Received: 9 August 2020 | Revised: 6 October 2020 | Accepted: 18 October 2020

DOI: 10.1002/cpe.6074

RESEARCH ARTICLE

WILEY

NMIEDA: Estimation of distribution algorithm based on normalized mutual information

Zhiyi Lin | Qing Su | Guobo Xie

School of Computers, Guangdong University of Technology, Guangzhou, China

Correspondence
Qing Su, School of Computers, Guangdong University of Technology, Guangzhou, China.
Email: 16045411@qq.com

Funding information
the National Natural Science Foundation of China, Grant/Award Number: 618002072; the Natural Science Foundation of Guangdong Province, Grant/Award Number: 2018A030313389; the Science and Technology Planning Project of Guangdong Province, Grant/Award Numbers: 2016B030306004, 2018B030323026; the Science and Technology Planning Project of Guangzhou City, Grant/Award Numbers: 201902020012, 201907010021

Summary

A new estimation of distribution algorithm based on normalized mutual information (NMIEDA) is proposed for overcoming the premature convergence of bivariate estimation of distribution algorithms. NMIEDA first uses normalized mutual information to measure the interaction between two variables and then generate a dependency forest model. Second, based on the concept of sporadic model building and a reward and punishment scheme in Selfish Gene, NMIEDA provides a new updating mechanism that accelerates the convergence speed. Finally, a new sampling mechanism is adopted in NMIEDA to improve the efficiency of sampling, which combines stochastic sampling, the opposition-based learning scheme and the mutation operator. The simulation results on benchmark problems and real-world problems demonstrate that NMIEDA often outperforms several other bivariate algorithms.

KEYWORDS

estimation of distribution algorithm, new sampling mechanism, new updating mechanism, NMIEDA, normalized mutual information

1 | INTRODUCTION

Nowadays, artificial intelligence (AI) is an important branch of computer science whose models are being increasingly utilized in many fields.[1-3] Among AI models, evolutionary algorithms (EAs) have proved to be an effective tool for solving various types of optimization problems. Different evolutionary algorithms (EAs) have been developed, such as particle swarm optimization (PSO),[4,5] pigeon-inspired optimization algorithm (PIO),[6] bat algorithm (BA),[7] firefly algorithm (FA),[8-10] artificial bee colony algorithm (ABC),[11-13] harmony search algorithm (HS),[14] cuckoo search algorithm (CS),[15,16] multiobjective evolutionary algorithms (MOEAs).[17-21] Estimation of distribution algorithms (EDAs)[22-24] is another EA paradigm.

Estimation of distribution algorithms (EDAs) learn a probabilistic model from the promising individuals of the previous generation and use this probabilistic model to generate the new population.[22-24] Recently, EDAs have attracted increasing attention from researchers, and various EDAs have been proposed for solving various types of optimization problems, including function optimization problems,[25-27] uncertain capacitated arc routing problems,[28] many-objective optimization,[29] dynamic optimization problems,[30,31] flowshop scheduling problems,[32,33] fuel-loading pattern optimization,[34] mixed-variable newsvendor problems,[35] and supermarket location problem.[36]

In these algorithms, a set of EDAs based on bivariable dependencies for discrete problems have been presented. Mutual information maximization for input clustering (MIMIC)[23,37] designed by de Bonet et al. employs a chain distribution. Combining optimizers with mutual information trees (COMIT),[23] which was developed by S. Baluja et al., uses a tree distribution as the probabilistic model. The bivariate marginal distribution algorithm (BMDA)[23] developed by M. Pelikan uses a forest distribution to estimate the relationships among the variables. Selfish gene on mutual information and entropy-based clusters (SGMIEC),[38] which was developed by Feng Wang et al., uses the mutual information entropy-based clustering construction to represent the relationships among the variables. These algorithms perform well for various discrete optimization benchmarks.

However, there are two major problems in these current bivariate EDAs. First, the size of the population seriously affects the performance of the bivariate EDAs.[24,39] On one hand, bivariate EDA under small population size is unable to represent the general information of the selected individuals accurately through building a probabilistic model and only the low-quality final solution can be obtained. On the other hand, bivariate EDA under large population size can lead to an increased complexity of evaluating populations, building and sampling probabilistic models.[24] Moreover, it is very time-consuming to update the probabilistic model at each generation. The second problem concerns the sampling method in EDAs. The common sampling method that is used by EDAs to sample new individuals is stochastic sampling. However, stochastic errors in sampling can lead to a loss of population diversity[24] and almost no research has been done on the mechanism for maintaining population diversity in the process of sampling.

To overcome these two problems, a new EDA based on normalized mutual information (NMIEDA) is proposed in this article. In the proposed NMIEDA, we first use normalized mutual information (NMI) to represent the relationships among the variables. Then, a new updating mechanism is produced by sporadic model building and a reward and punishment scheme. Furthermore, a new sampling mechanism is proposed in NMIEDA to maintain the population diversity. Finally, the effects of the new updating mechanism and sampling mechanism of NMIEDA are investigated experimentally. Furthermore, NMIEDA is also compared with other bivariate EDAs on several commonly utilized test problems and real-world Ising Spin Glass problems. Experimental results demonstrate the effectiveness of the new mechanisms and the superior performance of NMIEDA.

The rest of this article is organized as follows: Section 2 details the proposed NMIEDA. Section 3presents the experimental settings and results. Section 4 concludes this article.

2 | NMIEDA

In this section, the details of proposed NMIEDA will be introduced. First, we describe the initialization and updating of the probabilistic model in the following subsections, which are two important components of NMIEDA. Second, we discuss the new sampling mechanism, which can be used in other EDAs. Finally, the main framework of NMIEDA is provided.

2.1 | Initialization of the probabilistic model

Before the initialization of the probabilistic model is presented, we introduce the correlation measurement and the dependency forest model that are used by NMIEDA in Sections 2.1.1 and 2.1.2. After that, the initialization of the probabilistic model is presented in Section 2.1.3.

2.1.1 | Correlation measurement

In this article, we adopt the normalized mutual information (NMI) as the correlation measurement to identify the relationships among the variables. The definitions of mutual information and NMI are as follows:

Definition 1. Given two random discrete variables X and Y, $MI(X, Y)$ is the mutual information of X and Y:

$$MI(X, Y) = H(X) + H(Y) - H(X, Y), \tag{1}$$

where $H(X)$ and $H(Y)$ are the marginal entropies and $H(X, Y)$ is the joint entropy of X and Y. Since there is no upper limit on the mutual information, the correlation between X and Y is difficult to determine.

Definition 2. Normalized mutual information (NMI) is a normalization of the mutual information (MI) score to scale the relations between 0 (no correlation) and 1 (perfect correlation). The $NMI(X, Y)$ (normalized mutual information of X and Y) can be calculated by $H(X)$, $H(Y)$, and $MI(X, Y)$:[40]

$$NMI(X, Y) = 2\frac{MI(X, Y)}{H(X) + H(Y)}, \tag{2}$$

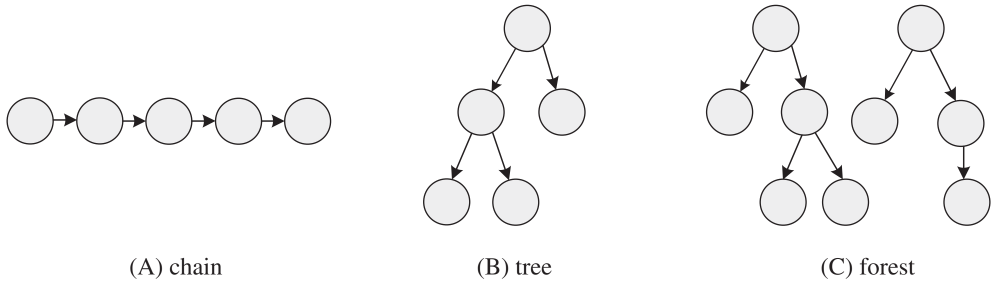
where $NMI(X, Y) \in [0, 1]$, and a higher value corresponds to a closer relation between variables X and Y.

NMI is a concept whose properties have been studied in several ways and has been used for medical image registration problems.[41] Compared with mutual information, it is easy to compare the correlation between X and Y with normalized mutual information.

2.1.2 | The probabilistic model of NMIEDA

Bivariate EDAs use different probabilistic models to learn variable interactions. MIMIC uses a factorization graph that has the structure of a chain (see Figure 1(A)). COMIT uses dependency trees to model promising solutions (see Figure 1(B)). NMIEDA, BMDA, and SGMIEC use a forest distribution that is based on a set of mutually independent trees (see Figure 1(C)).

FIGURE 1 Probabilistic models for bivariate EDAs


(A) chain

(B) tree

(C) forest

2.1.3 | Initialization of the probabilistic model

NMIEDA maintains a frequency matrix F that contains a number $F[x_i = u, x_j = v]$ for every pair of variables x_i and x_j and every combination of binary assignments to u and v, where $F[x_i = u, x_j = v]$ is an estimate of how many the promising individuals have bit $x_i = u$ and bit $x_j = v$. All values of $F[x_i = u, x_j = v]$ are initialized to a small constant before the first generation of NMIEDA and then updated by the information of the promising individuals. Then, we can calculate the unconditional probability $P(x_i)$ and the conditional probability $P(x_i, x_j)$ by the current values of $F[x_i = u, x_j = v]$. According to $P(x_i)$ and $P(x_i, x_j)$, the entropy $H(x_i)$, the mutual information $MI(x_i, x_j)$ and normalized mutual information $NMI(x_i, x_j)$ can be obtained in the end.

Let us denote by $K = \{x_1, x_2, \ldots, x_n\}$ the set of variables that have not yet been processed, and by $NMI(x_i, x_j)$ the normalized mutual information (NMI) between two variables x_i and x_j. Then, the pseudocode for the construction of a dependency forest model is shown in Algorithm 1.

Algorithm 1. The construction of a dependency forest model

```
Step1: Suppose an array $PR$ indicates the parents for each variable and $G$ indicates the group number of each variable. Let $S = \Phi$, $V = K$ and $g = 1$.
Step2:Randomly choose a variable $x_r$ from $V$, and let $PR(x_r) = 0, G(x_r) = g$, $S = S \cup \{x_r\}$ and $V = V - \{x_r\}$. For each $x_j \in V$ ,calculate $NMI(x_j, x_r)$.
Step3: If there are no more elements in Set $V$, finish. Otherwise, go to Step 4.
Step4: Choose the two variables $(x_i, x_j)$ with the maximum normalized mutual information where $x_i \in V$ , $x_j \in S$ and $G(x_j) = g$.
    If $NMI(x_i, x_j)$ exceeds the specified threshold $TR$, create an edge between the two variables, that is let $PR(x_i) = x_j$, $S = \{x_i\} \cup S$,
        $V = V - \{x_i\}$ , $G(x_i) = g$, and go to step 4.
    Otherwise, let $g = g+1$ and go to Step 2.
```

2.2 | Updating of the probabilistic model

In most EDAs, the probabilistic model is updated with the information of selected promising individuals that belong to the current population at each generation. However, an EDA under a small population size is unable to represent the probabilistic model accurately and cost of updating the probabilistic model at each generation is high. Hence, to avoid this situation, NMIEDA provides a new updating mechanism based on the concept of sporadic model building[42] to solve the first problem of current bivariate EDAs. The new updating mechanism updates the probabilistic model once every m generations (we call m sampling interval), whereas in the remaining generations, the probabilistic model from the previous generation is used and only the frequency matrix F is updated based on the reward and punishment scheme.

2.2.1 | Reward and punishment scheme

The selfish gene theory proposed by Richard Dawkins[43] claims that the basic element of evolution is the gene, rather than the population or individual. The key concept of the theory is that the natural selection acts through the survival of genes, not individuals. Individuals are simply the carriers of genes and the population is viewed as just a storage of genetic material.[43] In the view of this theory, most EDAs only consider the successful genes of the superior individuals, which result in the loss of the information of the other individuals, especially the inferior individuals. So, it is very important for the updating of probabilistic model depends on the frequencies of all genes from both the superior individuals and the inferior individuals. Therefore, we use a reward and punishment scheme to update the frequency matrix F according to the information of the worst individual g_{worst} and the best individual g_{best}. This scheme is presented as Algorithm 2.[38,44]

Algorithm 2. Pseudocode of the reward and punishment scheme

```
reward_punish_genes($g_{best}, g_{worst}$)
for $j = 1$ to $n$ do
    for $k = 1$ to $n$ do
        $F(g_{best}[j], g_{best}[k]) = F(g_{best}[j], g_{best}[k]) + \varepsilon;$
        $F(g_{worst}[j], g_{worst}[k]) = F(g_{worst}[j], g_{worst}[k]) - \varepsilon;$
    end
end
```

Herein, ε is the value of the feedback. Usually, it is set as follows:

$$\varepsilon = \frac{2 * N}{n * (n-1)}, \tag{3}$$

where N is the sample size using stochastic sampling and n is the problem dimension.

This scheme realizes the correction of the current model by adjusting the frequency matrix F, which can avoid the problem that probabilistic models cannot accurately represent the general information of the population when EDA has a small population size. Thus, the scheme improves the optimizing capability and avoids premature convergence to an extent.

Since the processing time of the reward and punishment scheme is much less than that of the probabilistic model building, the new updating mechanism decreases the overall processing time of the model building and leads to a significant speedup. In addition, the new updating mechanism can avoid premature convergence through the use of the reward and punishment scheme to adjust the parameters of the current model.

2.3 | Probabilistic model sampling mechanism

At each generation of NMIEDA, a new population is generated and new individuals of the new population are created by sampling the dependency forest model. To solve the second problem of current bivariate EDAs that is mentioned above, a new sampling mechanism, which can use the location information and the global statistical information of individuals, is proposed in NMIEDA. It incorporates two sampling operators and a mutation operator into NMIEDA for generating new individuals after building the parental distribution model. NMIEDA can benefit from the strengths of the various operators. A flowchart of the new sampling mechanism is presented Figure 2, and the details of the new sampling mechanism will be discussed.

2.3.1 | Stochastic sampling

New individuals are created in NMIEDA by sampling the dependency forest model. Stochastic sampling is performed according to Algorithm 3. The stochastic sampling process is executed for a fixed value of N.

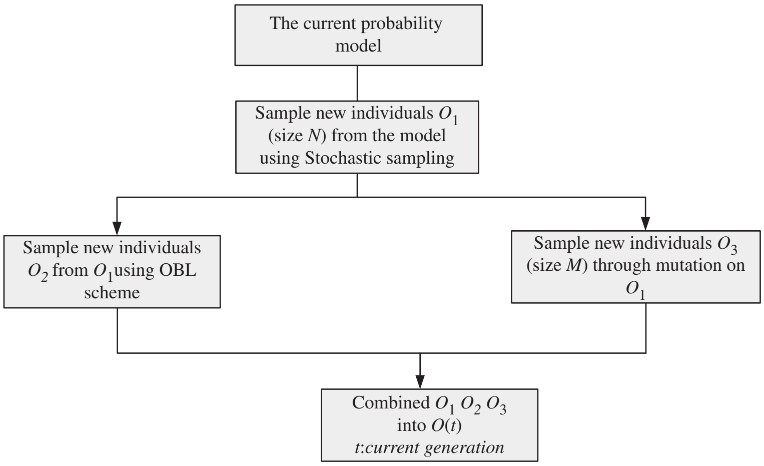

The current probability model
Sample new individuals O_1 (size N) from the model using Stochastic sampling
Sample new individuals O_2 from O_1 using OBL scheme
Sample new individuals O_3 (size M) through mutation on O_1
Combined O_1 O_2 O_3 into $O(t)$ t:current generation

FIGURE 2 Flowchart of the new sampling mechanism

Algorithm 3. Pseudocode of stochastic sampling

```
Step1: Set $K = V = \{x_1, x_2, \dots, x_n\}$
Step2: Choose a variable $x_r$ with $PR(x_r) = 0$ from $K$, generate $x_r$ using univariate frequency $p(x_r)$, set $K = K - \{x_r\}$.
Step3: If $K$ is empty, **finish**.
Step4: If there exists $x_i$ from $K$ that is connected to $x_j$ from $V - K$ in the dependency forest and $PR(x_i) = x_j$, choose $x_i$ from $K$.
   Otherwise, go to Step 2.
Step5: Generate $x_i$ using conditional probability $p(x_i|x_j)$.
Step6: Remove $x_i$ from $K$, set $K = K - \{xi\}$, and go to Step 4.
```

2.3.2 | Opposition-based learning (OBL)

Opposition-based learning (OBL) proposed by Thamid R. Tizhoosh[45] has been utilized in various soft computing areas.[46] According to OBL, it is easier to find a candidate solution closer to the global optimum by evaluating a solution x and its corresponding opposite solution x_o. The formula for calculating the opposite solution x_o is as follows:[46]

$$x_o = opposition(x) = \begin{cases} 1 - x_i & \text{if } x_i \in \{0, 1\} \\ a_i + b_i - x_i & \text{if } x_i \in \text{ R and } x_i \in [a_i \ b_i] \end{cases} \quad \forall i \in \{1, 2, \dots, n\}. \tag{4}$$

After sampling new individuals from the probability model, NMIEDA uses the OBL scheme to generate a solution for each individual that was obtained by stochastic sampling. In this way, it can cover the search space more effectively and accelerate the convergence of the optimization process.

2.3.3 | Mutation operator

Mutation can add new gene to the individual by altering one or more gene values from its initial value. Applying the mutation operation on a selected individual change the values of genes by mutation rate P_m according to the following formula:

$$mutation(x_i) = \begin{cases} 1 - x_i & \text{if } x_i \in \{0, 1\} \\ \text{rand}(a_i, b_i) & \text{if } x_i \in R \text{ and } x_i \in [a_i \ b_i] \end{cases}, \tag{5}$$

where x_i is a randomly chosen gene and $rand(a_i, b_i)$ is a random number between a_i and b_i that satisfies uniform distribution. At the early stage of the search, the diversity of the population is high. Therefore, the mutation rate P_m and the sample size M using this method must have small values to avoid slowing the convergence. Additionally, at the later stage of the search, the population diversity is low. To increase the population diversity, the mutation rate P_m and the sample size M must be large. Based on this discussion, dynamic parameter adaptive adjustments of P_m and M are proposed as follows:

$$P_m = 0.5 \times \frac{t}{t_{\max}}, \tag{6}$$

$$M = \frac{t}{5 * t_{\max}} N, \tag{7}$$

where t is the number of current generation, t_{max} is the maximum generation, and N denotes the sample size using stochastic sampling. According to Equations (6) and (7), this strategy can adjust P_m and M adaptively according to the searching process.

At each generation, new individuals are generated by stochastic sampling, opposition-based learning and the mutation operator. Thus, there always exist more than N new individuals that have been generated by only stochastic sampling, the purpose of which is to maintain the diversity of the new population at all times and improve the convergence reliability of NMIEDA.

2.4 Calibration of the proposed NIMIEDA

Considering all previous subsections, the NMIEDA is developed and presented in Algorithm 4.

Algorithm 4. Pseudocode of NMIEDA

```
Step1: Set the NMIEDA's parameters: the frequency matrix $F$, the sample size using stochastic sampling $N$, the sampling interval $m$, the threshold $TR$, etc.
    Let the current generation $t = 0$.
Step2: Generate a random population $P(t)$ and evaluate each individual.
Step3: Select a set of promising individuals $Q(t)$ from $P(t)$.
Step4: If $t\%m = 0$, calculate the frequency matrix $F$ according to $Q(t)$ and construct a new probabilistic model $G$ by Algorithm 1; otherwise, update the
    frequency matrix $F$ by Algorithm 2.
Step5: Sample new individuals $O(t)$ from $G$ using the new sampling mechanism.
Step6: Replace $P(t)$ with the selected $N$ best individuals from $O(t)$ and then select a set of promising individuals $Q(t)$ from $P(t)$.
Step7: If the stopping criterion is met, stop; otherwise, set the current generation $t = t + 1$ and go to step 4.
```

3 EXPERIMENTS

This section collects all the information about the experiments that are carried out to validate our proposed algorithm. First, we describe the set of test functions that are used. Then, we test the influence of the new updating mechanism and the new sampling mechanism. Third, the comparisons of NMIEDA with four bivariate EDAs are presented. Finally, two different sizes of two-dimensional Ising Spin Glass (ISG) problems are tested for demonstrating the feasibility of NMIEDA in solving real-world problems.

3.1 Benchmark functions

A test suite with five additively decomposable functions and one overlapping function is used in our experimental study. The functions are defined as follows.

1. **OneMax funciton**

$$f_{one}(x) = \max\left(\sum_{i=1}^{n} x_i\right) \quad x_i \in \{0, 1\}, \tag{8}$$

where x_i represents the ith bit of the input string, which has length n. The OneMax function is a decomposable function of order 1 and attains its optimum value on the string of all ones.

2. **Quadratic function**

$$f_{Quad}(x) = \max\left(\sum_{i=1}^{n} F(x_{2i-1}, x_{2i})\right) \quad x_{2i-1}, x_{2i} \in \{0, 1\}, \tag{9}$$

where $F(s, t) = 0.9 - 0.9(s + t) + 1.9st$.

The Quadratic function is a decomposable function of order 2 and the dependencies do not form cycles. The Quadratic function is attained its optimum value on the string of all ones.

3. **Deceptive function of order 3 (DEC-3)**

$$f_{Dec}(x) = \max\left(\sum_{i=1}^{n/3} F(x_i, x_{i+1}, x_{i+2})\right) \quad x_i \in \{0, 1\}, \tag{10}$$

where $F(r,s,t) = \begin{cases} 1 & \text{if } r+s+t=3 \\ 0 & \text{if } r+s+t=2 \\ 0.8 & \text{if } r+s+t=1 \\ 0.9 & \text{if } r+s+t=0 \end{cases}$ DEC-3 is a nonoverlapping, additively decomposable composition of order 3. This problem has many locally optimal solutions.

4. **Trap function of order 5 (Trap-5)**

$$f_{Trap}(x) = \max\left(\sum_{i=1}^{n/5} F(x_{5i-4} + x_{5i-3} + x_{5i-2} + x_{5i-1} + x_{5i})\right) \quad x_i \in \{0,1\}, \tag{11}$$

where $s = x_{5i-4} + x_{5i-3} + x_{5i-2} + x_{5i-1} + x_{5i}$ and $F(s) = \begin{cases} 5-s & \text{if } s<5 \\ 5 & \text{otherwise} \end{cases}$.

Trap-5 is a nonoverlapping deceptive function and characterized by multivariate correlation. An n-bit Trap-5 has $(2^{n/5} - 1)$ local optima and one global optimum.

5. **Satisfaction function**

$$f_{Satfp}(x) = \max\left(\sum_{i=1}^{n} F(x_{5i-4}, x_{5i-3}, x_{5i-2}, x_{5i-1}, x_{5i})\right) \quad x_i \in \{0,1\}, \tag{12}$$

where $f(x_{5i-4}, x_{5i-3}, x_{5i-2}, x_{5i-1}, x_{5i}) = \begin{cases} 5 & \text{if all variables equal to 1} \\ 0 & \text{otherwise} \end{cases}$. The function is a decomposable function of order 5. The optimum solution of the satisfaction function is attained on the string with ones in all positions.

6. **Four Peaks**

$$f_{four}(x) = \max\left[tail(0,x), head(1,x)\right] + R(x), \tag{13}$$

where $tail(0,x)$ =number of trailing 0's in the vector x, $head(1,x)$ =number of leading 1's in the vector x and

$$R(x) = \begin{cases} D & \text{if tail}(0,x) > 0.1 \times D \wedge \text{ head}(1,x) > 0.1 \times D \\ 0 & \end{cases}, \tag{14}$$

where D is the problem size. The relationship between the variables of Four Peaks is the chain formation and the two ends of the chain mutually overlap.

Various dimensions are tested for the six functions: $n = 10$ to 200 with a step size of 10 for five functions, and $n = 9$ to 210 with a step size of 18 for the DEC-3 function.

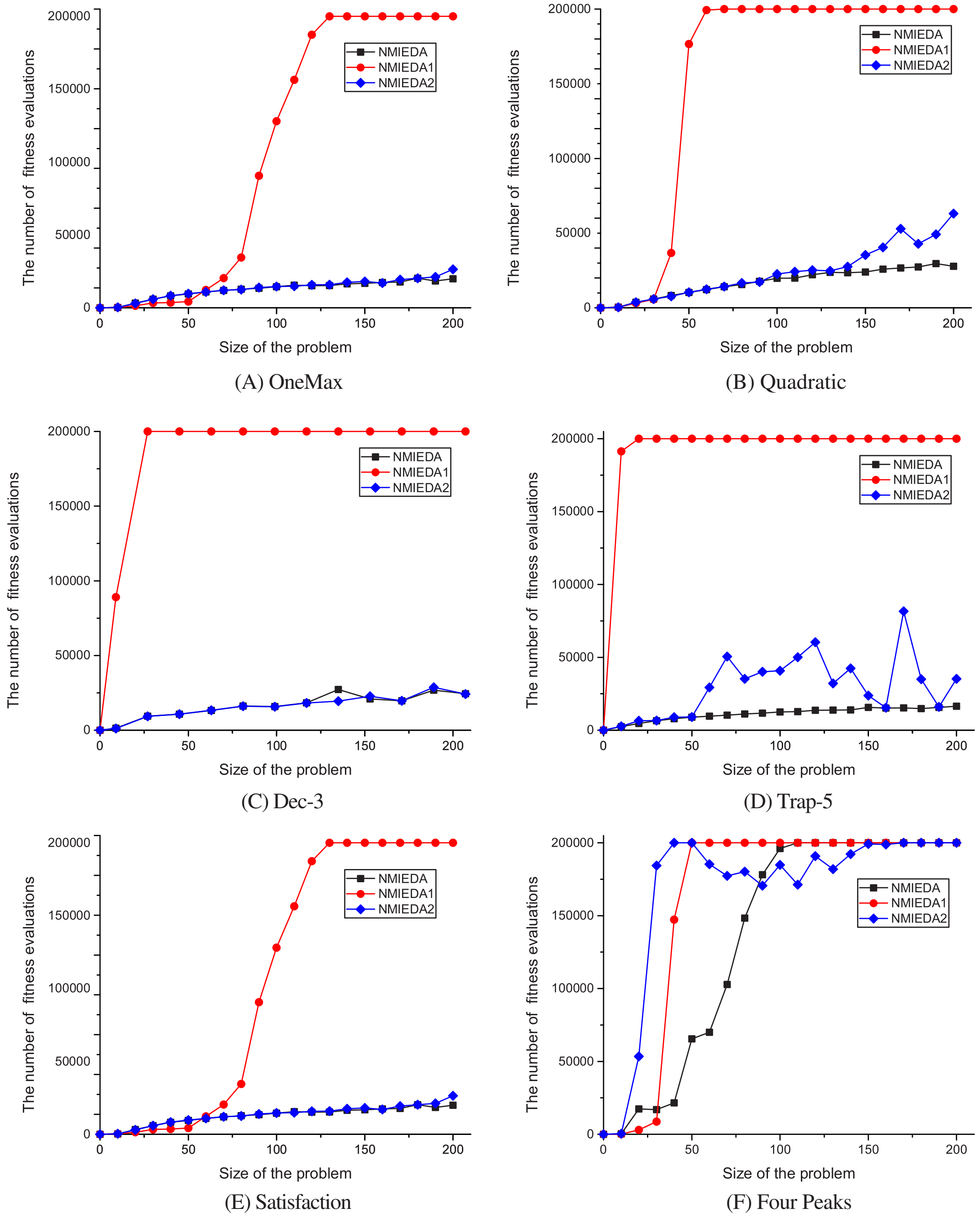
3.2 | Influence of the new updating mechanism and the new sampling mechanism

First, we study the influence of the new updating mechanism and the new sampling mechanism in NMIEDA. For the proposed NMIEDA, we set the number of samples that are generated by stochastic sampling to $N = 20$, the sampling interval to $m = 5$ and the threshold to $TR = 0.9$. Denote by NMIEDA1 the NMIEDA without the new updating mechanism (set the sampling interval $m = 1$) and by NMIEDA2 the NMIEDA without the new sampling mechanism (use stochastic sampling only). Two measures are adopted to verify the performance of three algorithms:

1. Convergence velocity: The convergence velocity of an algorithm can be calculated by the average number of fitness evaluations that are needed to reach the globally optimal solution.
2. Convergence reliability: The convergence reliability of an algorithm under various numbers of variables can be measured by the difference between the globally optimal solution and the average result of converged solutions.

The statistical results of the convergence velocities that are obtained by NMIEDA, NMIEDA1, and NMIEDA2 over 20 independent runs on six problems are presented in Figure 3.

According to Figure 3, NMIEDA and NMIEDA2 often require far fewer fitness evaluations to converge in solving the six tested problems, except for OneMax, compared with NMIEDA1. For example, NMIEDA1 cannot get the global optimal solution if the size of the Trap-5 problem is larger than



(A) OneMax

(B) Quadratic

(C) Dec-3

(D) Trap-5

(E) Satisfaction

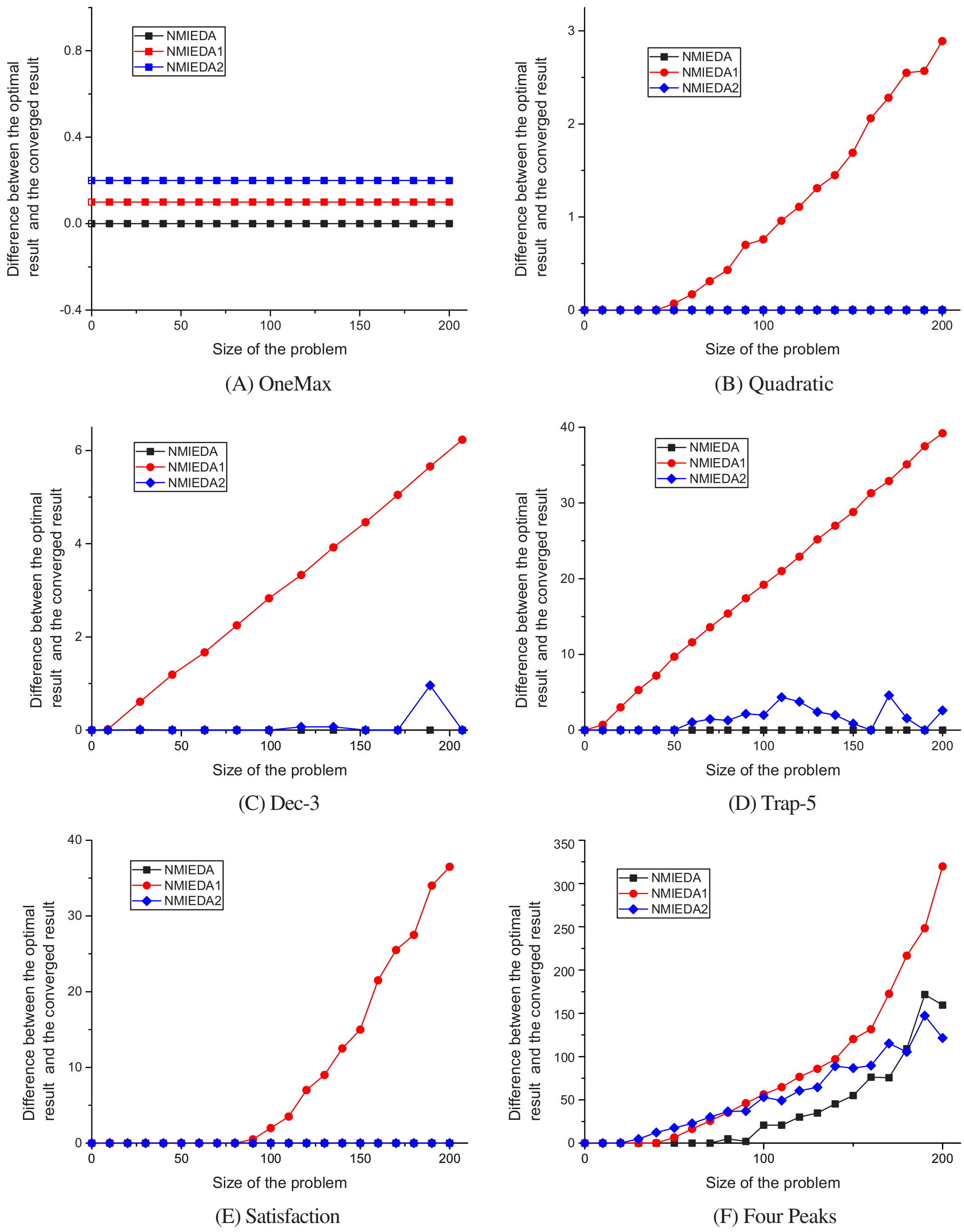
(F) Four Peaks

FIGURE 3 Convergence velocities of three algorithms

20 and the size of the Dec-3 problem is larger than 27. From Figure 3, NMIEDA and NMIEDA2 are consistent with each other on OneMax, Dec-3, and Satisfaction. However, NMIEDA outperforms NMIEDA2 on Quadratic, Trap-5, and Four Peaks. It is evident from Figure 3that NMIEDA1 shows large fluctuations in solving the Trap-5 and Four Peaks problems. For example, NMIEDA1 can obtain the global optimum of the 160-dimensional Trap-5 problem in only 15,423 fitness evaluations, and 81,571 fitness evaluations for the 170-dimensional Trap-5 problem.

The convergence reliability results are shown in Figure 4. For convenience of comparison, the curve of NMIEDA1 in Figure 4(A) is moved vertically by +0.1 and the curve of NMIEDA2 in Figure 4(A) is moved vertically by +0.2. According to Figure 4, NMIEDA and NMIEDA2 are able to identify high-quality solutions and outperform NMIEDA1 significantly on five problems. Moreover, NMIEDA outperforms NMIEDA2 for high-dimensional Trap-5, Four Peaks, and Dec-3 problems.

According to the abovementioned analytical results, we conclude that the new updating mechanism plays a significant role in the performance of NMIEDA for higher-order functions, which improves the optimization capability of NMIEDA and avoids premature convergence. We also conclude that the new sampling mechanism significantly contributes to the convergence reliability of NMIEDA in solving high-dimensional optimization problems.



(A) OneMax

(B) Quadratic

(C) Dec-3

(D) Trap-5

(E) Satisfaction

(F) Four Peaks

FIGURE 4 Convergence reliabilities of three algorithms

3.3 | Comparisons of NMIEDA with four bivariate EDAs

Next, the optimization results of NMIEDA are compared against four other bivariate EDAs: BMDA, MIMIC, COMIT, and SGMIEC. All algorithms are implemented in VC++6.0 on the same PC.

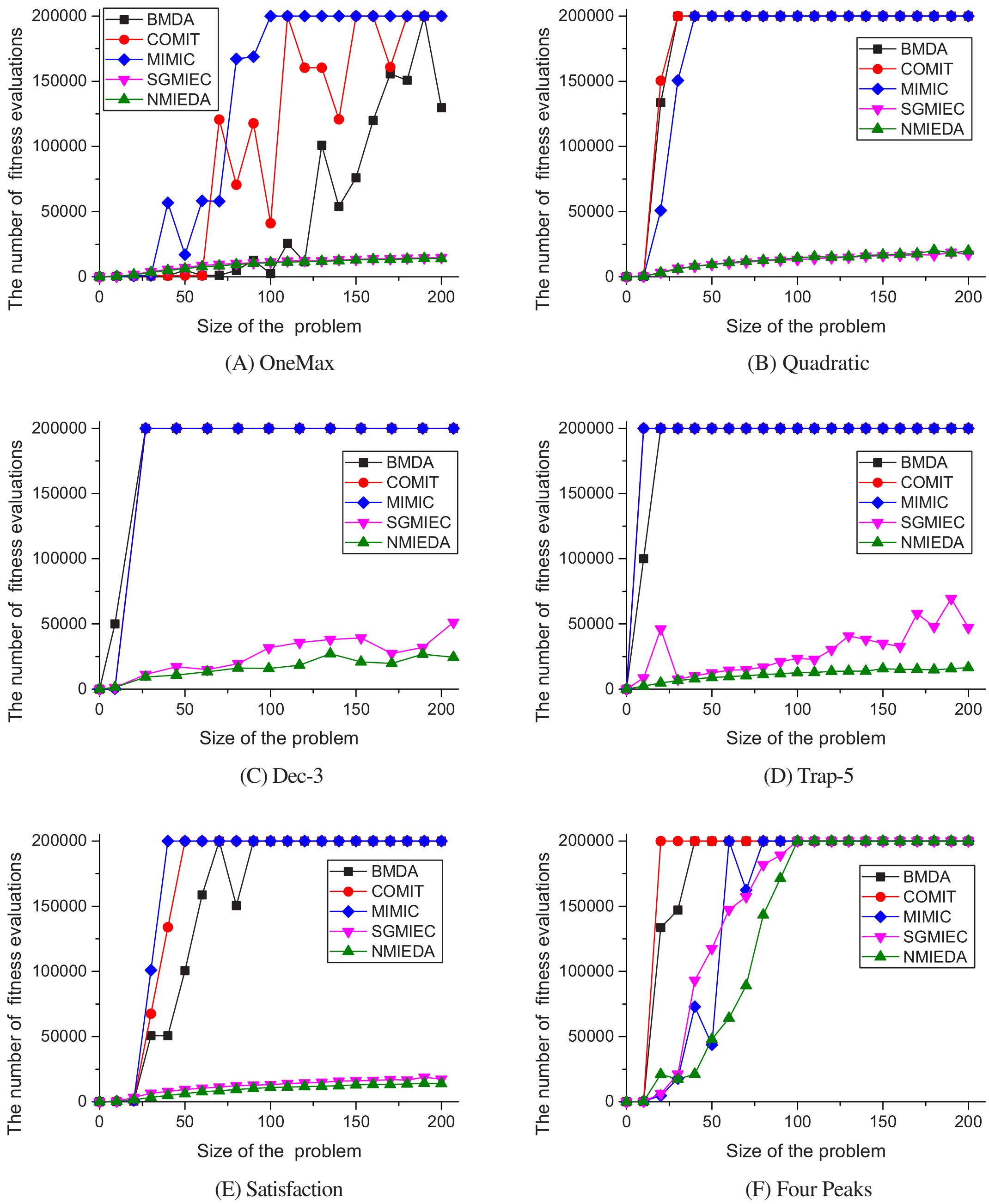
In our experiments, the parameter settings of the five algorithms are held constant and set as follows: The sample size at each generation is fixed to 200 and a truncation selection percentage of 50% is used for COMIT, BMDA, and MIMIC. For comparison, the optimal individual of MIMIC is reserved per generation. For SGMIEC, the number of samples $N = 7$, the feedback value $\varepsilon = 0.01 \times N$ and the learning factor $\alpha = 0.1$. For the proposed NMIEDA, we set the number of samples that are generated by stochastic sampling to $N = 20$, the sampling interval to $m = 5$ and the threshold to $TR = 0.9$.

For each algorithm-function-dimension combination, 20 independent runs are performed. For a fair comparison, all algorithms terminate when the maximum number of fitness evaluations (200,000) is reached. Meanwhile, the number of fitness evaluations that are needed to attain the global optimum is recorded. Three measures are adopted to compare the performance of five algorithms, including the two measures that are described in Section3.2. Another measure is described as follows:

1. Convergence process: The convergence process of an algorithm under a fixed number of variables can be described as the average fitness along with the number of evaluations for the algorithm.

3.3.1 | Convergence velocity

First, the convergence velocities of BMDA, COMIT, MIMIC, SGMIEC, and NMIEDA are tested and compared. Figure 5 shows the experimental results. According to the results in Figure 5, NMIEDA often requires far fewer fitness evaluations to converge in solving the six tested problems



(A) OneMax

(B) Quadratic

(C) Dec-3

(D) Trap-5

(E) Satisfaction

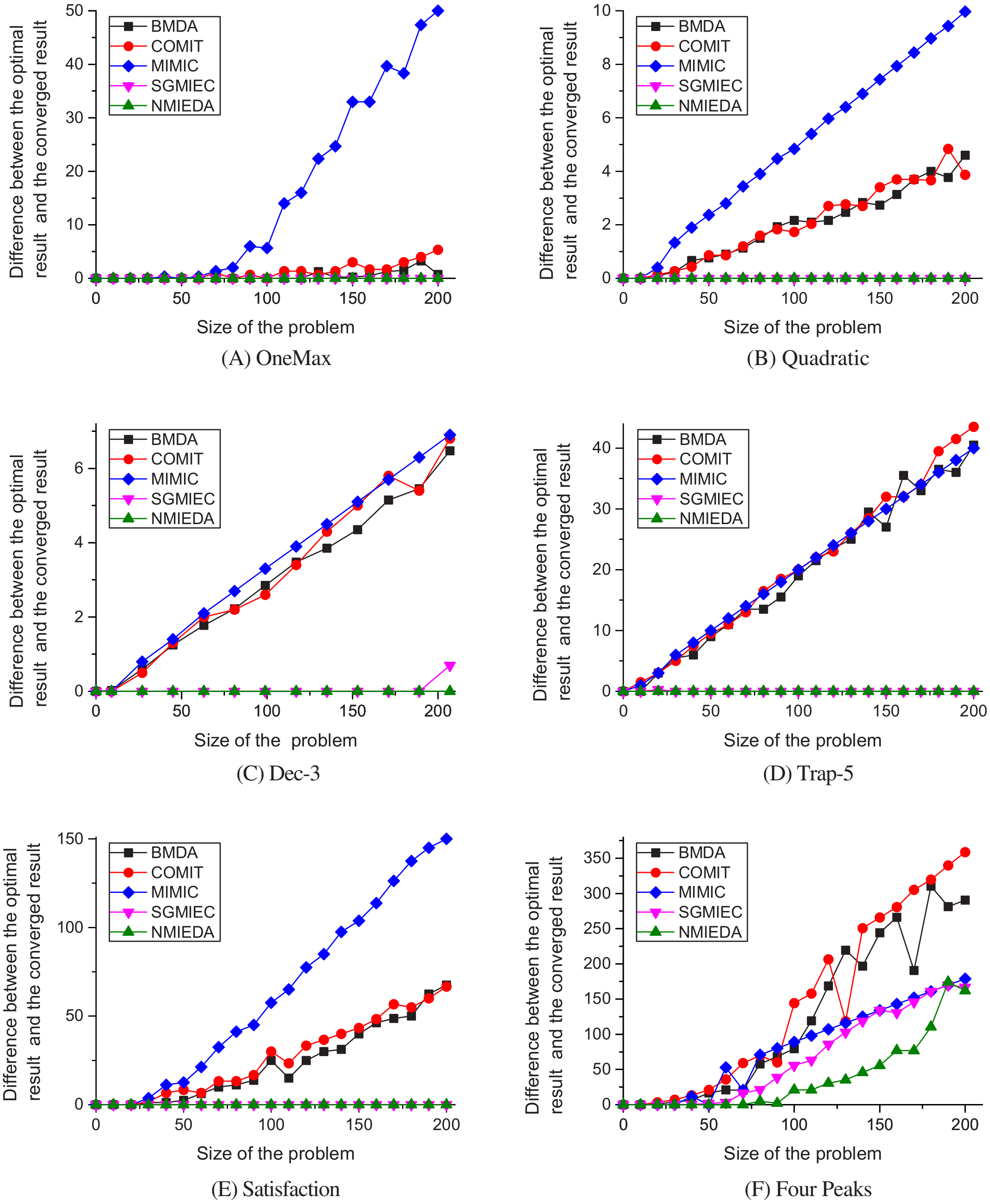
(F) Four Peaks

FIGURE 5 Convergence velocities of five bivariate EDAs

when compared with COMIT, BMDA, and MIMIC, especially in high dimensions. For example, NMIEDA can find the global optimum for the Dec-3 problem in less than 30,000 fitness evaluations until problem size reaches 207, while COMIT, BMDA, and MIMIC cannot obtain the global optimum if the problem size is larger than 27. SGMIEC requires a similar number of evaluations compared to NMIEDA. However, it is obvious from the results that SGMIEC's performance is influenced considerably more than that of NMIEDA by increases in the number of variables in solving Dec-3, Trap-5, and Four Peaks. For example, NMIEDA can find the best solution in the 60-dimensional Trap-5 problem in only approximately 9580 fitness evaluations and in 16,440 fitness evaluations for the 200-dimensional Trap-5 problem.

3.3.2 | Convergence reliability

Second, the convergence reliabilities of BMDA, COMIT, MIMIC, SGMIEC, and NMIEDA are tested and compared. Figure 6 shows the experimental results. According to Figure 6, the performances of BMDA, COMIT, and MIMIC are not satisfactory on any of the six tested problems. For example, the difference between the globally optimal value and the average value of converged solutions for the Four Peaks of COMIT is as large as 300



(A) OneMax

(B) Quadratic

(C) Dec-3

(D) Trap-5

(E) Satisfaction

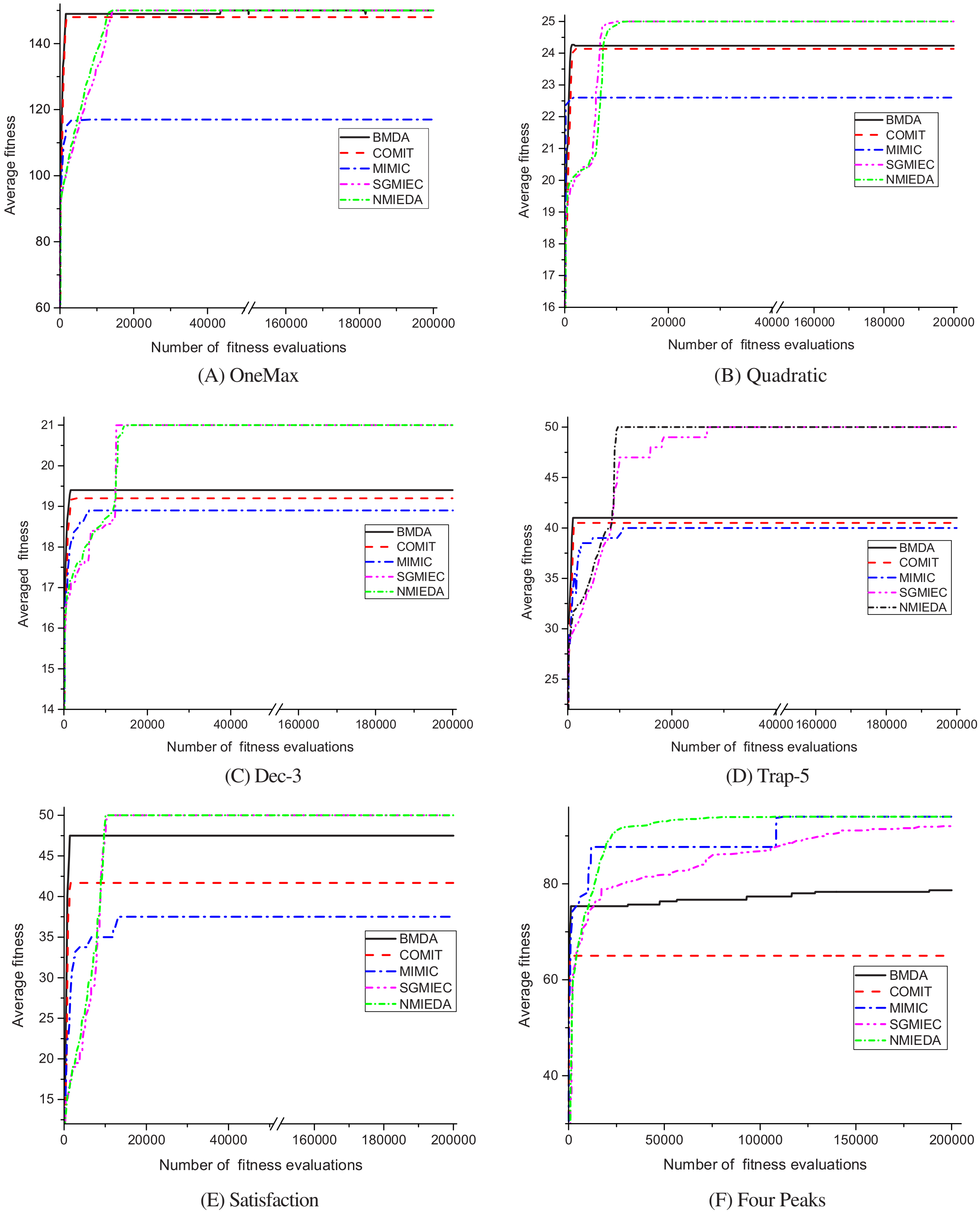
(F) Four Peaks

FIGURE 6 Convergence reliabilities of five bivariate EDAs

when the problem size increases to 200. Furthermore, the curves of COMIT, BMDA, and MIMIC show large fluctuations, which implies poor stability. From Figure 6, NMIEDA and SGMIEC can identify high-quality solutions and significantly outperform BMDA, COMIT, and MIMIC on all tested six problems. However, it is clear that NMIEDA obtains better performance than SGMIEC for Trap-5 and Four Peaks. From Figures 5 and 6, NMIEDA achieves the best mean solution quality with much faster speed than BMDA, COMIT, MIMIC, and SGMIEC.

3.3.3 | Convergence process

Finally, the convergence processes of BMDA, COMIT, MIMIC, SGMIEC, and NMIEDA are tested and compared. Figure 7 shows the experimental results. Although the convergence speeds of NMIEDA and SGMIEC are slower than those of other three algorithms in the early stage, they avoid



(A) OneMax

(B) Quadratic

(C) Dec-3

(D) Trap-5

(E) Satisfaction

(F) Four Peaks

FIGURE 7 Convergence processes of five bivariate EDAs

local optimization by successfully maintaining a balance between population diversity and fast convergence. The population of BMDA, COMIT, and MIMIC becomes increasingly homogeneous and the lack of diversity leads to BMDA, COMIT, and MIMIC fall into local optimization after a quick convergence.[38] We also see from Figure 7 that NMIEDA converges slightly faster than SGMIEC, except on the Quadratic and Dec-3 problems. The convergence process of NMIEDA is fast and stable.

3.4 | Application of NMIEDA

Ising Spin Glass (ISG) problem is a widely-known problem of statistical physics.[47] Let us suppose $\sigma = \{\sigma_1, \sigma_2, \dots, \sigma_n\}$ indicates a set of spin variables and a real number J_{ij} represents a coupling coefficient from pairs of spins σ_i and σ_j. Then, the ISG problem can be defined by an energy function H.

$$H(\sigma) = -\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} J_{ij}\sigma_i\sigma_j, \tag{15}$$

where each spin σ_i can obtain a value from $\{+1, -1\}$. For convenience, we reformulate the ISG problem as a maximization problem.

$$-H(\sigma) = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} J_{ij}\sigma_i\sigma_j. \tag{16}$$

Two different sizes of two dimensional ISG problems are performed in this experiment. They are $8\times 8(n=64)$, $10\times 10(n=100)$. We ran 20 independent runs of BMDA, COMIT, MIMIC, SGMIEC, and NMIEDA for each of the eight instances. Table1 exhibits the statistical comparison of BMDA, COMIT, MIMIC, SGMIEC, and NMIEDA on all eight instances of the ISG problems. The numbers of Table1 in bold denote the best value of all algorithms.

From Table1, the performance of NMIEDA is better than that of BMDA, COMIT, MIMIC, and SGMIEC. For example, NMIEDA can find the global optimum for the ISG problems with 64 spins and can obtain the best solutions than the other four algorithms for the ISG problems with 100 spins. Therefore, it is proved that NMIEDA outperforms BMDA, COMIT, MIMIC, SGMIEC in finding the ground states of ISG problems.

It is concluded that NMIEDA obtains better overall results on six benchmark functions and ISG problems than the other four algorithms.

TABLE 1 Statistical comparison of the algorithms for the ISG problems

Algorithm	Results	I-64-1	I-64-2	I-64-3	I-64-4	I-100-1	I-100-2	I-100-3	I-100-4
NMIEDA	Best	**84**	**86**	**88**	**94**	**142**	**132**	**138**	**138**
	Mean	**84**	**86**	**88**	**94**	**138.4**	**131.2**	**138**	**134**
	Deviation	**0**	**0**	**0**	**0**	**2.27**	**1.69**	**0**	**1.88**
BMDA	Best	80	82	80	90	122	116	126	130
	Mean	73.6	75.2	72.4	78	112.66	108.88	117.56	114.44
	Deviation	4.3	7.07	5.15	8.22	6.63	6.25	6.46	7.86
COMIT	Best	72	70	68	78	106	112	114	114
	Mean	57	62.67	60	63.3	100.18	97.45	97.3	104.8
	Deviation	8.38	5.61	5.91	8.41	3.73	7	9.09	7.78
MIMIC	Best	80	82	76	82	110	108	102	106
	Mean	71.82	70.73	72	75.45	100.66	89.6	95.2	94.8
	Deviation	4.94	6.4	3.1	5.45	5.65	8.26	6.54	8.6
SGMIEC	Best	**84**	**86**	**88**	**94**	138	128	**138**	130
	Mean	78.28	83.09	82.8	86.3	129.6	122.4	128	121.6
	Deviation	4.3	2.81	4.12	5.77	4.79	4.69	5.42	6.09

4 | CONCLUSIONS

This article has presented a forest model with NMI measure capable of finding better solutions than other four bivariate EDAs, which was called estimation of distribution algorithm based on normalized mutual information (NMIEDA).

The NMIEDA uses two techniques that are not commonly adopted in other bivariate EDAs, One is the updating mechanism, the other is the hybrid sampling mechanism.

In the literature, most bivariate EDAs build probabilistic model at each generation. In NMIEDA, the structure of NMIEDA's probabilistic model is built once every m (sampling interval) generations, which reduces the computational cost of NMIEDA. Moreover, the frequency matrix *F* is updated by a reward and punishment scheme in the remaining generations, which increases the reliability of probabilistic model during the algorithm's execution.

It is noted that new individuals are created by probabilistic models in bivariate EDAs with single sampling method and without the location information from the previous population. In NMIEDA, we combine the advantages of three sampling methods for generating better offspring individuals to use individual location information and population distribution information.

Experiments on six benchmark functions for a variety of problem sizes have demonstrated that the new updating mechanism and the new sampling mechanism significantly contribute to the performance of NMIEDA. We also compared NMIEDA with BMDA, COMIT, MIMIC, and SGMIEC on six benchmark problems as well as real-world Ising Spin Glass problems for a variety of problem sizes. The experimental results show that NMIEDA significantly better performance than those obtained by BMDA, COMIT, and MIMIC. The experimental results also indicate that NMIEDA slightly outperforms SGMIEC for larger dimensions on most functions and ISG problems.

Despite its good performance, NMIEDA may be difficult to choose the appropriate parameters and apply it for other optimization problems directly. Therefore, our future work will focus on further investigating the effects of the NMIEDA's parameters and applying NMIEDA to other types of optimization problems.

ACKNOWLEDGMENT

This work has been supported by the National Natural Science Foundation of China (618002072), the Natural Science Foundation of Guangdong Province (2018A030313389), the Science and Technology Planning Project of Guangzhou City (201902020012, 201907010021), and the Science and Technology Planning Project of Guangdong Province (2016B030306004, 2018B030323026).

ORCID

Zhiyi Lin https://orcid.org/0000-0002-3464-3472

REFERENCES

1. Haenlein M, Kaplan A. A brief history of artificial intelligence: on the past, present, and future of artificial intelligence. *Calif Manag Rev*. 2019;61(4):5-14. http://dx.doi.org/10.1177/0008125619864925.
2. Hassan MU, Rehmani MH, Chen J. Differential privacy techniques for cyber physical systems: a survey. *IEEE Commun Surv Tutor*. 2020;22(1):746-789. http://dx.doi.org/10.1109/comst.2019.2944748.
3. Cui Z, Xue F, Zhang S, et al. A hybrid blockchain-based identity authentication scheme for multi-WSN. *IEEE Trans Serv Comput*. 2020;13(2):241-251. http://dx.doi.org/10.1109/tsc.2020.2964537.
4. Cui Z, Zhang J, Wu D, et al. Hybrid many-objective particle swarm optimization algorithm for green coal production problem. *Inf Sci*. 2020;518:256-271. http://dx.doi.org/10.1016/j.ins.2020.01.018.
5. Wang F, Zhang H, Li K, Lin Z, Yang J, Shen X-L. A hybrid particle swarm optimization algorithm using adaptive learning strategy. *Inf Sci*. 2018;436-437:162-177. http://dx.doi.org/10.1016/j.ins.2018.01.027.
6. Cui Z, Zhang J, Wang Y, et al. A pigeon-inspired optimization algorithm for many-objective optimization problems. *Sci China(Inf Sci)*. 2019;62(7):131-138. http://dx.doi.org/10.1007/s11432-018-9729-5.
7. Cui Z, Cao Y, Cai X, Cai J, Chen J. Optimal LEACH protocol with modified bat algorithm for big data sensing systems in Internet of Things. *J Parall Distrib Comput*. 2019;132:217-229. http://dx.doi.org/10.1016/j.jpdc.2017.12.014.
8. Wang H, Wang W, Zhou X, et al. Firefly algorithm with neighborhood attraction. *Inf Sci*. 2017;382-383:374-387. http://dx.doi.org/10.1016/j.ins.2016.12.024.
9. Wang H, Wang W, Cui L, et al. A hybrid multi-objective firefly algorithm for big data optimization. *Appl Soft Comput*. 2018;69:806-815. http://dx.doi.org/10.1016/j.asoc.2017.06.029.
10. Wang H, Wang W, Cui Z, Zhou X, Zhao J, Li Y. A new dynamic firefly algorithm for demand estimation of water resources. *Inf Sci*. 2018;438:95-106. http://dx.doi.org/10.1016/j.ins.2018.01.041.
11. Wang H, Wang W, Xiao S, Cui Z, Xu M, Zhou X. Improving artificial Bee colony algorithm using a new neighborhood selection mechanism. *Inf Sci*. 2020;527:227-240. http://dx.doi.org/10.1016/j.ins.2020.03.064.
12. Hui W, Wang W, Zhou X, et al. Artificial bee colony algorithm based on knowledge fusion. *Complex Intell Syst*. 2020. http://dx.doi.org/10.1007/s40747-020-00171-2.
13. Agarwal P, Mehta S. ABC_DE_FP: a novel hybrid algorithm for complex continuous optimization problems. *Int J Bio-Inspir Comput*. 2019;14(1):46-61. https://doi.org/10.1504/IJBIC.2019.101176.

14. Tawhid MA, Ali AF. Multidirectional harmony search algorithm for solving integer programming and minimax problems. *Int J Bio-Inspir Comput*. 2019;13(3):141-158. http://dx.doi.org/10.1504/ijbic.2019.099179.
15. Tang H, Xue F. Cuckoo search algorithm with different distribution strategy. *Int J Bio-Inspir Comput*. 2019;13(4):234-241. http://dx.doi.org/10.1504/ijbic.2019.100150.
16. Cai X, Niu Y, Geng S, et al. An under-sampled software defect prediction method based on hybrid multi-objective cuckoo search. *Concurr Comput Pract Exper*. 2020;32(5):e5478. http://dx.doi.org/10.1002/cpe.5478.
17. Wang F, Zhang H, Li Y, Zhao Y, Rao Q. External archive matching strategy for MOEA/D. *Soft Comput*. 2018;22(23):7833-7846. http://dx.doi.org/10.1007/s00500-018-3499-9.
18. Wang F, Li Y, Zhang H, Hu T, Shen X-L. An adaptive weight vector guided evolutionary algorithm for preference-based multi-objective optimization. *Swarm Evolut Comput*. 2019;49:220-233. http://dx.doi.org/10.1016/j.swevo.2019.06.009.
19. Cai X, Hu Z, Chen J. A many-objective optimization recommendation algorithm based on knowledge mining. *Inf Sci*. 2020;537:148-161. http://dx.doi.org/10.1016/j.ins.2020.05.067.
20. Cai X, Hu Z, Zhao P, Zhang WS, Chen J. A hybrid recommendation system with many-objective evolutionary algorithm. *Expert Syst Appl*. 2020;159:113648. http://dx.doi.org/10.1016/j.eswa.2020.113648.
21. Wang F, Li Y, Liao F, Yan H. An ensemble learning based prediction strategy for dynamic multi-objective optimization. *Appl Soft Comput*. 2020;96:106592. http://dx.doi.org/10.1016/j.asoc.2020.106592.
22. Lozano JA, Larrañaga P, Inza I, Bengoetxea E. *Towards a New Evolutionary Computation. Advances on Estimation of Distribution Algorithms. Stud Fuzziness & Soft Computing*. New York,NY: Springer-Verlag; 2006.
23. Pelikan M, Hauschild MW, Lobo FG. Estimation of distribution algorithms. *Springer Handbook of Computational Intelligence*. New York,NY: Springer; 2015:899-928.
24. Hauschild M, Pelikan M. An introduction and survey of estimation of distribution algorithms. *Swarm Evolut Comput*. 2011;1(3):111-128. http://dx.doi.org/10.1016/j.swevo.2011.08.003.
25. Zhou A, Sun J, Zhang Q. An estimation of distribution algorithm with cheap and expensive local search methods. *IEEE Trans Evol Comput*. 2015;19(6):807-822. http://dx.doi.org/10.1109/tevc.2014.2387433.
26. PourMohammadBagher L, Ebadzadeh MM, Safabakhsh R. Graphical model based continuous estimation of distribution algorithm. *Appl Soft Comput*. 2017;58:388-400. http://dx.doi.org/10.1016/j.asoc.2017.04.066.
27. Liang Y, Ren Z, Yao X, Feng Z, Chen A, Guo W. Enhancing Gaussian estimation of distribution algorithm by exploiting evolution direction with archive. *IEEE Trans Cybern*. 2020;50(1):140-152. http://dx.doi.org/10.1109/tcyb.2018.2869567.
28. Wang J, Tang K, Lozano JA, Yao X. Estimation of the distribution algorithm with a stochastic local search for uncertain capacitated arc routing problems. *IEEE Trans Evol Comput*. 2016;20(1):96-109. http://dx.doi.org/10.1109/tevc.2015.2428616.
29. Sun Y, Yen GG, Yi Z. Reference line-based estimation of distribution algorithm for many-objective optimization. *Knowl-Based Syst*. 2017;132:129-143. http://dx.doi.org/10.1016/j.knosys.2017.06.021.
30. Jiang M, Qiu L, Huang Z, Yen GG. Dynamic multi-objective estimation of distribution algorithm based on domain adaptation and nonparametric estimation. *Inf Sci*. 2018;435:203-223. http://dx.doi.org/10.1016/j.ins.2017.12.058.
31. Wang Z, Gong M. Dynamic deployment optimization of near space communication system using a novel estimation of distribution algorithm. *Appl Soft Comput*. 2019;78:569-582. http://dx.doi.org/10.1016/j.asoc.2019.02.045.
32. Shao Z, Pi D, Shao W. Estimation of distribution algorithm with path relinking for the blocking flow-shop scheduling problem. *Eng Optim*. 2018;50(5):894-916. http://dx.doi.org/10.1080/0305215x.2017.1353090.
33. Xue Y, Rui Z, Yu X, Sang X, Liu W. Estimation of distribution evolution memetic algorithm for the unrelated parallel-machine green scheduling problem. *Memetic Comput*. 2019;11(4):423-437. http://dx.doi.org/10.1007/s12293-019-00295-0.
34. Thakur A, Singh B, Gupta A, Duggal V, Bhatt K, Krishnani PD. Improvement in estimation of distribution algorithm (EDA) for fuel loading pattern optimization in AHWR. *Thorium-Energy for the Future*. New York, NY: Springer; 2019:357-364.
35. Wang F, Li Y, Zhou A, Tang K. An estimation of distribution algorithm for mixed-variable newsvendor problems. *IEEE Trans Evol Comput*. 2020;24(3):479-493. http://dx.doi.org/10.1109/tevc.2019.2932624.
36. Zhou B-H, Tan F. A self-adaptive estimation of distribution algorithm with differential evolution strategy for supermarket location problem. *Neural Comput Appl*. 2020;32(10):5791-5804.
37. Doerr B, Krejca MS. Bivariate estimation-of-distribution algorithms can find an exponential number of optima. Paper presented at: Proceedings of the 2020 Genetic and Evolutionary Computation Conference, GECCO'20, Association for Computing Machinery, New York, NY; 2020:796–804.
38. Wang F, Lin Z, Yang C, Li Y. Using selfish gene theory to construct mutual information and entropy based clusters for bivariate optimizations. *Soft Comput*. 2011;15(5):907-915. http://dx.doi.org/10.1007/s00500-010-0557-3.
39. Hong Y, Kwong S, Ren Q, Wang X. Over-selection: an attempt to boost EDA under small population size. Paper presented at: Proceedings of the 2007 IEEE Congress on Evolutionary Computation, Singapore; 2007:1075-1082.
40. Kvålseth TO. On normalized mutual information: measure derivations and properties. *Entropy*. 2017;19(11):631. http://dx.doi.org/10.3390/e19110631.
41. Yang T, Tang Q, Li L, Song J, Zhu C, Tang L. Nonrigid registration of medical image based on adaptive local structure tensor and normalized mutual information. *J Appl Clin Med Phys*. 2019;20(6):99-110.
42. Pelikan M, Sastry K, Goldberg DE. Sporadic model building for efficiency enhancement of the hierarchical BOA. *Genet Program Evolvable Mach*. 2008;9(1):53-84. http://dx.doi.org/10.1007/s10710-007-9052-8.
43. Dawkins R. *The Extended Selfish Gene*. Oxford, UK: Oxford University Press; 2016.
44. Wang F, Yang C, Lin Z, Li Y, Yuan Y. Hybrid sampling on mutual information entropy-based clustering ensembles for optimizations. *Neurocomputing*. 2010;73(7–9):1457-1464. http://dx.doi.org/10.1016/j.neucom.2009.11.011.
45. Tizhoosh H. Opposition-based learning: a new scheme for machine intelligence. Paper presented at: Proceedings of the International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06), Sydney, Australia; 2005;1:695-701.

46. Xu Q, Wang L, Wang N, Hei X, Zhao L. A review of opposition-based learning from 2005 to 2012. *Eng Appl Artif Intell*. 2014;29:1-12. http://dx.doi.org/10.1016/j.engappai.2013.12.004.
47. Ahn CW, An J, Yoo J-C. Estimation of particle swarm distribution algorithms: Combining the benefits of PSO and EDAs. *Inf Sci*. 2012;192:109-119. http://dx.doi.org/10.1016/j.ins.2010.07.014.

How to cite this article: Lin Z, Su Q, Xie G. NMIEDA: Estimation of distribution algorithm based on normalized mutual information. *Concurrency Computat Pract Exper*. 2021;33:e6074. https://doi.org/10.1002/cpe.6074