

# The effectiveness of mutation operation in the case of Estimation of Distribution Algorithms

Hisashi Handa\*

*Department of Information Technology, Faculty of Engineering, Okayama University,  
Tsushima-Naka 3-1-1, Okayama 700-8530, Japan*

Received 28 February 2005; received in revised form 8 July 2006; accepted 15 July 2006

## Abstract

The Estimation of Distribution Algorithms are a class of evolutionary algorithms which adopt probabilistic models to reproduce individuals in the next generation, instead of conventional crossover and mutation operators. In this paper, mutation operators are incorporated into Estimation of Distribution Algorithms in order to maintain the diversities in EDA populations. Two kinds of mutation operators are examined: a bitwise mutation operator and a mutation operator taking account into the probabilistic model. In experiments, we do not only compare the proposed methods with conventional EDAs on a few fitness functions but also analyze sampled probabilistic models by using KL-divergence. The experimental results shown in this paper elucidate that the mutation operator taking account into the probabilistic model improve the search ability of EDAs.

© 2006 Elsevier Ireland Ltd. All rights reserved.

**Keywords:** Estimation of Distribution Algorithms; Mutation operation; KL-divergence

## 1. Introduction

Recently, Estimation of Distribution Algorithms (EDAs) have attracted much attention by Evolutionary Algorithm researchers due to their search abilities (Larrañaga and Lozano, 2003). Genetic operators such as crossover and mutation are not adopted in the EDAs. Instead of this, new populations are generated from the probabilistic model estimated in each generation. This probabilistic model is estimated from the genetic information of individuals, which are selected from the population in each generation. Such reproduction procedure by using the probabilistic model allows EDAs to search for optimal solutions effectively. However, it significantly decreases the diversity of the genetic infor-

mation in the generated population when the population size is not large enough.

This paper discusses the effectiveness of mutation operation in the case of EDAs. Although conventional EDAs do not employ mutation to reproduce individuals in the next generation as we mentioned above, mutation operators are incorporated into conventional EDAs in order to maintain the diversities in populations. Two kinds of mutation operators are examined: a bitwise mutation used in Simple Genetic Algorithm (SGA) (Goldberg, 1989), and a mutation operator taking account into the structure of probabilistic model estimated by EDAs. In order to confirm the effectiveness of the proposed approach, computational simulations on four-peaks problems,  $Fc_4$  function, and MAXSAT problems are carried out (Larrañaga and Lozano, 2003; De Bonet et al., 1996). In order to reveal more about the effectiveness of mutation operation in the case of EDAs, the distributions of reproduced individuals for

\* Tel.: +81 86 251 8250; fax: +81 86 251 8250.

E-mail address: [handa@sdic.okayama-u.ac.jp](mailto:handa@sdic.okayama-u.ac.jp).

each method are analyzed, which is based on Kullback–Leibler divergence (KL-divergence).

Related works are described as follows: the effectiveness of mutation operators in the case of conventional evolutionary algorithms has been studied a long time. Ochoa empirically studied a well-known heuristic with respect to mutation: better mutation probability is around  $1/L$  (string length) (Ochoa, 2002). The relationship between mutual information and entropy was discussed by Toussaint (2003).

In the next section, three kinds of the EDAs, which are employed for computational experiments, will be briefly introduced. Moreover, we will describe how to incorporate mutation operations into the procedure of EDAs. Then, the computational experiments will be examined in Section 4. Section 5 will conclude this paper.

## 2. Estimation of Distribution Algorithms

### 2.1. General framework of EDAs

Estimation of Distribution Algorithms are a class of evolutionary algorithms which adopt probabilistic models to reproduce individuals in the next generation, instead of conventional crossover and mutation operations. The probabilistic model is represented by conditional probability distributions for each variable. This probabilistic model is estimated from the genetic information of selected individuals in the current generation. Hence, the pseudo-code of EDAs can be written as Fig. 1, where  $D_l$ ,  $D_{l-1}^s$ , and  $p_l(\mathbf{x})$  indicate the set of individuals at  $l$ th generation, the set of selected individuals at  $l-1$ th generation, and estimated probabilistic model at  $l$ th generation, respectively (Larrañaga and Lozano, 2003). As described in this figure, the main calculation procedure of the EDAs is that (1) firstly, the  $N$  selected individuals are selected from the population in

the previous generation. (2) Secondly, the probabilistic model is estimated from the genetic information of the selected individuals. (3) A new population whose size is  $M$  is then sampled by using the estimated probabilistic model. (4) Finally, the new population is evaluated. (5) Steps (1)–(4) are iterated until stopping criterion is reached.

This paper discusses the effectiveness of mutation operation in the case of UMDA, MIMIC, and EBNA. These EDAs differ in the representation and estimation methods of the probabilistic model. Since this study is relevant with the representation of the probabilistic models, the representation of the probabilistic models in EDAs will be briefly described in the following sections.

#### 2.1.1. UMDA

Mühlenbein proposed Univariate Marginal Distribution Algorithm (UMDA) in 1996 (Larrañaga and Lozano, 2003; Mühlenbein and Paaß, 1996). As indicated by its name, each variable of the probabilistic model in this algorithm is assumed to be independent of the other variables. That is, the probability distribution  $p_l(\mathbf{x})$  is denoted by a product of univariate marginal distributions, i.e.,

$$p_l(\mathbf{x}) = \prod_{i=1}^n p_l(x_i),$$

where  $p_l(x_i)$  denotes the univariate marginal distribution such that a variable  $X_i$  has a value  $x_i$  at generation  $l$ .

#### 2.1.2. MIMIC

De Bonet et al. proposed Mutual Information-Maximizing Input Clustering (MIMIC), a kind of EDAs whose probabilistic model is constructed with bivariate dependency such as COMIT (Combining Optimizers with Mutual Information Trees) (Larrañaga and Lozano, 2003; De Bonet et al., 1996; Baluja, 2002). Whilst the COMIT generates a tree as dependency graph, the probabilistic model of the MIMIC is represented by a chain based upon a permutation  $\pi$ :

$$p_l(\mathbf{x}) = \prod_{j=1}^{n-1} p_l(x_{i_{n-j}} | x_{i_{n-j+1}}) p_l(x_{i_n}),$$

where the permutation  $\pi = (i_1, i_2, \dots, i_n)$  indicates a sequence of variable indices. This permutation is obtained in every generation. In Fig. 2, the permutation  $\pi$  is set to be  $(i_1, i_2, \dots, i_5) = (5, 2, 4, 1, 3)$  for instance. Furthermore, note that the conditional probability  $p_l(x_{i_{n-j}} | x_{i_{n-j+1}})$  is an abbreviated form of  $p_l(X_{i_{n-j}} = x_{i_{n-j}} | X_{i_{n-j+1}} = x_{i_{n-j+1}})$ .

```

Procedure Estimation of Distribution Algorithm
begin
  initialize  $D_0$ 
  evaluate  $D_0$ 
  until Stopping criterion is reached
     $D_l^s \leftarrow$  Select  $N$  individuals from  $D_{l-1}$ 
     $p_l(\mathbf{x}) \leftarrow$  Estimate the probabilistic model from  $D_l^s$ 
     $D_l \leftarrow$  Sampling  $M$  individuals from  $p_l(\mathbf{x})$ 
    evaluate  $D_l$ 
  end
end

```

Fig. 1. Pseudo-code of Estimation of Distribution Algorithms.

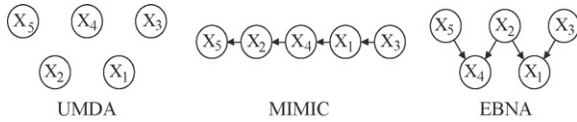


Fig. 2. Probabilistic models for UMDA, MIMIC, and EBNA.

### 2.1.3. EBNA

Like Bayesian Optimization Algorithm (BOA) and Learning Factorized Distribution Algorithm (LFDA), the Estimation of Bayesian Networks Algorithms (EBNA) proposed by Larrañaga et al. adopts Bayesian network as the probabilistic model (Pelikan, 2002; Larrañaga and Lozano, 2003; Pelikan et al., 1999; Mühlenbein and Mahnig, 1999; Mühlenbein, 1998; Larrañaga et al., 2000). That is, the probabilistic model used in the EBNA is written as follows: suppose that  $S$  is the network structure of Bayesian network. In addition,  $U_i^S$  and  $u_i^S$  denote a set of parent variables of variable  $X_i$  and a combination of values in the set of parent variables  $U_i^S$ , respectively. For instance, in Fig. 2, sets  $U_1^S, U_3^S$  of the parent variables of variables  $X_1$  and  $X_3$  are defined as  $\{X_2, X_3\}$  and  $\emptyset$ , respectively. Thus, probability distribution in this case is

$$p_l(\mathbf{x}) = \prod_{i=1}^n p_l(x_i | u_i^S).$$

This conditional probability  $p_l(x_i | u_i^S)$  is also abbreviated for  $p_l(X_i = x_i | X_{j_1} = x_{j_1}, \dots, X_{j_k} = x_{j_k})$  where  $X_{j_1}, \dots, X_{j_k} \in U_i^S$ .

## 3. Mutation operators for EDAs

As mentioned in the previous section, this paper incorporates two kinds of mutation operators: a bitwise mutation used in SGA, and a mutation operator taking account into the structure of probabilistic model estimated by EDAs. The timing of carrying out both mutation operations is the same: mutation operators are applied after “sampling  $M$  individuals from the estimated probabilistic model  $p_l(\mathbf{x})$ ” in Fig. 1. The remain part of this subsection describes both mutation operators.

### 3.1. Bitwise mutation

This mutation operation is the same as used in SGA: let  $P_m$  be the mutation probability. A mutation event at each variable occurs with mutation probability  $P_m$ . The genetic information in sampled individuals is flipped if the mutation event occurs. UMDA, MIMIC, and EBNA with the bitwise mutation are referred as UMDAwBM, MIMICwBM, and EBNAwBM, respectively.

### 3.2. Mutation operators taking account into the structure of probabilistic models

The bitwise mutation does not affect the values in neighbor variables on the probabilistic model. That is, the mutation events occur independently with probability  $P_m$ . On the other hand, the mutation operator introduced here affects the values in descendent variables in the probabilistic model. This notion is applied into MIMIC and EBNA in this paper since the probabilistic model in UMDA does not assume the dependency between variables. MIMIC and EBNA with this mutation operator are referred as MIMICwM and EBNAwM, respectively. The following paragraphs explain them.

In the case of MIMIC, the probabilistic model is represented by a chain of variables so that the change at a certain variable by mutation operator affects values at succeeding variables. Using Fig. 2 as an example, if a mutation event occurs at a variable  $X_1$ , values in succeeding variables  $X_4, X_2, X_5$  might be changed in this mutation operation. The mutation operation for MIMIC is described as follows: at every variable, a mutation event occurs randomly with the mutation probability. If the mutation event occurs at a variable  $X_{ij}$ , the value  $x_{ij}$  at the variable  $X_{ij}$  is flipped into  $x'_{ij} (=1 - x_{ij})$ . If the conditional probability  $p_l(X_{i,j-1} | X_{ij} = x'_{ij})$  is defined in the probabilistic model estimated at  $l$ th generation, the value at succeeding variable  $X_{i,j-1}$  is resampled by using the conditional probability  $p_l(X_{i,j-1} | X_{ij} = x'_{ij})$ . As a consequence of this resampling, if the value at the variable  $X_{i,j-1}$  is changed, the resampling procedure is carried out again for succeeding variable  $X_{i,j-2}$ . This resampling is iterated whilst such changes are continuing.

The mutation operation for EBNA is similar operation as for MIMIC. However, multi-variable dependency in the probabilistic model of EBNA should be taken account. Suppose that a mutation event occurs in a variable  $X_2$  in Fig. 2. In this case, resampling is examined for variables  $X_1$  and  $X_4$  by referring to conditional probabilities  $p_l(X_1 | X_2 = x'_2, X_3 = x_3)$  and  $p_l(X_4 | X_2 = x'_2, X_5 = x_5)$ , respectively.

## 4. Experiments

### 4.1. Experimental settings

This paper examines the effectiveness of mutation operation in the case of EDAs on three kinds of fitness functions: four-peaks function,  $Fc_4$  function, and MAXSAT problems. For first two functions, we investigate how many trials is needed to achieve to optimal solutions for all algorithms. Hence, two indices to evaluate the effectiveness of algorithms are adopted:

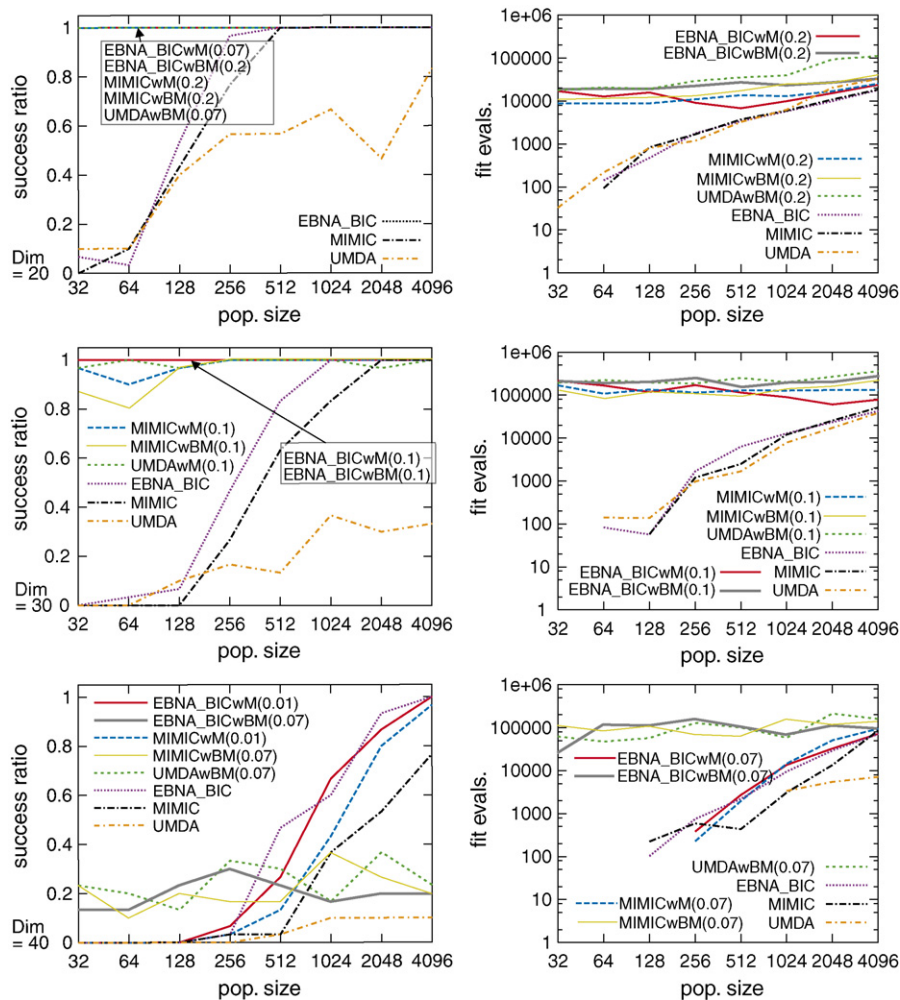


Fig. 3. Experimental results for the four-peaks problems. Success ratio (left), the number of fitness evaluations until finding out optimal solutions (right); problem dimension = 20 (upper), 30 (middle), and 40 (lower).

success ratio (SR) and the number of fitness evaluations until finding out optimal solutions (NOE). The SR is defined as the fraction of runs in which can find out optimal solutions. The NOE in this paper is averaged value over “success” runs. If the SR is 0, the NOE is not defined. Hence, lines in graphs in Figs. 3, 4 and 7 are not plotted for undefined NOE. On the other hand, the solution quality obtained by the proposed

methods and conventional methods for MAXSAT problems are examined.

This paper compares the proposed methods with conventional methods, that is, UMDA, MIMIC and EBNA. This paper employs EBNA<sub>BIC</sub> as EBNA. Genetic parameters used in each examination is summarized in Table 1. For each tuple of parameters indicated in the table, trials are examined. The number

Table 1  
Genetic parameters for each problem

	Four-peaks (20, 30 and 40 variables)	Four-peaks (60 and 70 variables)	$Fc_4$	MAXSAT
Mutation probability	0.2, 0.1, 0.07, 0.05, 0.02, 0.01	0.02, 0.01, 0.005, 0.002, 0.001	0.2, 0.1, 0.07, 0.05, 0.02, 0.01	0.02, 0.01, 0.005, 0.002, 0.001
No. of individuals	32, 64, 128, 256, 512, 1024, 2048, 4096	1024, 2048, 4096, 8192	32, 64, 128, 256, 512, 1024, 2048, 4096	256, 512, 1024, 2048, 4096, 8192
No. of fitness evaluation	1,000,000	1,000,000	100,000	200,000

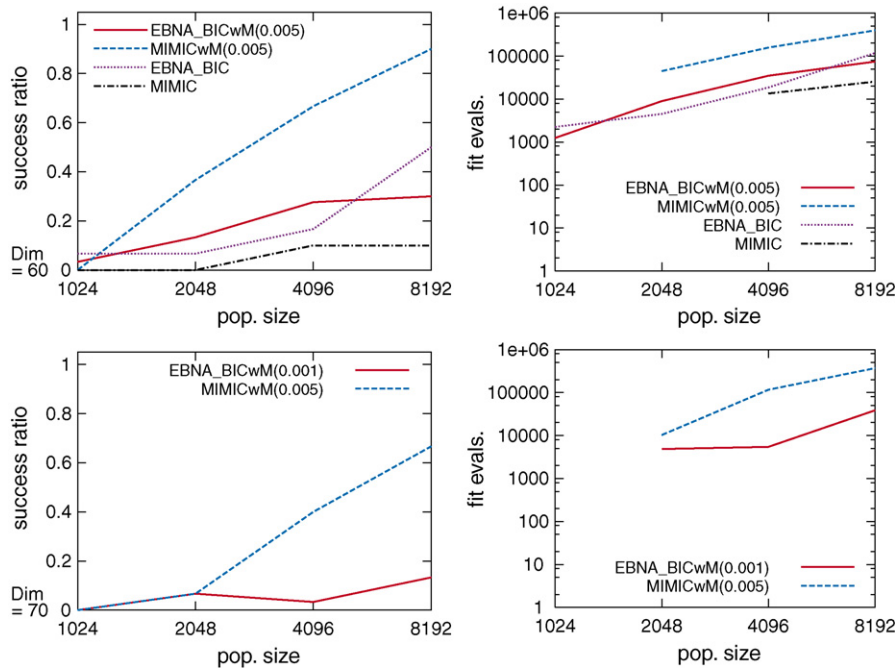


Fig. 4. Experimental results for the four-peaks problems. Success ratio (left), the number of fitness evaluations until finding out optimal solutions (right); problem dimension = 60 (upper), and 70 (lower).

of trials for each tuple is set to be 30 for four-peaks and  $Fc_4$  function, and 10 for each problem instance of MAXSAT. Benchmark problems for MAXSAT, which consists of 50 problem instances for each couple of variables and clauses (Battiti and Protasi, 1997), are used.<sup>1</sup> Additionally, for four-peaks and  $Fc_4$  function, only the best result for the proposed methods over various values of mutation probabilities is plotted. Common settings for all problems are as follows: The number of selected individuals  $N$  is set to be half of the number of individuals  $M$ . The truncation selection method, which selects the best  $N$  individuals from  $M$  individuals, to constitute the selected individuals, is adopted.

## 4.2. Test functions

### 4.2.1. Four-peaks function

$$F_{\text{four-peak}}(T, \mathbf{x}) = \max(\text{head}(x_1, \mathbf{x})) + \max(\text{tail}(x_n, \mathbf{x})) + R(T, \mathbf{x}),$$

$$R(T, \mathbf{x}) = \begin{cases} \frac{1}{2}n & \text{if } (\text{head}(x_1, \mathbf{x}) > T) \text{ and } (\text{tail}(1 - x_1, \mathbf{x}) > T) \\ 0 & \text{otherwise,} \end{cases}$$

where  $\text{head}(b, \mathbf{x})$  and  $\text{tail}(b, \mathbf{x})$  denote the number of contiguous leading bits set to  $b$  in  $\mathbf{x}$ , and the number of contiguous trailing bits set to  $b$  in  $\mathbf{x}$ , respectively. The parameter  $T$  is set to be  $2/n - 1$  in this paper. There are two optimal solutions: 000...0011...111 and 111...1100...000. Further-

more, there are two sub-optimal solutions: 111...1111...111 and 000...0000...000 which can be easily achieved to.

### 4.2.2. $Fc_4$ function

At first, we describe two functions:  $F_{\text{cuban1}}^3$  and  $F_{\text{cuban1}}^5$ :

$$F_{\text{cuban1}}^3(X_1, X_2, X_3) = \begin{cases} 0.595 (X_1, X_2, X_3) = (0, 0, 0) \\ 0.200 (X_1, X_2, X_3) = (0, 0, 1) \\ 0.595 (X_1, X_2, X_3) = (0, 1, 0) \\ 0.100 (X_1, X_2, X_3) = (0, 1, 1) \\ 1.000 (X_1, X_2, X_3) = (1, 0, 0) \\ 0.050 (X_1, X_2, X_3) = (1, 0, 1) \\ 0.090 (X_1, X_2, X_3) = (1, 1, 0) \\ 0.150 (X_1, X_2, X_3) = (1, 1, 1) \end{cases}$$

$F_{\text{cuban1}}^5(X_1, X_2, X_3, X_4, X_5)$  is set to be  $4F_{\text{cuban1}}^3(X_1, X_2, X_3)$  if  $X_2 = X_4$  and  $X_3 = X_5$ . Otherwise,  $F_{\text{cuban1}}^5(X_1, X_2, X_3, X_4, X_5) = 0$ . Then, function  $Fc_4$  is defined as follows

$$Fc_4(\mathbf{x}) = \sum_{c=1}^r F_{\text{cuban1}}^5(X_{5c-4}, X_{5c-3}, X_{5c-2}, X_{5c-1}, X_{5c}),$$

where  $n = 5r$ . This function has only one optimal solution.

### 4.2.3. MAXSAT

In order to solve the MAXSAT problems, we have to find out an assign of values such that the number of satisfied clauses is

<sup>1</sup> <http://rtm.science.unitn.it/intertools/sat/>.



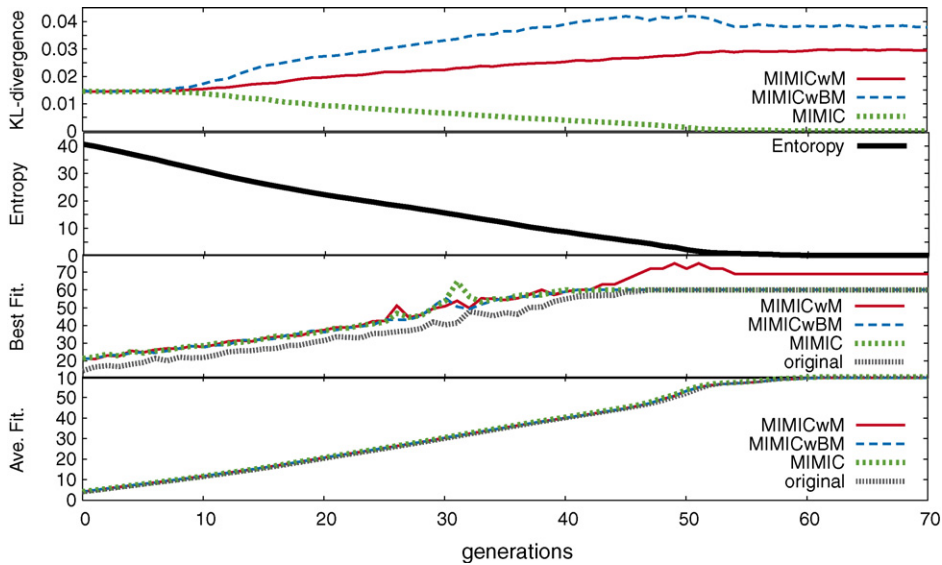


Fig. 5. Experimental results on four-peaks problems. KL-divergence, entropy, the best fitness and averaged fitness over 30 runs.

maximized. That is, this problem is formulated as the following conjunctive normal form (CNF):

$$\bigwedge_j \left( \bigvee_{k_i \in c(j)} k_i \right),$$

where  $c(j)$  denotes a set of literals which belongs in the  $j$ th clauses, and,  $k_i$  indicates literals.

### 4.3. Experimental results

#### 4.3.1. Results on four-peaks function

Figs. 3 and 4 depict the experimental results for the four-peaks problems. Conventional methods, MIMIC and EBNA, with larger population size could find out optimal solutions when dimension = 20. It is difficult to solve for four-peaks problems by separately assigning alleles at each variable, so that UMDA could not solve the four-peaks problems effectively. The proposed method improves the search ability of the conventional EDAs in the viewpoint of the success ratio. Especially, the proposed method with smaller population size could solve the four-peaks problems when they were easy problems. Moreover, only MIMICwM and EBNAwM can find out optimal solutions in the four-peaks function with 70 variables.

#### 4.3.2. Analysis on four-peaks function

In order to investigate the reason why the proposed method, especially, MIMICwM, worked well on the four-peaks problem, this subsection analyzes the genetic information in sampled individuals by MIMIC, MIMICwBM, and MIMICwM. First, we measure how the genetic information in sampled individuals by MIMIC, MIMICwBM, and MIMICwM differs from the probabilistic model. The analysis is carried out by following steps:

1. A normal run by using MIMIC is carried out. At the same time, probabilistic model for each generation is stored.
2. For each stored probabilistic model, new genetic information is sampled by using three different way, i.e., sampling without mutation, sampling with bitwise mutation, and

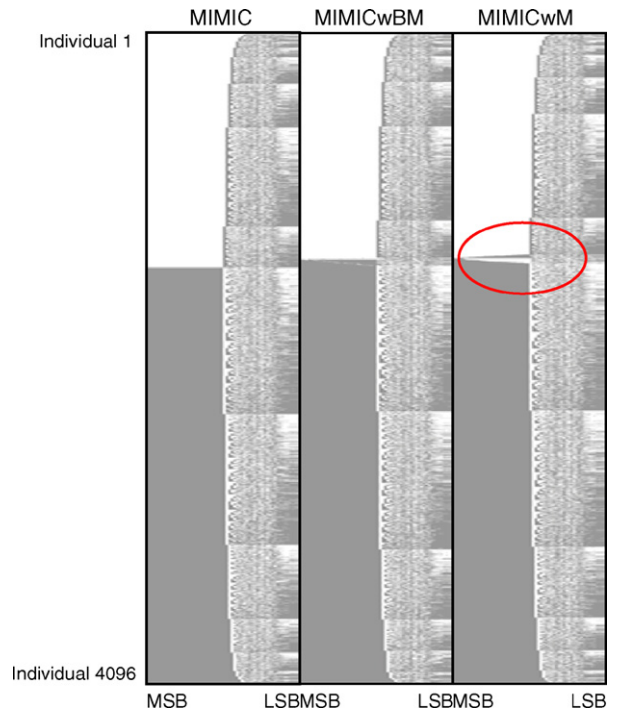


Fig. 6. Experimental results on four-peaks problems. Genetic information sampled by MIMIC, MIMICwBM, and MIMICwM. Individuals are sorted in a certain order in Section 4.3. A dot is painted for value 1's at corresponding loci.

sampling with mutation taking account into the structure of the probabilistic model.

- For each sampling method, a probabilistic model is estimated by using the same structure of the stored probabilistic model. Furthermore, how the estimated probabilistic model differs from the stored probabilistic model is measured by using KL-divergence  $K(\mathbf{x})$ :

$$K(\mathbf{x}) = -p_l^s(X_{i_n}) \log \frac{p_l^s(X_{i_n})}{p^e(X_{i_n})} - \sum_{j=1}^{n-1} p_l^s(X_{i_j}|X_{i_{j+1}}) \log \frac{p_l^s(X_{i_j}|X_{i_{j+1}})}{p^e(X_{i_j}|X_{i_{j+1}})},$$

where  $p_l^s(\cdot)$  denote the stored probabilistic model at  $l$ th generation. On the other hand,  $p^e(\cdot)$  is the probabilistic model estimated from genetic information sampled by each sampling method.

Fig. 5 shows the temporal changes of several quantities in the above experience. The KL-divergence between stored probabilistic model and the probabilistic model estimated from genetic information sampled by each sampling method, entropy of stored probabilistic model, the best and averaged fitness over 30 runs. The entropy is calculated as follows

$$-p_l^s(X_{i_n}) \log p_l^s(X_{i_n}) - \sum_{j=1}^{n-1} p_l^s(X_{i_j}|X_{i_{j+1}}) \log p_l^s(X_{i_j}|X_{i_{j+1}}).$$

As depicted in the figure, the entropy of stored probabilistic models are decreasing. The KL-divergence between MIMIC and the stored probabilistic models is also decreasing. On the other hand, the KL-divergences between the proposed methods and the stored probabilistic models are increasing. The MIMICwBM indicates greater KL-divergence than MIMICwM even if the expected number of changing values in populations of MIMICwM is more than the one of MIMICwBM.

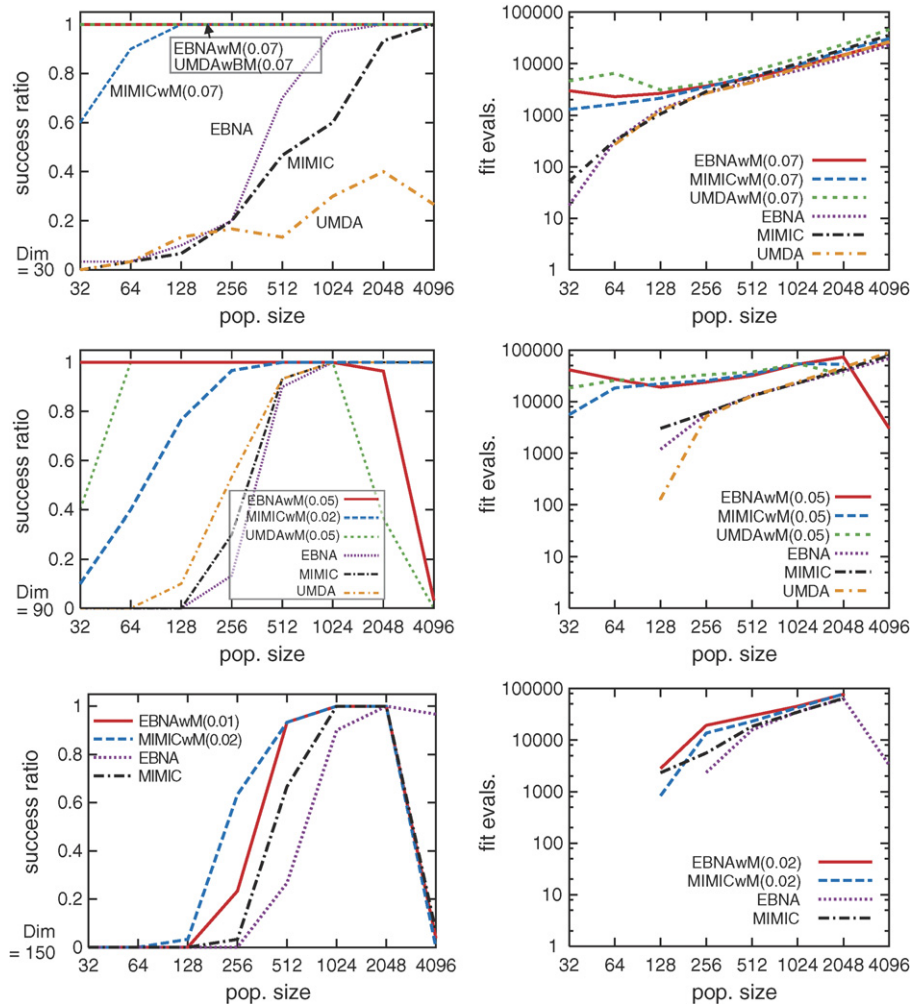


Fig. 7. Experimental results for the  $F_{c4}$  function. Success ratio (left), the number of fitness evaluations until finding out optimal solutions (right); problem dimension = 30 (upper), 90 (middle), and 150 (lower).

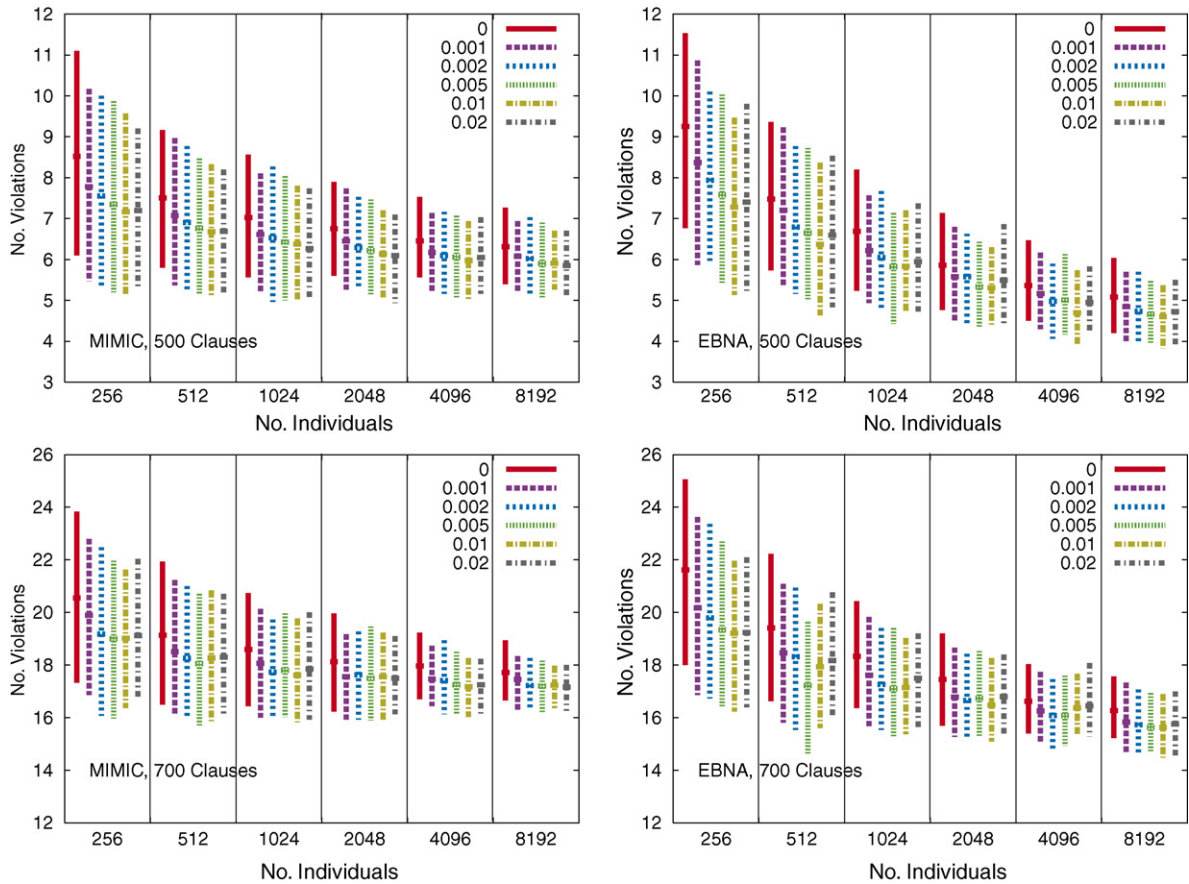


Fig. 8. Experimental results for MAXSAT problems with 100 variables. MIMIC (left) and EBNA (right); 500 clauses (upper) and 700 clauses (lower).

Next, genetic information sampled by MIMIC, MIM-ICwBM, and MIMICwM is plotted in Fig. 6. In this figure, 4096 individuals sampled by each method are classified two groups: individuals whose the most significant bit is 1, and individuals whose the most significant bit is 0. The first group is sorted by descending order of head(0, ·) and the dictionary order. The second group is sorted by ascending order of head(1, ·) and the dictionary order. After sorting, they are concatenated again. For value 1's, a dot is painted. Hence, in the figure, the horizontal axis in the three depiction denotes gene loci, and the vertical axis indicates individuals. As elucidated in this figure, all three methods have two kinds of solutions is coded simultaneously: head(0, ·) and head(1, ·). We can see that MIMICwM (right depiction) try to search for mixtures of two kind solutions by looking at circled area in the figure.

#### 4.3.3. Results on $Fc_4$ function and MAXSAT

Next, experiments on  $Fc_4$  problems as delineated in Fig. 7 are carried out. In this section, experimental results for MIM-ICwBM and EBNAwBM are not plotted because of ease of seeing graphs. The number of fitness evaluations in each run

was limited to 100,000 so that it was impossible for the proposed method whose population size was set to be 4096 to solve the  $Fc_4$  problems with 90 and 150 variables. Except for this, the proposed methods shows better performance in the sense of the success ratio.

Finally, Fig. 8 investigates the quality of acquired solutions on 3-MAXSAT problems with 100 variables. Upper graphs show the results for 500 clauses. On the other hand, lower graphs are the results for 700 clauses. Graphs on the left side and the right side indicates results of MIMIC and EBNA, respectively. For each of the number of population size in all graphs in the figure, six lines are plotted: the solid line denotes the conventional method. Other dashed lines represents corresponding mutation probabilities. As mentioned above, there are 50 problem instances for each couple of (variable, clauses). Ten trials are examined for each problem instance. The highest and lowest points indicates the averaged number of unsatisfied clauses for worst and best solutions in 10 trials, respectively. Moreover, the short horizontal lines crossed to corresponding vertical lines means the averaged value over all (500) trials. All solutions used to depict the graphs is acquired when



the number of fitness evaluations achieves to 200,000. These graphs reveal that the mutation operator proposed in this paper improves the quality of solutions which are acquired after the convergence.

## 5. Conclusion

This paper discussed the effectiveness of mutation in the case of Estimation of Distribution Algorithms from empirical viewpoints. Comparisons on two deceptive functions carried out in Section 4 elucidate that (1) the proposed method works well even if the population size  $M$  of EDAs is not large enough, and (2) only MIMICwM and EBNawM could solve for the most difficult four-peaks problems applied in this paper. Additionally, the computational results for MAXSAT problems reveal that the mutation operator proposed in this paper improves the quality of solutions after the convergence. Finally, analyses on four-peaks function by using KL-divergence explain that MIMICwM try to search for mixtures of two kind solutions. It enables the proposed method to search for better solutions effectively.

## Acknowledgements

This work was carried out while the author was a visiting research fellow at CERCIA, school of computer science, the university of Birmingham, UK. This visit was supported by the Ministry of Education, Science, Sports and Culture, Japan through Overseas Advanced Educational Research Practice Support Program.

## References

- Baluja, S., 2002. Using a priori knowledge to create probabilistic models for optimization. *Int. J. Approx. Reason.* 31 (3), 193–220.
- Battiti, R., Protasi, M., 1997. Reactive search, a history-sensitive heuristic for MAX-SAT. *ACM J. Exp. Algorithm.* 2 (2).
- De Bonet, J.S., et al. 1996. MIMIC: finding optima by estimating probability densities. *Adv. Neural Inform. Process. Syst.* 9, 424–430.
- Goldberg, D.E., 1989. *Genetic Algorithm in Search, Optimization, and Machine Learning*. Addison-Wesley.
- Larrañaga, P., Lozano, J.A. (Eds.), 2003. *Estimation of Distribution Algorithms*. Kluwer Academic Publishers.
- Larrañaga, P., et al. 2000. Combinatorial optimization by learning and simulation of Bayesian. Uncertainty in artificial intelligence. *Proceedings of the 16th Conference*, 343–352.
- Mühlenbein, H., 1998. The equation for the response to selection and its use for prediction. *Evolut. Comput.* 5 (3), 303–346.
- Mühlenbein, H., Mahnig, T., 1999. FDA—a scalable evolutionary algorithms for the optimization of additively decomposed functions. *Evolut. Comput.* 7 (4), 353–376.
- Mühlenbein, H., Paaß, G., 1996. From recombination of genes to the estimation of distributions. I. Binary parameters. *Parallel Problem Solving from Nature, PPSN*, vol. IV, 178–187.
- Ochoa, G., 2002. Setting the mutation rate: scope and limitations of the  $1/L$  Heuristics. *Proceedings of 2002 Genetic and Evolutionary Computation Conference*, 495–502.
- Pelikan, M., 2002. Bayesian optimization algorithm: From single level to hierarchy. PhD Thesis. University of Illinois at Urbana-Champaign, Urbana, IL.
- Pelikan, M., et al. 1999. BOA: the Bayesian optimization algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 1, pp. 525–532.
- Toussaint, M., 2003. The structure of evolutionary exploration: on crossover, building blocks, and Estimation-Of-Distribution Algorithms. *Proceedings of 2003 Genetic and Evolutionary Computation Conference LNCS 2724*, vol. 2, pp. 1444–1455.