Estimation of Distribution Algorithms with Mutation

Hisashi Handa

Okayama University, Tsushima-Naka 3-1-1,
Okayama 700-8530, JAPAN
`handa@sdc.it.okayama-u.ac.jp`
`http://www.sdc.it.okayama-u.ac.jp/~handa/index-e.html`

Abstract. The Estimation of Distribution Algorithms are a class of evolutionary algorithms which adopt probabilistic models to reproduce the genetic information of the next generation, instead of conventional crossover and mutation operations. In this paper, we propose new EDAs which incorporate mutation operator to conventional EDAs in order to keep the diversities in EDA populations. Empirical experiments carried out this paper confirm us the effectiveness of the proposed methods.

1 Introduction

Recently, Estimation of Distribution Algorithms (EDAs) have been attracted much attention in genetic and evolutionary computation community due to their search abilities [1]. Genetic operators such like crossover and mutation are not adopted in the EDAs. In the EDAs, a new population is generated from the probabilistic model constituted by a database containing the genetic information of the selected individuals in the current 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 information in the generated population when the population size is not large enough.

In this paper, we discuss on the effectiveness of mutation operation in the case of EDAs. We propose new EDAs which incorporate mutation operator to conventional EDAs in order to keep the diversities in EDA populations. 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.

Related works are described as follows: The effectiveness of mutation operator in the case of conventional genetic and evolutionary computation 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) [2]. The relationship between mutual information and entropy was discussed by Toussaint [3].

In the next section, we will briefly introduce three kinds of the EDAs, which are employed for our experiments. Moreover, we will describe the basic notion of

G.R. Raidl and J. Gottlieb (Eds.): EvoCOP 2005, LNCS 3448, pp. 112–121, 2005.

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

Fig. 1. Pseudo code of Estimation of Distribution Algorithms

Estimation of Distribution Algorithms with mutation, i.e., the proposed method. Then, computational experiments are examined in section 3. Section 4 will conclude this paper.

2 Estimation of Distribution Algorithms

2.1 General Framework of EDAs

The Estimation of Distribution Algorithms are a class of evolutionary algorithms which adopt probabilistic models to reproduce the genetic information of the next generation, instead of conventional crossover and mutation operations. The probabilistic model is represented by conditional probability distributions for each variable (locus). 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}^{Se}, and $p_l(\mathbf{x})$ indicate the set of individuals at l^{th} generation, the set of selected individuals at $l-1^{\text{th}}$ generation, and estimated probabilistic model at l^{th} generation, respectively [1]. The representation and estimation methods of the probabilistic model are devised by each algorithm. As described in this figure, the main calculation procedure of the EDAs is that (1) the N selected individuals are selected from the population in the previous generation, (2) then, the probabilistic model is estimated from the genetic information of the selected individuals, (3) a new population whose size id M is sampled by using the estimated probabilistic model, and (4) finally, the new population is evaluated.

In this paper, we discuss the effectiveness of mutation operation in case of UMDA, MIMIC, and EBNA. The difference between these EDAs is the representation and estimation of the probabilistic models. Since our study is relevant to the representation of the probabilistic models, we will briefly describe EDAs with a focus on the representation as follows:

- **UMDA:** Mühlenbein proposed UMDA (Univariate Marginal Distribution Algorithm) in 1996 [1, 7]. As indicated by its name, the variables of the

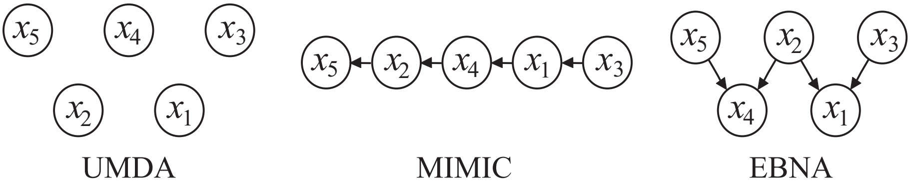

Fig. 2. Probabilistic models for UMDA, MIMIC, and EBNA

probabilistic model in this algorithm is assumed to be independent from 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 $X_i = x_i$ at a variable X_i at generation l.

- **MIMIC:** De Bonet *et al.* proposed MIMIC [1, 8], a kind of EDAs whose probabilistic model is constructed with bivariate dependency such like COMIT [9]. While the COMIT generates a tree as dependency graph, the probabilistic model of the MIMIC is based upon a permutation π.

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

where the permutation π is represented by $(i_1, i_2, \ldots, i_n)$, and is obtained in each generation. In Fig. 2, the permutation π is set to be $(i_1, i_2, \ldots, i_5) = (5, 2, 4, 1, 3)$ for instance.

- **EBNA:** Like BOA and LFDA [10, 11], the EBNA (Estimation of Bayesian Networks Algorithms), proposed by Larrañaga *et al.*, adopts Bayesian Network (BN) as the probabilistic model [1, 12]. That is, the probabilistic model used in the EBNA is written as follows: Suppose that S is the network structure of Bayesian Network, then,

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

where Pa_i^S denotes a set of parent variables of i^{th} variable. For instance, in Fig. 2, sets of the parent variables Pa_1^S, Pa_3^S of variable x_1 and x_3 are defined as $\{x_2, x_3\}$ and $\oslash$, respectively.

2.2 Estimation of Distribution Algorithms with Mutation

In this paper, we incorporate mutation operators into EDAs. The followings introduce incorporated mutation operators for each of UMDA, MIMIC, and EBNA: In the case of UMDA, we adopt the bitwise mutation operator which is the same as SGA: after each bit is decided in accordance with estimated marginal distribution, the mutation operator independently changes the allele of loci with the mutation probability. That is, the succeeding bit production is not affected by the changes by the mutation operator.

On the other hand, since the probabilistic model of MIMIC is represented by a chain of variables, the changes at a certain locus by mutation operator affects the production of alleles at succeeding loci. The mutation operation for MIMIC is described as follows: After producing at a certain locus x_i, whether mutation operation is carried out is randomly decided with the mutation probability. If the mutation operation is occurred, the allele at the locus x_i is flipped. Now, suppose that above mutation operation is carried out at the last produced locus $x_{i_{j+1}}$ in order π. If the conditional probability $p_l(x_{i_{n-j}}|x_{i_{j+1}})$ for flipped allele is not defined[1] the former value produced at first is used for $x_{i_{j+1}}$.

Finally, mutation operation for EBNA is similar to the one in the case of MIMIC, that is, we should take into consideration for succeeding bit production. Now, we assume that we would like to decide the allele at a certain locus x_i and q variables (loci) in the parent set Pa_i^S are flipped their alleles by past mutation events. The conditional probability $p(x_i|Pa_i^S)$ for flipped alleles is used iff such conditional probability is defined. Otherwise, find a defined conditional probability $p(x_i|Pa_i^S)$ such that the number of flipped alleles is maximum, and use it to produce allele at the current locus.

3 Experiments

3.1 Experimental Settings

This paper examines the effectiveness of mutation operation in the case of EDAs on three kinds of fitness functions, whose explanation is described in the next subsection, Four-peaks function, Fc_4 function, and MAXSAT problems. In this paper, we compare the proposed methods with corresponding conventional methods, that is, UMDA, MIMIC and EBNA. We will represent corresponding proposed method as UMDAwM, MIMICwM, and EBNAwM, respectively. This paper employs EBNA_{BIC} as EBNA. For first two functions, we investigate how many trials these algorithms can achieve to optimal solution effectively. Hence, we adopt two indices to evaluate the effectiveness of algorithms: success ratio (SR) and the number of fitness evaluations until finding optimal solutions (NOE). The SR is defined as the fraction of runs in which find optimal solutions. The NOE in this paper is averaged value over "success" runs. If the SR

---

[1] If the flipped allele is not occurred in selected individuals D_{l-1}^{Se}, we cannot calculate the conditional probability $p_l(x_{i_{n-j}}|x_{i_{j+1}})$ for flipped allele.

Table 1. Genetic parameters for each problem

	Four-Peaks (20 and 40 var.'s)	Four-Peaks (60 and 80 var.'s)	Fc_4	MAXSAT
Mutation Prob.	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. Indiv.	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. Fit. Eval.	1,000,000	1,000,000	100,000	200,000

is 0, the NOE is not defined. Hence, lines in graphs in Fig. 3 and Fig. 4 are not plotted for undefined NOE. On the other hand, we examine the solution quality obtained by the proposed methods and conventional methods for MAXSAT problems.

Genetic parameters used in each examination is summarized in Table 1. For each tuple of parameters indicated in the table, trial is examined. The number 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. We use benchmark problems for MAXSAT which consists of 50 problem instances for each couple of variables and clauses [14][15]. Moreover, for Four-peaks and Fc_4 function, we only plot the best result for the proposed methods over various values of mutation probabilities. 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. We use the truncation selection method, which selects the best N individuals form M individuals, to constitute the selected individuals.

3.2 Test Functions

Four-Peaks Function [8]

$$F_{\text{four-peak}}(T,\mathbf{x}) = \max(\text{head}(x_1,\mathbf{x})) + \max(\text{tail}(1-x_n,\mathbf{x})) + R(T,\mathbf{x})$$

$$R(T,\mathbf{x}) = \begin{cases} \frac{3}{2}n & \text{if}(\text{head}(x_1,\mathbf{x}) > T))(\text{tail}(1-x_n,\mathbf{x}) > T) \\ 0 & \text{otherwise,} \end{cases}$$

where head$(b,\mathbf{x})$ and 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: $000\ldots0011\ldots111$ and $111\ldots1100\ldots000$. Furthermore, there are two sub-optimal solutions: $111\ldots1111\ldots111$ and $000\ldots0000\ldots000$ which can be easily achieved to.

***Fc*₄** [1]
At first, we describe two functions: F^3_{cuban1} and F^5_{cuban1}

$$F^3_{cuban1}(x_1,x_2,x_3) = \begin{cases} 0.595 & \text{for } (x_1,x_2,x_3) = (0,0,0) \\ 0.200 & \text{for } (x_1,x_2,x_3) = (0,0,1) \\ 0.595 & \text{for } (x_1,x_2,x_3) = (0,1,0) \\ 0.100 & \text{for } (x_1,x_2,x_3) = (0,1,1) \\ 1.000 & \text{for } (x_1,x_2,x_3) = (1,0,0) \\ 0.050 & \text{for } (x_1,x_2,x_3) = (1,0,1) \\ 0.090 & \text{for } (x_1,x_2,x_3) = (1,1,0) \\ 0.150 & \text{for } (x_1,x_2,x_3) = (1,1,1) \end{cases}$$

$$F^5_{cuban1}(x_1,x_2,x_3,x_4,x_5) = \begin{cases} 4F^3_{cuban1}(x_1,x_2,x_3) & \text{if } x_2 = x_4 \text{ and } x_3 = x_5 \\ 0 & \text{otherwise.} \end{cases}$$

Then, function Fc_4 is defined as follows:

$$Fc_4(\mathbf{x}) = \sum_{c=1}^{r} F^5_{cuban1}(x_{5c-4}, x_{5c-3}, x_{5c-2}, x_{5c-1}, x_{5c}),$$

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

MAXSAT
In order to solve the MAXSAT problems, we have to find an assign of values such that the number of satified clauses is maximized. That is, this problem is formulated as the following CNF (Conjunctive Normal Form):

$$\bigwedge_j (\bigvee_{l_i \in cl(j)} l_i),$$

where $cl(j)$ denotes a set of literals which belongs in the j^{th} clauses. Moreover, l_i indicates literals.

3.3 Experimental Results

Fig. 3 depicts the experimental results for the Four-peaks problems. Conventional methods, MIMIC and EBNA, with larger population size could find optimal solutions when dimension = 20. It is difficult to solve for Four-peaks problems by assigning alleles at each locus independently, so that UMDA could not solve the four-peaks problems effectively. The proposed method improve 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 are easy problems. Moreover, only MIMICwM could solve for Four-peaks problems with 80 variables.

Next, we carried out experiments on Fc_4 problems as delineated in Fig. 4. The number of fitness evaluations in each run was limited to 100000 so that it

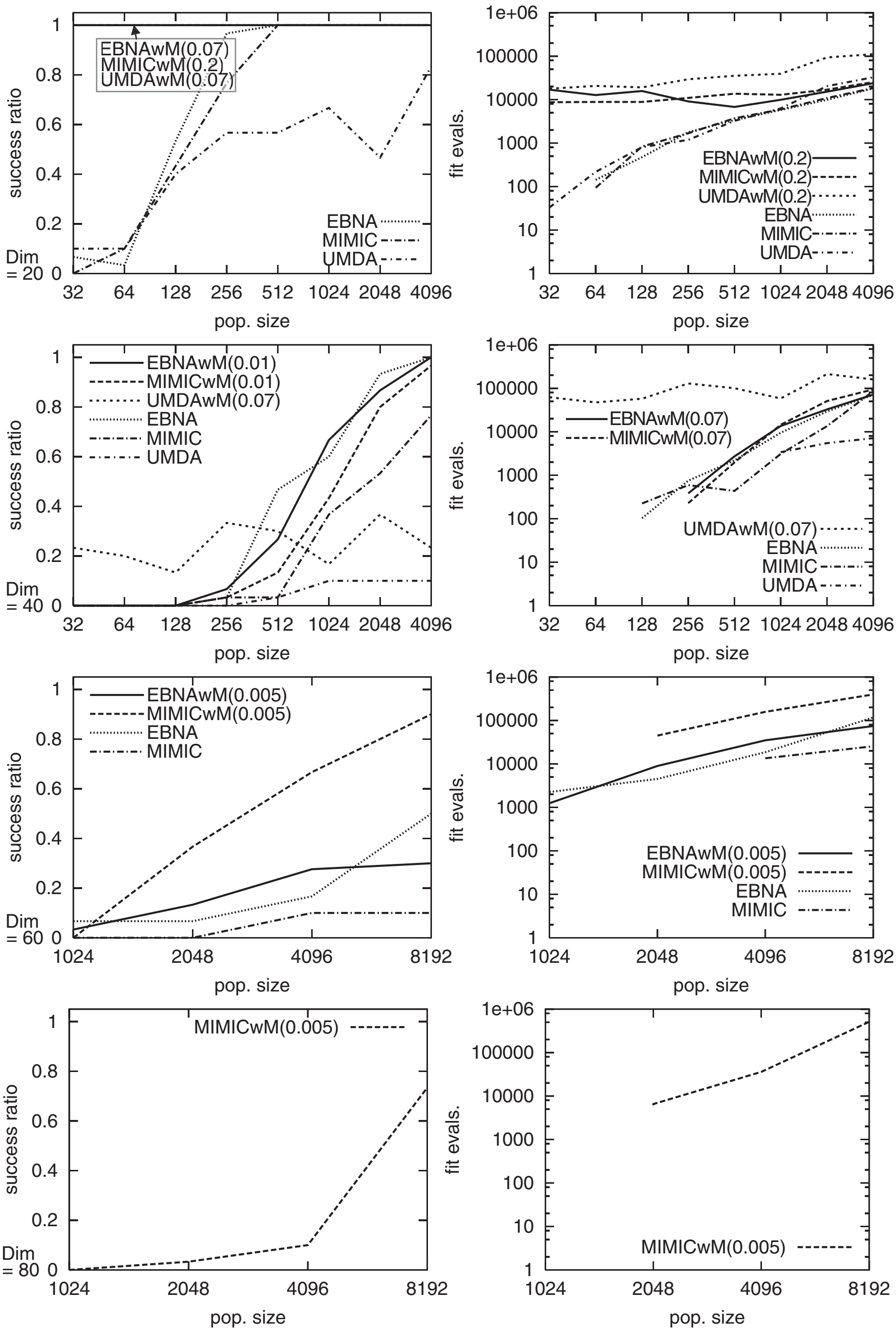

Fig. 3. Experimental results for the Four-peaks problems: Success ratio (LEFT), the number of fitness evaluations until finding optimal solutions (RIGHT); Problem dimension = 20 (UPPER), 40, 60, and 80 (LOWER)

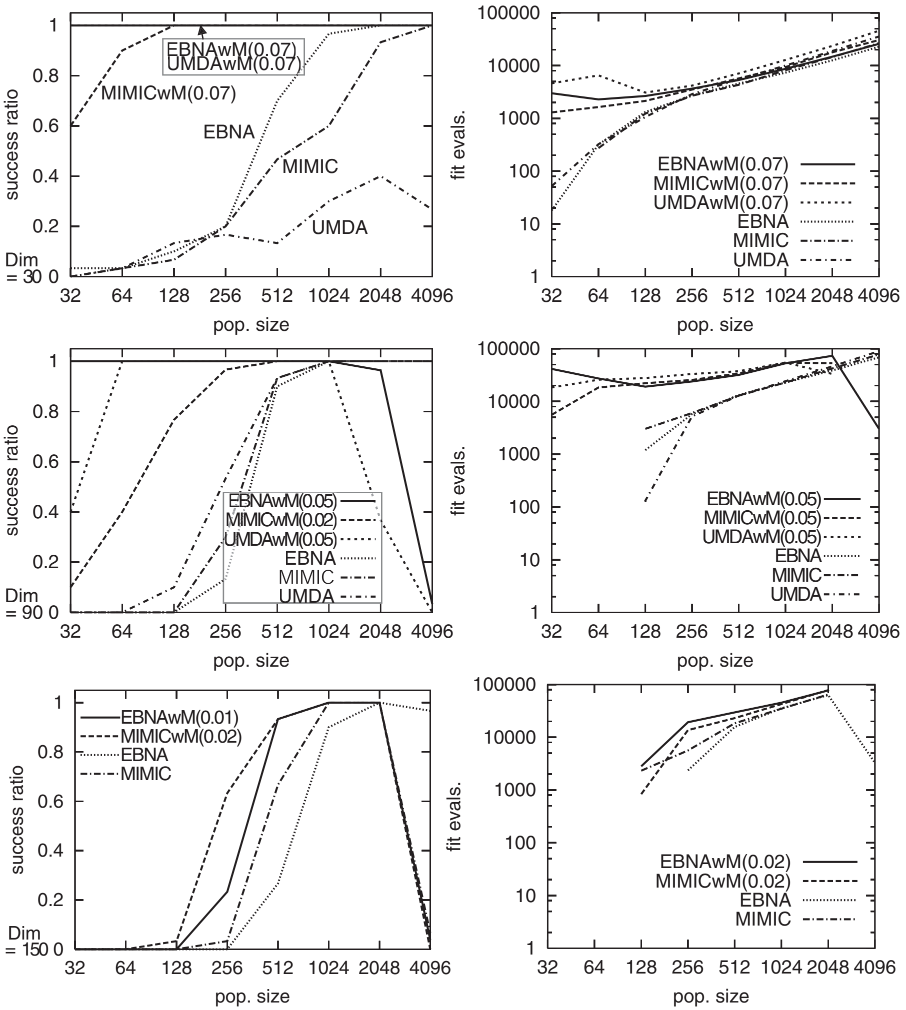

Fig. 4. Experimental results for the Fc_4 function: Success ratio (LEFT), the number of fitness evaluations until finding optimal solutions (RIGHT); Problem dimension = 30 (UPPER), 90 (MIDDLE), and 150 (LOWER)

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. 5 investigated 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 result of MIMIC and EBNA, respectively. For each of the number of population size in all graphs in the figure, 6 lines are plotted: the solid line denotes the conventional method. Other dashed lines

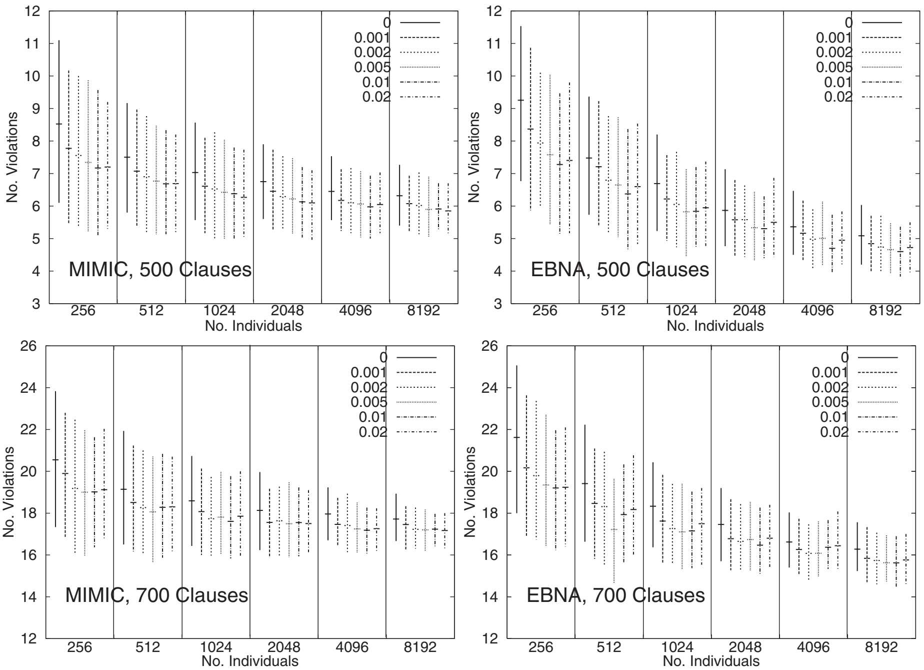

Fig. 5. Experimental results for MAXSAT problems with 100 variables: MIMIC (LEFT) and EBNA (RIGHT); 500 clauses (UPPER) and 700 clauses (LOWER)

represents corresponding mutation probabilities. As mentioned above, there are 50 problem instances for each couple of (variable, clauses). 10 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.

4 Conclusion

In this paper, we discussed on the effectiveness of mutation in the case of Estimation of Distribution Algorithms from empirical viewpoints. Comparisons on two deceptive functions carried out in section 3 elucidate that (1) the proposed method works well even if the population size M of EDAs is not large enough, and (2) only MIMICwM could solve for the most difficult four-peaks problems applied in this paper. Moreover, the computational results for MAXSAT problems reveal that the mutation operator proposed in this paper improves the quality of solutions after the convergence.

References

1. P. Larrañaga and J. A. Lozano Editors: *Estimation of Distribution Algorithms*, Kluwer Academic Publishers (2002)
2. *Proc. of 2002 Genetic and Evolutionary Computation Conference*, (2003) 495–502
3. Marc Toussaint: The Structure of Evolutionary Exploration: On Crossover, Buildings Blocks, and Estimation-Of-Distribution Algorithms, *Proc. of 2003 Genetic and Evolutionary Computation Conference*, LNCS 2724, **2** (2003) 1444–1455
4. Vose, M.D.: The simple genetic algorithm: foundations and theory. MIT Press (1999)
5. González, C., Lozano J.A., Larrañaga, P.: Mathematical Modeling of Discrete Estimation of Distribution Algorithms. Larrañaga, P. and Lozano, J.A. *Eds.*, *Estimation of Distribution Algorithms.* Kluwer Academic Publishers (2002) 147–163
6. M. Pelikan: Bayesian optimization algorithm: From single level to hierarchy, Ph.D. thesis, University of Illinois at Urbana-Champaign, Urbana, IL. Also IlliGAL Report No. 2002023 (2002)
7. H. Mühlenbein and G. Paaß:From Recombination of genes to the estimation of distributions I. Binary parameters. *Parallel Problem Solving from Nature - PPSN IV* (1996) 178–187
8. J. S. De Bonet *et al.*: MIMIC: Finding optima by estimating probability densities, *Advances in Neural Information Processing Systems* **9** (1996)
9. S. Baluja: Using a priori knowledge to create probabilistic models for optimization *International J. of Approximate Reasoning*, **31(3)** (2002) 193–220
10. M. Pelikan *et al.*: BOA: The Bayesian optimization algorithm, *Proceedings of the Genetic and Evolutionary Computation Conference* **1** (1999) 525–532
11. H. Mühlenbein and T. Mahnig: FDA - a scalable evolutionary algorithms for the optimization of additively decomposed functions, *Evolutionary Computation* **7(4)** (1999) 353–376
12. P. Larrañaga *et al.*: Combinatorial Optimization by Learning and Simulation of Bayesian, *Uncertainty in Artificial Intelligence, Proceedings of the Sixteenth Conference* (2000) 343–352
13. The equation for the response to selection and its use for prediction, *Evolutionary Computation*, **5(3)** (1998) 303–346
14. `http://rtm.science.unitn.it/intertools/sat/`
15. R. Battiti and M. Protasi: Reactive Search, a history-sensitive heuristic for MAX-SAT, *ACM Journal of Experimental Algorithmics*, **2(2)** (1997)