrix. If the  $v_i^k$  (in the  $k$  row and  $i$  column) of the  $i$  softmax unit is  $v_i^k = 1$ , then the value of the  $i$  decision variable in the input data is  $k$  and the rest of the  $i$  softmax unit is  $v_i^{k'} = 0$ ,  $k' \neq k$ . For example, if the number of decision variables in a problem is 6 and each decision variable contains ten integer values from 0 to 9, the total search space of the problem is  $10^6$ . Therefore, the input data of a feasible solution (signed an individual  $X$ ) is composed of 6 decimal coding sequences, and the coding sequence corresponding to the individual  $X$  is as follows:

$$\text{Individual } X \quad \begin{array}{|c|c|c|c|c|c|} \hline 5 & 7 & 4 & 1 & 3 & 0 \\ \hline \end{array} \quad (4)$$

The alleles of the individual  $X$  correspond to the values of the decision variables. The code sequence of individual  $X$  is

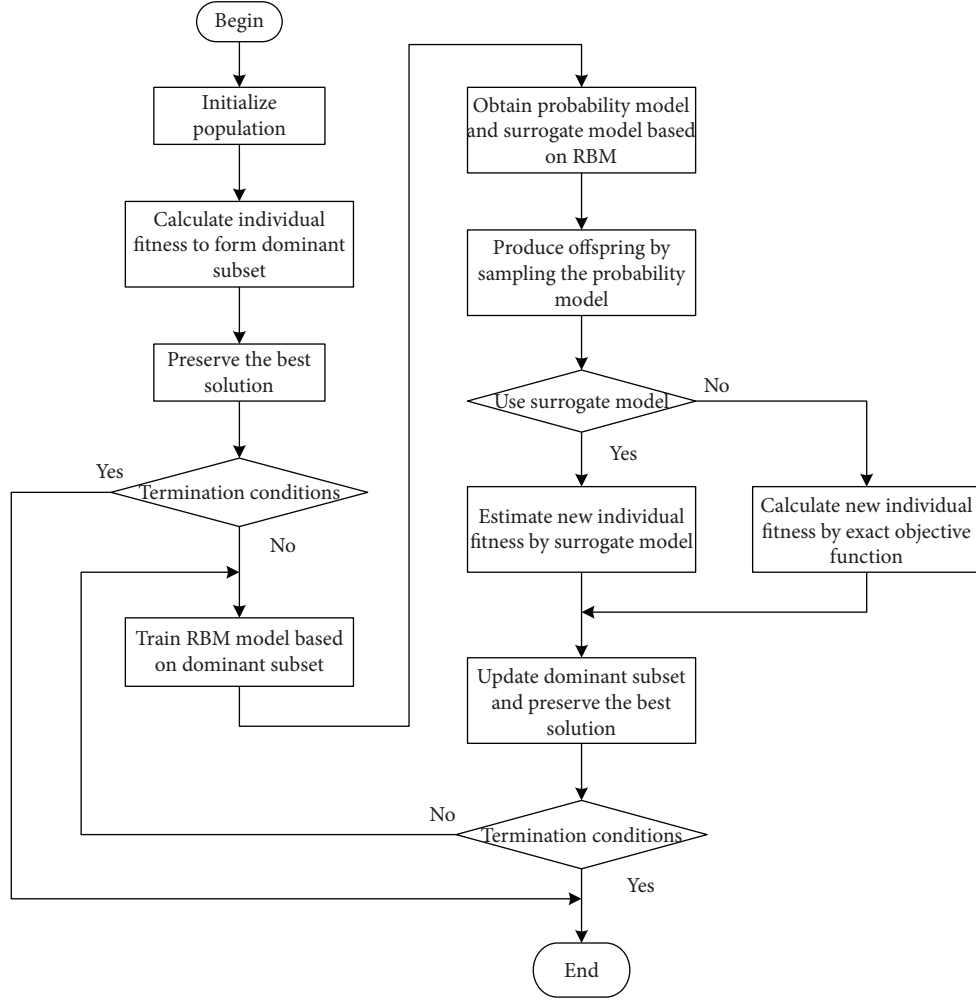


FIGURE 2: Flow diagram of the RBM-assisted EDA.

converted to softmax units consisting of a  $10 \times 6$  binary matrix  $V$ :

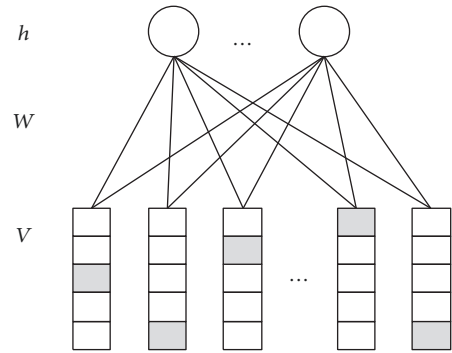
$$V = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (5)$$


FIGURE 3: Architecture diagram of softmax RBM.

Mathematically, the conditional distribution probabilities of the visible unit  $v_i^k$  and the hidden unit  $h_j$  are formulated as (6) and (7), respectively.

$$p_{\theta}(h_j = 1|V) = \sigma \left( b_j + \sum_{i=1}^m \sum_{k=1}^K v_i^k W_{ij}^k \right), \quad (6)$$

$$p_{\theta}(v_i^k = 1|h) = \frac{\exp(a_i^k + \sum_{j=1}^F h_j W_{ij}^k)}{\sum_{l=1}^K \exp(a_i^l + \sum_{j=1}^F h_j W_{ij}^l)}, \quad (7)$$

where  $W_{ij}^k$  is the connection weight between the  $k$  binary unit of the  $i$  visible softmax unit  $v_i$  and the  $j$  hidden unit  $h_j$ ;  $a_i^k$  is the bias of the visible softmax unit  $v_i^k$ ; and  $b_j$  is the bias of the hidden unit  $h_j$ .

The population of the RBMAEDA is  $\text{Pop}(g) = \{X_i, i = 1, 2, \dots, M\}$ , and the population size is  $M$ . Each individual  $X_i$  is represented by  $N$  decision variables, which corresponds to the visible softmax units of the input data  $V_i$  in the softmax RBM model. Subsequently, the individual fitness in the initial  $\text{Pop}(g)$  is calculated according to exact expensive function. The dominant subset  $\text{Sub}(g)$  is formed by selecting  $S(S \leq M)$  better individuals  $X_i \in \{X_1, X_2, \dots, X_S\}$  according to the truncation selection strategy. Then,  $\text{Sub}(g)$  is used as training data for the softmax RBM model by using a CD learning algorithm [39].

**3.2. Probability Model Based on Softmax RBM.** When the training process of the softmax RBM has been finished, the probabilistic distribution is constructed by clamping the distributed information of the alleles of the dominant individuals into the marginal probability of each decision variable in the softmax RBM model. According to (7), the activation probability  $p_{\theta}(v_i^k = 1|h)$  of the visible softmax units  $V = \{V_1, V_2, \dots, V_N\}$  is calculated by using the activation probability  $p_{\theta}(h_j = 1|V)$  of the hidden units  $h = \{h_1, h_2, \dots, h_F\}$  in the trained softmax RBM model. The probability model  $P(V)$  based on the softmax RBM is designed as follows:

$$P(V) = \begin{bmatrix} p_{\theta}(v_1^1 = 1|h) & p_{\theta}(v_2^1 = 1|h) & \cdots & p_{\theta}(v_N^1 = 1|h) \\ p_{\theta}(v_1^2 = 1|h) & p_{\theta}(v_2^2 = 1|h) & \cdots & p_{\theta}(v_N^2 = 1|h) \\ \vdots & \vdots & \vdots & \vdots \\ p_{\theta}(v_1^K = 1|h) & p_{\theta}(v_2^K = 1|h) & \cdots & p_{\theta}(v_N^K = 1|h) \end{bmatrix}. \quad (8)$$

The pseudocode for the probabilistic modelling is presented in Algorithm 1.

The probability model  $P(V)$  is constructed and calculated based on the softmax RBM model. By sampling  $P(V)$  with the roulette,  $(M - S)$  new potential individuals, which have the distributed representative information of the decision variables of the promising solutions, are subsequently generated for the next generation  $\text{Pop}(g + 1)$ .

The roulette sampling technique is formulated as follows:

$$x_i = \begin{cases} 1, & \text{if } \text{random}(0, 1) \leq p_{\theta}(v_i^k = 1|h), \\ 2, & \text{if } p_{\theta}(v_i^k = 1|h) < \text{random}(0, 1) \leq \sum_{j=1}^2 p_{\theta}(v_i^k = 1|h), \\ \vdots \\ N, & \text{if } \sum_{j=1}^{N-1} p_{\theta}(v_i^k = 1|h) < \text{random}(0, 1) \leq \sum_{j=1}^N p_{\theta}(v_i^k = 1|h), \end{cases} \quad (9)$$

where  $x_i$  is the  $i$  decision variable of the new individual and  $\text{random}(0, 1)$  is a random value ranging from  $[0, 1]$ .

**3.3. Surrogate Model Based on Softmax RBM.** According to Section 3.2, a softmax RBM model with the distributed representative information of the decision variables of the promising solutions is trained and obtained, which models the topology of the better solutions in the search space. Assuming that an individual  $X_i = \{x_{i1}, x_{i2}, \dots, x_{iN}\}$  is transformed into the softmax units  $V_i = \{v_{i1}^k, v_{i2}^k, \dots, v_{iN}^k\}$  for the softmax RBM model, the energy value  $E_{\theta}(V_i, h)$  of the individual  $X_i = \{x_{i1}, x_{i2}, \dots, x_{iN}\}$  is formulated as follows:

$$E_{\theta}(V_i, h) = - \sum_{n=1}^N \sum_{f=1}^F \sum_{k=1}^K W_{nf}^k h_f v_n^k - \sum_{n=1}^N \sum_{k=1}^K v_n^k a_n^k - \sum_{f=1}^F h_f b_f. \quad (10)$$

The energy value  $E_{\theta}(V_i, h)$  of the individual  $X_i$  indicates the adaptation level to the RBM topology of the promising solutions. The lower the energy value, the better is the stability of the RBM. It may be considered that the feature of the individual  $X_i$  is consistent with that of the promising solutions, and the individual  $X_i$  is superior others. Consequently, we can utilize the energy value  $E_{\theta}(V_i, h)$  of the individual  $X_i$  in the softmax RBM to estimate the individual fitness of  $X_i$ . The surrogate model based on the softmax RBM estimates the individual fitness of  $X_i$  and is defined as follows:

$$\hat{f}(V_i) = - \frac{E_{\theta}(V_i, h) - \min(E_{\theta}(V_i, h))}{\sum_{i=1}^M E_{\theta}(V_i, h)}, \quad (11)$$

where  $\min(E_{\theta}(V_i, h))$  represents the minimum value in all of the  $E_{\theta}(V_i, h)$ .

In addition, because the softmax RBM model is trained by unsupervised learning, the implicit distributed information extracted from the training data is relatively rough at the early stages of the EDA evolutionary process. To better obtain the distributed representative information of the decision variables of the promising solutions, the surrogate model based on the softmax RBM needs to be dynamically updated with the change of the dominant solutions in the evolutionary process.

In this paper, the top 50% individuals are selected from the current  $\text{Sub}(g)$  as the subset  $D_{\text{top}}$  by the surrogate model, while the top 50% individuals are also selected from  $\text{Sub}(g)$  as the subset  $D_{\text{top}}'$  by the real fitness function. The proportional similarity coefficient  $\gamma$  defined in (12) is used:

$$\gamma = \frac{\text{Num}_{\text{intersection}}}{\text{Num}_{\text{real}}}, \quad (12)$$

where  $\text{Num}_{\text{intersection}}$  is the size of the intersection between  $D_{\text{top}}$  and  $D_{\text{top}}'$ ; and  $\text{Num}_{\text{real}}$  is the number of  $D_{\text{top}}'$ .

Accordingly, the  $\gamma$  threshold is set to  $\gamma_{\text{th}}$ . If  $\gamma$  is greater than  $\gamma_{\text{th}}$ , then the surrogate model based on the softmax



```

Begin
Do while (maximum number of training epochs is not reached)
  #Positive Phase
  1. Construct the conditional probability of the hidden units  $p_{\theta}(h_j | V)$  given the visible softmax values according to (6)
  2. From  $p_{\theta}(h_j | V)$ , sample the states of the hidden units  $\langle h_j \rangle_0$ 
  #Negative Phase
  3. Construct the conditional probability of the visible softmax units  $p_{\theta}(v_i^k | h)$  given the states of the hidden units according to (7).
     Reconstruct the states of the visible softmax units  $\langle v_i \rangle_1$  by sampling the constructed conditional probability  $p_{\theta}(v_i^k | h)$ 
  4. Construct the conditional probability of the hidden units  $p_{\theta}(h_j | V)$  given the sampled visible softmax values according to (6).
     Reconstruct again the states of the hidden units  $\langle h_j \rangle_1$  by sampling the constructed conditional probability  $p_{\theta}(h_j | V)$ 
  #Updating of weights
  5. Update the weights and biases
End Do
  #Construction of the probability model
  6. Calculate the Probability Model  $P(V)$  according to (8).
End

```

ALGORITHM 1: Pseudocode of the probabilistic modelling.

```

Begin
  1. Initialization: at generation  $g = 0$ , randomly generate  $M$  candidate solutions to form the initial population  $Pop(g)$ 
Do while (Termination conditions are not met)
  2. Evaluation: According to the real fitness function, calculate the individual fitness of all solutions in  $Pop(g)$ , and preserve the
     best solution in the current population
  3. Selection: Select  $S$  better individuals by using the truncation selection strategy and rank in order to form the dominant subset
      $Sub(g)$ 
  4. Modelling: Train a softmax RBM model based on  $Sub(g)$ , and then build a probability model  $P(V)$  based on the softmax RBM
     according to (8)
  5. Surrogate model: Construct the surrogate model based on the softmax RBM. The  $\gamma$  value is calculated according to (12)
     to manage the surrogate model, and then the surrogate model estimates the individual fitness according to (10) and (11) to
     participate in the evolutionary process.
  6. Sampling: Produce  $(M - S)$  offspring by sampling the probability model  $P(V)$  with roulette sampling according to (9)
  7. Updating population: Merge the  $(M - S)$  offspring into  $Sub(g)$  to form the new population  $Pop(g + 1)$ .  $g = g + 1$ 
End Do
End

```

ALGORITHM 2: Pseudocode of RBMAEDA.

RBM can effectively replace the real fitness function and estimate the individual fitness to guarantee most of better solutions selected from the population. Meanwhile, the estimated individuals in  $D_{top}$  are reevaluated by using the real fitness function. Otherwise, the real fitness function is still used. Subsequently, the better individuals and their real fitness are added to the dominant subset  $Sub(g)$  to replace some worse individuals for updating  $Sub(g)$ . Therefore, the model management of the surrogate model based on the softmax RBM guarantees the accuracy and reliability of the individual fitness estimated by the surrogate model, which will dynamically track the feature information of the promising solutions and effectively provide support for guiding the evolutionary optimization progress.

**3.4. Implementation of RBM-Assisted EDA.** The pseudocode of the proposed RBMAEDA is presented in Algorithm 2.

Along with the softmax RBM training process and the EDA evolutionary progress alternately, the distributed

representative information learned by the softmax RBM sufficiently and accurately represents the probabilistic distribution of the decision variables of the promising solutions at later stages of the evolutionary process. Knowledge of superior solutions effectively and efficiently guides the exploration of the search space and yields the optimization progress by combining evolutionary optimization with machine learning.

## 4. Experiments and Results

In these experiments, the computing platform is Python 3.5 on a Dell computer with an Intel Core i5-4590 CPU 3.30 GHz and 4 GB RAM. To comprehensively analyze the performance of the proposed RBMAEDA, the experiments are tested with a series of benchmark tests [40, 41] shown in Table 1.

All these benchmark problems have continuous functions with discrete variables and different characteristics with discontinuous landscapes (dimensions from 6 to 30). They are all minimization problems. The number of decision

TABLE 1: Benchmark test problems.

Problem	Function	Dimension	Property
F1	Sphere	6	Multimodal
F2	Rastrigrin	10	Very rugged
F3	Ellipsoid	10	Unimodal
F4	Schwefel 2.4	15	Multimodal
F5	Rosenbrock	15	Narrow valley
F6	Step	20	Unimodal
F7	Ackley	20	Rugged
F8	Rastrigrin	30	Very rugged
F9	Griewank	30	Rugged

TABLE 2: Experimental parameters.

Parameter	Setting
Population	(70–90)* dimension
Termination conditions	Optima are found or maximum epoch is 200
Number of independent runs	10
Number of hidden units in RBM	2* number of visible units
Learning rate	0.1
Momentum	0.5–0.9
Number of training epochs in the RBM	10

variables and other characteristics is presented in Table 1. In this section, we conduct three series of experiments to verify the effectiveness of the probability model based on the softmax RBM, the surrogate model based on the softmax RBM, and the proposed RBMAEDA for complex optimization problems in Subsections 4.1, 4.2, and 4.3, respectively.

**4.1. Performance of the Probability Model Based on Softmax RBM.** To verify the effectiveness of the probability model based on the softmax RBM, EDA based on the softmax RBM (denoted by softmaxRBM-EDA) is compared with the traditional EDA and REDA-E [32] with the same common parameters. Three algorithms are different from each other in their different probability model and model updating strategy. The comparative experiments are conducted to evaluate the performance of the three algorithms using 9 benchmark problems in Table 1. In the comparative experiments, 10 independent runs are performed for each algorithm. The experimental parameters are shown in Table 2.

A series of performance indicators, including the average optimal solution, the standard deviation, the success rate, and the search time, are used to measure the performance of each algorithm. These performance indicators in the experiments are described as follows:

- (1) Avg.  $\pm$  St: the average and standard deviation of the best solutions for 10 trials
- (2) Rate (%): the success rate of reaching the global optimal solution in 10 trials, reflecting the effectiveness of each algorithm

- (3) Time (s): the total search time until reaching the best solution, reflecting the efficiency of each algorithm

The comparative experimental results for the three algorithms are shown in Table 3.

By observing the results in Table 3, the following conclusions can be obtained:

- (1) The average and standard deviation of the softmaxRBM-EDA are the smallest among the three algorithms in most of the test problems, and the success rate of the softmaxRBM-EDA is the highest among the three algorithms. The traditional EDA and REDA-E have very lower success rate and completely trapped in the local optima for F4 and F5, mainly because this type of problems has very rugged landscapes and obtains the optimal solutions at nonzero point. In particular, note that Rosenbrock function is a multimodal problem with narrow valley. EDA and REDA-E have never reached the global optimal solution for F5 in the 10 independent executions, so they are not appropriate for handling these multimodal problems with the very rugged landscapes or the narrow valley. However, softmaxRBM-EDA can be better able to solve this kind of problems
- (2) SoftmaxRBM-EDA shows a substantial improvement of solution quality and searching efficiency for those very challenging problems in terms of the very rugged landscapes and the narrowness of the optimal valley. Specifically, the softmaxRBM-EDA performed 122 iterations (approximately 153.50 s) on average to obtain the best solution with an average of 7.40 and a standard deviation of 16.94 and had an 80% success rate in reaching the global optimum for F4. EDA executed approximately 27 iterations (approximately 2.77 s) on average and reached the local optimal solution with an average of 9.0 and a standard deviation of 1.94, which had not reached the global optimum for F4 even once. REDA-E performed 87 iterations (approximately 177.51 s) on average and obtained the optimal solution with only 40% success rate for F4, which had an average of 21.0 and a standard deviation of 20.73
- (3) Regarding the search time, softmaxRBM-EDA is faster than REDA-E but slower than EDA for those problems partly because it spends time extracting the feature information to construct the probability model based on the RBM for guiding the evolutionary progress. However, softmaxRBM-EDA does not cause too much computational burden. That computational cost is deserved for jumping out of the local optima and achieving the global optima. In addition, the probability model based on statistics in EDA does not contribute to design the appropriate surrogate model in the EDA framework

In summary, the proposed softmaxRBM-EDA improves the quality of the best solutions and the efficiency of

TABLE 3: Comparative experimental results.

Problem	EDA			REDA-E			softmaxRBM-EDA		
	Avg. $\pm$ St	Rate	Time	Avg. $\pm$ St	Rate	Time	Avg. $\pm$ St	Rate	Time
F1	$0 \pm 0$	100	0.12	$0 \pm 0$	100	48.54	<b><math>0 \pm 0</math></b>	<b>100</b>	2.88
F2	$0 \pm 0$	100	0.75	$0 \pm 0$	100	70.75	<b><math>0 \pm 0</math></b>	<b>100</b>	23.45
F3	$0 \pm 0$	100	0.59	$0 \pm 0$	100	78.81	<b><math>0 \pm 0</math></b>	<b>100</b>	42.07
F4	$9.0 \pm 1.94$	0	2.77	$21.0 \pm 20.73$	40	177.51	<b><math>7.40 \pm 16.94</math></b>	<b>80</b>	153.50
F5	$33.70 \pm 62.30$	0	2.62	$2146.20 \pm 1885.78$	0	220.39	<b><math>258.0 \pm 355.60</math></b>	<b>30</b>	146.11
F6	$0 \pm 0$	100	5.27	$0 \pm 0$	100	272.49	<b><math>0 \pm 0</math></b>	<b>100</b>	242.45
F7	$0 \pm 0$	100	9.30	$0 \pm 0$	100	295.22	<b><math>0 \pm 0</math></b>	<b>100</b>	129.92
F8	$0 \pm 0$	100	10.33	$1.0 \pm 1.70$	60	1012.42	<b><math>0 \pm 0</math></b>	<b>100</b>	222.02
F9	$0 \pm 0$	100	11.88	$1.89e - 03 \pm 4.93e - 03$	60	1066.39	<b><math>0 \pm 0</math></b>	<b>100</b>	332.95

searching process and outperforms the other algorithms for complex problems with discontinuous landscapes. It is mainly because the probability model based on the softmax RBM in softmaxRBM-EDA can extract the distributed representative information of the decision variables of the promising solutions to participate in the evolutionary process. Then, softmaxRBM-EDA can generate new solutions with the gene information of the better solutions by sampling the probability model based on the softmax RBM. Furthermore, with the increase of solutions, the probability model based on the softmax RBM will be updated by the increasing feature information of the promising solutions to continually optimize the candidate solutions and comprehensively guide the evolutionary progress. SoftmaxRBM-EDA enhances the exploration and exploitation abilities to adaptively adjust the optimal direction for the solution quality and the population diversity. Therefore, softmaxRBM-EDA has the better solution quality, convergence rate, and stability for complex problems. For dynamically illustrating the evolutionary progress, the iteration evolutionary processes of the EDA, REDA-E, and softmaxRBM-EDA approaches are presented for solving F1, F4, F5, and F8 test problems. For the fairness of the comparative experiments, each algorithm will be executed and evolved sufficiently. The convergence profiles of the three algorithms for F1, F4, F5, and F8 are plotted in Figure 4.

In Figure 4, the horizontal coordinates indicate the number of iterations, and the vertical coordinates indicate the fitness of the best solution for each generation. As can be seen from Figure 4, EDA can reach the global optima of F4 and F5 along with the evolutionary progress, and REDA-E does not succeed in obtaining the global optima of F1, F5, and F8, falling into the local optima. Accordingly, softmaxRBM-EDA has achieved the best solutions for all of the problems. Although REDA-E requires less iterations than softmaxRBM-EDA to reach the final solutions, REDA-E spends much more search time in each generation and has more total time than softmaxRBM-EDA. The main reason is that the computational cost for sampling mechanism and training RBM in REDA-E is too large with increasing dimensions. Nevertheless, softmaxRBM-EDA emphasizes the exploitation of the promising regions and offers a predictive guidance on the exploration of optimal solutions. It is verified that the probability model based on

the softmax RBM is feasible and effective as a probability model of EDA in softmaxRBM-EDA for solving complex optimization problems. Therefore, softmaxRBM-EDA performs more effectively and efficiently than the other algorithms in most of the test problems.

*4.2. Performance of the Surrogate Model Based on Softmax RBM.* In this subsection, investigations are carried out to analyze the proposed surrogate model based on the softmax RBM in RBMAEDA. If the relative order relationship of the dominant individuals can be guaranteed during the surrogate model used in EDA, the individual selection based on the surrogate model and the updating of the probabilistic model will not have harmful effect on the evolutionary progress. So we can consider that the surrogate model in EDA is feasible and effective. In this experiment, one evolutionary process of RBMAEDA for the Griewank function with 10 dimensions is recorded, and the results show that RBMAEDA finds the optimal solution in the 11th generation, where  $\gamma_{th} = 0.6$ . The real individual fitness and the estimated individual fitness of the dominant subset  $Sub(g)$  will be calculated for comparison in RBMAEDA and are shown at the 2nd, 3rd, and 7th generations in Figure 5.

In Figure 5, the horizontal coordinate indicates the index of individuals, and the vertical coordinate indicates the real individual fitness and the estimated individual fitness. Figure 5 shows that the individual fitness estimated by the surrogate model based on the softmax RBM can mainly follow the trajectory of the real individual fitness. Meanwhile, the threshold of the proportional similarity coefficient  $\gamma_{th} = 0.6$  means that 60% better solutions of  $Sub(g)$  can be selected by the surrogate model. The surrogate model and the real fitness function have high similarity ranking in individual evaluation, so the relative rank of individuals in  $Sub(g)$  can be generally represented by the surrogate model based on the softmax RBM. Therefore, it is feasible and effective to estimate the individual fitness by the surrogate model based on the softmax RBM in RBMAEDA.

*4.3. Performance of the RBMAEDA.* To illustrate the performance of the proposed RBMAEDA, RBMAEDA is compared with the committee-based active learning for surrogate-assisted particle swarm optimization (CAL-SAPSO) algorithm [17] presented in 2017 on a series of benchmark



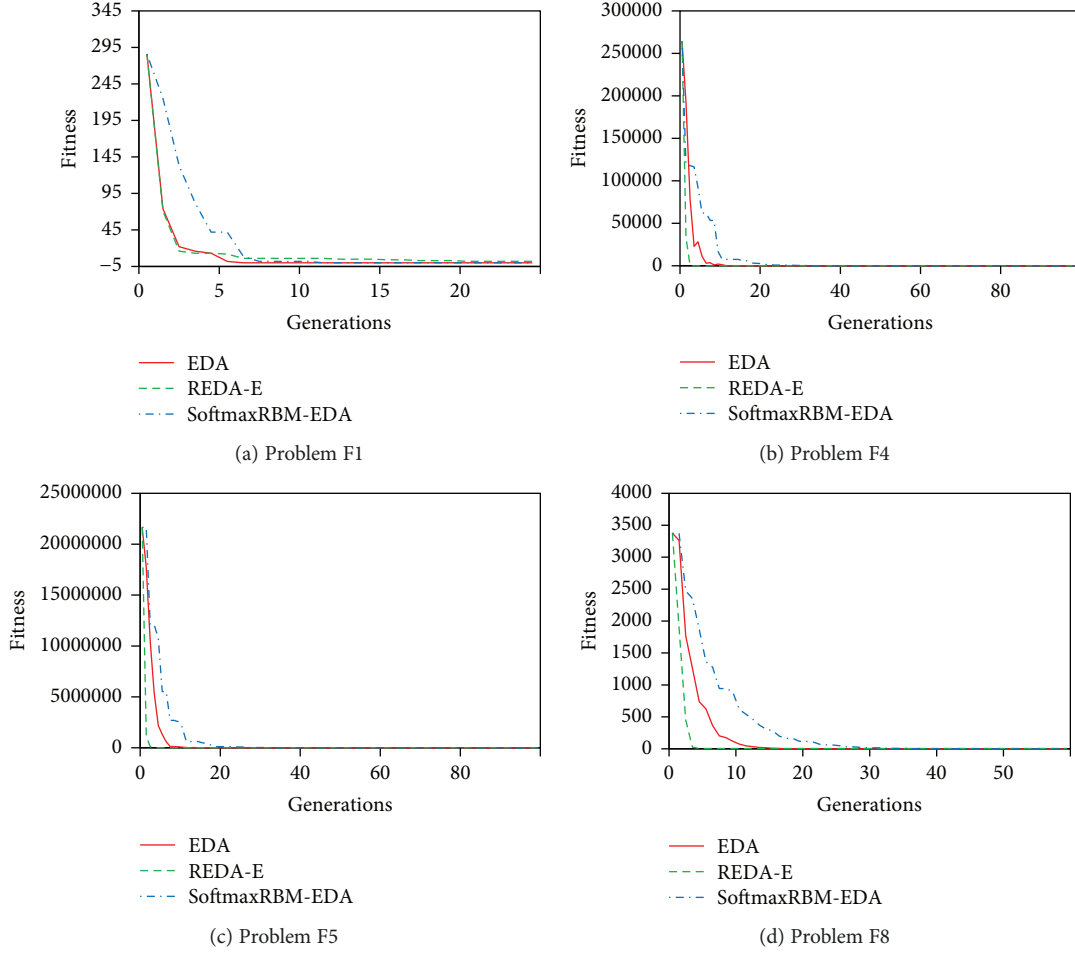


FIGURE 4: Comparative results of the EDA, REDA-E, and SoftmaxRBM-EDA.

problems for a limited computation budget. The comparison algorithm is an algorithm with outstanding performance at present, using the same comparison function to compare with the results provided in [17]. In the experiment, the average value and the standard deviation in 10 trials are used to measure the performance of these algorithms. The experimental results on benchmark problems of different dimensions are presented in Table 4.

As shown in Table 4, the following conclusions can be obtained:

- (1) RBMAEDA outperforms the CAL-SAPSO algorithm on most of the test problems. Furthermore, with the increase of the dimension of those problems, RBMAEDA has still the stable average values of the final solutions to reach close to the optima. For example, the average value and the standard deviation of RBMAEDA are  $1.0E-01 \pm 3.16E-01$  for the 10-dimension Rastrigin, while those of the 20- and 30-dimension Rastrigin are  $5.0E-01 \pm 8.50E-01$  and  $4.0E-01 \pm 6.99E-01$ , respectively
- (2) However, RBMAEDA performs worse than CAL-SAPSO on the Rosenbrock function. The main reason is that Rosenbrock function has a very narrow

and deep peak of the fitness landscape around the global optimum. In addition, the CAL-SAPSO algorithm can deal with computationally expensive optimization problems with continuous variables, which will help CAL-SAPSO reach closer to the optimum. Nevertheless, RBMAEDA is specially designed for computationally expensive optimization problems with discrete variables, for which it is more difficult to find the optimum

The experimental results demonstrate the advantages of RBMAEDA over the compared algorithm on the benchmark problems with different dimensions. Therefore, RBMAEDA can effectively solve medium-scale complex computationally expensive optimization problems with discrete variables.

To further verify the performance of the RBMAEDA, RBMAEDA is compared with two popular SAEAs, the SMAS algorithm [18] and the SMDN algorithm [19] presented in 2016, on a series of benchmark problems with a limited computation budget. In particular, the SMDN algorithm is a state-of-the-art algorithm for complex computationally expensive optimization problems with discrete variables, using the same comparison function to compare with the results provided in [19]. In the experiment, three indicators are used to measure the performance of these algorithms:

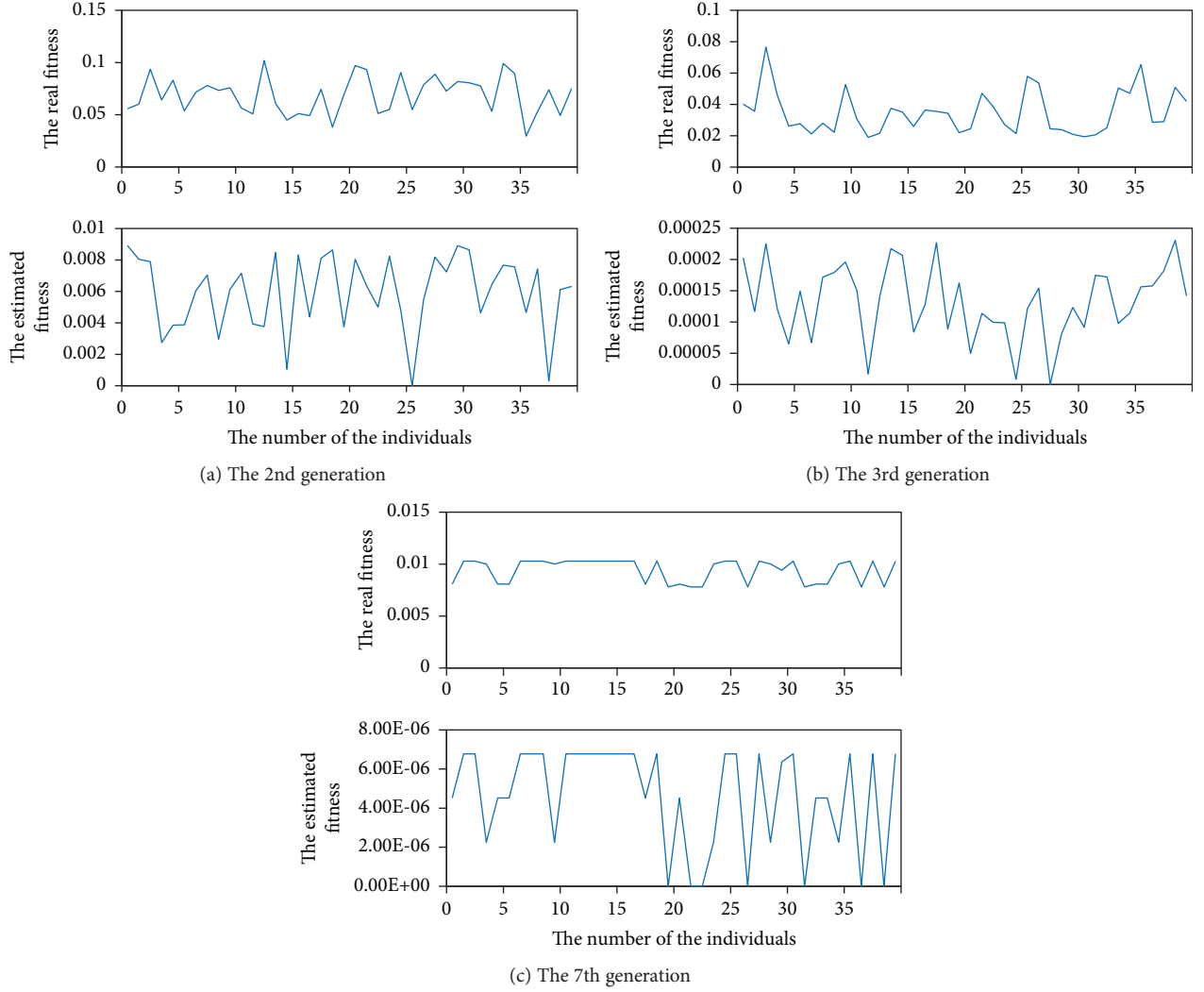


FIGURE 5: Comparative results of the real individual fitness and the estimated individual fitness.

the average value, the standard deviation, and the success rate in the 10 trials. The experimental results for benchmark problems of different dimensions are presented in Table 5.

By observing the results in Table 5, the following conclusions can be obtained:

- (1) RBMAEDA outperforms the other algorithms on most of the test problems. For example, RBMAEDA has the highest success rate in reaching optimality for the Rastrigin function and the Griewank function in three algorithms, and RBMAEDA has the best average value and the lowest standard deviation for those problems in three algorithms. The average values of the final solutions are also stable and close to the optimal of those problems
- (2) RBMAEDA has a relative lower success rate for the Rosenbrock function. It is mainly because that the Rosenbrock function has the very narrow valley of the fitness landscape, which enhances the difficulty of complex optimization problems with discrete variables in the searching process

- (3) With the increase of the dimension of these problems, RBMAEDA is still able to obtain comparable high quality solutions with a limited budget of FEs, but the success rate is gradually declining. The main reason is that the computational cost and complexity for training the softmax RBM model in RBMAEDA will increase with the dimension increasing of the problems, which leads to some trouble for the surrogate to learn the fitness landscape and estimate the individual fitness in the evolutionary process

In summary, the proposed RBMAEDA utilizes the probability model based on the softmax RBM and the surrogate model based on the softmax RBM, which makes full use of the knowledge and information of the promising solutions, to enhance the exploration and exploitation abilities in the evolutionary process. Meanwhile, RBMAEDA performs well on unimodal and multimodal problems or problems with the rugged landscapes by using a limited computing budget when the number of decision variables changes from 5 to 30. For these problems, RBMAEDA only consumes about 20% to 40% of the number of FEs of

TABLE 4: Experimental results comparing with CAL-SAPSO.

Problem (dimensions)	CAL-SAPSO Avg. $\pm$ St	RBMAEDA Avg. $\pm$ St
Ellipsoid (10)	$8.79E - 01 \pm 8.51E - 01$	<b><math>0 \pm 0</math></b>
Ackley (10)	$2.01E + 01 \pm 2.44E - 01$	<b><math>0 \pm 0</math></b>
Griewank (10)	$1.12 \pm 1.21E - 01$	<b><math>2.72E - 05 \pm 8.59E - 05</math></b>
Rastrigin (10)	$8.88E + 01 \pm 2.26E + 01$	<b><math>1.0E - 01 \pm 3.16E - 01</math></b>
Rosenbrock (10)	<b><math>1.77 \pm 3.80E - 01</math></b>	$9.55E + 01 \pm 9.29E + 01$
Ellipsoid (20)	<b><math>1.58 \pm 4.83E - 01</math></b>	$2.80 \pm 8.51$
Ackley (20)	$2.01E + 01 \pm 0$	<b><math>3.97E - 02 \pm 1.25E - 01</math></b>
Griewank (20)	$1.06 \pm 3.66E - 02$	<b><math>2.75E - 05 \pm 8.71E - 05</math></b>
Rastrigin (20)	$7.51E + 01 \pm 1.44E + 01$	<b><math>5.0E - 01 \pm 8.50E - 01</math></b>
Rosenbrock (20)	<b><math>1.89 \pm 3.32E - 01</math></b>	$2.63E + 02 \pm 1.72E + 02$
Ellipsoid (30)	$4.02 \pm 1.08$	<b><math>3.0 \pm 6.60</math></b>
Ackley (30)	$1.62E + 01 \pm 4.13E - 01$	<b><math>2.73E - 01 \pm 5.76E - 01</math></b>
Griewank (30)	$9.95E - 01 \pm 3.99E - 02$	<b><math>2.32E - 04 \pm 4.80E - 04</math></b>
Rastrigin (30)	$8.78E + 01 \pm 1.65E + 01$	<b><math>4.0E - 01 \pm 6.99E - 01</math></b>
Rosenbrock (30)	<b><math>1.76 \pm 3.96E - 01</math></b>	$1.39E + 03 \pm 9.66E + 02$

TABLE 5: Experimental results comparing with SMAS and SMDN.

Problem (dimensions)	SMAS		SMDN		RBMAEDA	
	Avg. $\pm$ St	Rate	Avg. $\pm$ St	Rate	Avg. $\pm$ St	Rate
Problem 8.3 (5)	$-733.35 \pm 3.62$	45	$-737 \pm 0$	100	<b><math>-737 \pm 0</math></b>	<b>100</b>
Rastrigin (10)	$8.30 \pm 5.12$	0	$1.25 \pm 1.16$	30	<b><math>0.1 \pm 0.32</math></b>	<b>90</b>
Ellipsoid (15)	$0 \pm 0$	100	$0 \pm 0$	100	<b><math>0 \pm 0</math></b>	<b>100</b>
Rosenbrock (15)	$84.53 \pm 99.44$	10	<b><math>0.2 \pm 0.89</math></b>	<b>95</b>	$3.03E + 02 \pm 1.89E + 02$	10
Step (20)	$0.05 \pm 0.22$	95	$0 \pm 0$	100	<b><math>0 \pm 0</math></b>	<b>100</b>
Ackley (20)	$0 \pm 0$	100	<b><math>0 \pm 0</math></b>	<b>100</b>	$3.97E - 02 \pm 1.25E - 01$	90
Griewank (20)	$0.96 \pm 0.04$	0	$0.87 \pm 0.08$	0	<b><math>2.75E - 05 \pm 8.71E - 05</math></b>	<b>90</b>

softmaxRBM-EDA to get comparable high quality solutions. Furthermore, RBMAEDA has a better solution quality, stability, convergence rate, and scalability for solving computationally expensive optimization problems with complex discontinuous landscapes.

## 5. Conclusions

In this paper, a novel RBMAEDA algorithm is proposed to solve complex computationally expensive optimization problems with discrete variables. The RBM probability model and its dynamic updating mechanism take full advantage of the feature information of the better solutions to guide the evolutionary progress. In addition, the surrogate model based on the RBM and model management are adopted to replace FEs, which enhances the searching efficiency and reduces the computational cost. The experimental results demonstrate that the proposed RBMAEDA outperforms other algorithms for most of the test problems and is effective to deal with complex computationally expensive optimization problems. These researches will further deepen and enrich the theoretical research and practical application of SAEAs. For future investigations, there is considerable development

potential in the combination of deep learning and intelligent optimization algorithms to solve complex problems.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was jointly supported by the National Natural Science Foundation of China under grant no. 61473298 and no. 61473299.

## References

- [1] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, "Handling computationally expensive multiobjective optimization problems with evolutionary algorithms-a survey," in *Technical Report Series B, Scientific Computing No. B 4/2015*,

Department of Mathematical Information Technology, University of Jyväskylä, 2015.

- [2] Q. Kang, S. W. Feng, M. C. Zhou, A. C. Ammari, and K. Sedraoui, "Optimal load scheduling of plug-in hybrid electric vehicles via weight-aggregation multi-objective evolutionary Algorithms," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2557–2568, 2017.
- [3] X. Qi, G. Wu, K. Boriboonsomsin, and M. J. Barth, "Development and evaluation of an evolutionary algorithm-based online energy management system for plug-in hybrid electric vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 2181–2191, 2017.
- [4] A. Orlandi, "Differential evolutionary multiple-objective sequential optimization of a power delivery network," *IEEE Transactions on Electromagnetic Compatibility*, vol. 60, no. 3, pp. 754–760, 2018.
- [5] Y.-D. Hong and B. Lee, "Evolutionary optimization for optimal hopping of humanoid robots," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 2, pp. 1279–1283, 2017.
- [6] M. A. Dulebenets, "Application of evolutionary computation for berth scheduling at marine container terminals: parameter tuning versus parameter control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 25–37, 2018.
- [7] Z. Zhang and Y. Tian, "A novel resource scheduling method of netted radars based on Markov decision process during target tracking in clutter," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, 2016.
- [8] M. Antonelli, D. Bernardo, H. Hagrass, and F. Marcelloni, "Multiobjective evolutionary optimization of type-2 fuzzy rule-based systems for financial data classification," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 2, pp. 249–264, 2017.
- [9] L. Zhang, H. Pan, Y. Su, X. Zhang, and Y. Niu, "A mixed representation-based multiobjective evolutionary algorithm for overlapping community detection," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2703–2716, 2017.
- [10] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 5, pp. 481–494, 2002.
- [11] 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.
- [12] Y. Jin, "Surrogate-assisted evolutionary computation: recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.
- [13] X. Y. Sun, D. W. Gong, and X. P. Ma, "Directed fuzzy graph-based surrogate model-assisted interactive genetic algorithms with uncertain individual's fitness," in *2009 IEEE Congress on Evolutionary Computation*, pp. 2395–2402, Trondheim, Norway, May 2009.
- [14] X. Sun, D. Gong, Y. Jin, and S. Chen, "A new surrogate-assisted interactive genetic algorithm with weighted semi-supervised learning," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 685–698, 2013.
- [15] C. Sun, Y. Jin, J. Zeng, and Y. Yu, "A two-layer surrogate-assisted particle swarm optimization algorithm," *Soft Computing*, vol. 19, no. 6, pp. 1461–1475, 2015.
- [16] C. Sun, Y. Jin, R. Cheng, J. Ding, and J. Zeng, "Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 4, pp. 644–660, 2017.
- [17] H. Wang, Y. Jin, and J. Doherty, "Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2664–2677, 2017.
- [18] B. Liu, Q. Zhang, and G. G. E. Gielen, "A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 180–192, 2014.
- [19] B. Liu, N. Sun, Q. Zhang, V. Grout, and G. Gielen, "A surrogate model assisted evolutionary algorithm for computationally expensive design optimization problems with discrete variables," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1650–1657, Vancouver, BC, Canada, July 2016.
- [20] J. L. Chávez-Hurtado and J. E. Rayas-Sánchez, "Polynomial-based surrogate modeling of RF and microwave circuits in frequency domain exploiting the multinomial theorem," *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 12, pp. 4371–4381, 2016.
- [21] A. Ciccazzo, G. di Pillo, and V. Latorre, "A SVM surrogate model-based method for parametric yield optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 7, pp. 1224–1228, 2016.
- [22] R. G. Regis, "Evolutionary programming for high-dimensional constrained expensive black-box optimization using radial basis functions," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 326–347, 2014.
- [23] T. Akhtar and C. A. Shoemaker, "Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection," *Journal of Global Optimization*, vol. 64, no. 1, pp. 17–32, 2016.
- [24] N. Azzouz, S. Bechikh, and L. Said, "Steady state IBEA assisted by MLP neural networks for expensive multi-objective optimization problems," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 581–588, Vancouver, BC, Canada, 2014, ACM.
- [25] 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.
- [26] J. Jang, J. M. Lee, S.-g. Cho et al., "Space-time kriging surrogate model to consider uncertainty of time interval of torque curve for electric power steering motor," *IEEE Transactions on Magnetics*, vol. 54, no. 3, pp. 1–4, 2018.
- [27] A. T. W. Min, Y.-S. Ong, A. Gupta, and C.-K. Goh, "Multi-problem surrogates: transfer evolutionary multiobjective optimization of computationally expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 8, pp. 1–14, 2015.
- [28] P. Yang, K. Tang, and X. Yao, "Turning high-dimensional optimization into computationally expensive optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 143–156, 2018.
- [29] P. Smolensky, "Information processing in dynamical systems: foundations of harmony theory," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*, D. E. Rumelhart and J. L. McClelland, Eds., pp. 194–281, McGraw-Hill, New York, 1986.



- [30] W. Dong, T. Chen, P. Tino, and X. Yao, "Scaling up estimation of distribution algorithms for continuous optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 6, pp. 797–822, 2013.
- [31] A. Kabán, J. Bootkrajang, and R. J. Durrant, "Toward large-scale continuous EDA: a random matrix theory perspective," *Evolutionary Computation*, vol. 24, no. 2, pp. 255–291, 2015.
- [32] V. A. Shim, K. C. Tan, and C. Y. Cheong, "An energy-based sampling technique for multi-objective restricted Boltzmann machine," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 6, pp. 767–785, 2013.
- [33] Y. Chen, X. Sun, D. Gong, Y. Zhang, J. Choi, and S. Klasky, "Personalized search inspired fast interactive estimation of distribution algorithm and its application," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 4, pp. 588–600, 2017.
- [34] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.
- [35] H. B. Sailor and H. A. Patil, "Novel unsupervised auditory filterbank learning using convolutional RBM for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2341–2353, 2016.
- [36] L. F. Polania and K. E. Barner, "Exploiting restricted Boltzmann machines and deep belief networks in compressed sensing," *IEEE Transactions on Signal Processing*, vol. 65, no. 17, pp. 4538–4550, 2017.
- [37] N. Lu, T. Li, X. Ren, and H. Miao, "A deep learning scheme for motor imagery classification based on restricted Boltzmann machines," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 6, pp. 566–576, 2017.
- [38] L.-W. Kim, "DeepX: deep learning accelerator for restricted Boltzmann machine artificial neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1441–1453, 2018.
- [39] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proceedings of the 24th international conference on Machine learning - ICML '07*, pp. 791–798, Corvalis, Oregon, USA, June 2007.
- [40] M. Jamil and X. S. Yang, "A literature survey of benchmark functions for global optimisation problems," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, no. 2, pp. 150–194, 2013.
- [41] K. E. Parsopoulos and M. N. Vrahatis, "Recent approaches to global optimization problems through particle swarm optimization," *Natural Computing*, vol. 1, no. 2/3, pp. 235–306, 2002.

