LETTER

Estimation of Distribution Algorithm Incorporating Switching

Kenji TSUCHIE[†], ***Nonmember***, **Yoshiko HANADA**[†], ***Member***, ***and*** **Seiji MIYOSHI**[†a)], ***Senior Member***

SUMMARY We propose an "estimation of distribution algorithm" incorporating switching. The algorithm enables switching from the standard estimation of distribution algorithm (EDA) to the genetic algorithm (GA), or vice versa, on the basis of switching criteria. The algorithm shows better performance than GA and EDA in deceptive problems.

key words: *estimation of distribution algorithm, genetic algorithm, Bayesian network, switching*

1. Introduction

The genetic algorithm (GA) is an optimization algorithm based on the mechanics of genetic operators: selection, crossover and mutation [1], [2]. Using these operators, GA tries to derive more suitable solutions by combining building blocks, which are promising partial solutions of short length. This implies that these operators allow GA to search for optima globally. However, crossover and mutation do not consider dependences among variables. Therefore, they often destroy building blocks and prevent the search for a global optimum.

On the other hand, the estimation of distribution algorithm (EDA) involves the use of a stochastic model instead of crossover and mutation in GA [3]. EDA can consider dependences among variables because it uses Bayesian networks [4] for the stochastic model. As a result, EDA can solve complex problems [5]. However, EDA tends to lose the diversity of the population because it generates individuals based on a single stochastic model. Therefore, EDA often yields only a local optimum.

Since GA and EDA have properties different from each other, studies on GA and EDA have been conducted independently. However, we try to build an algorithm that has the advantages of both GA and EDA. In this letter, we propose the estimation of distribution algorithm incorporating switching (SEDA) that appropriately switches the process from EDA to GA, or vice versa.

2. Estimation of Distribution Algorithm Incorporating Switching

In the estimation of distribution algorithm incorporating switching (SEDA), attempts are made to derive more suitable solutions by switching the process from EDA to GA, or vice versa, according to *switching criteria*. The base of the process in SEDA is EDA. Table 1 shows the operation of processes in SEDA.

In step 1, SEDA sets $t = 0$ and generates initial population $P(0)$ randomly; t is the generation number. We will explain the switching flag later in detail together with step 10.

Steps 2 to 5 are processes corresponding to EDA. In step 2, SEDA selects a set of promising individuals $S(t)$ from $P(t)$ with the elitist selection, which is usually used in EDA [6]. In step 3, SEDA constructs the Bayesian network B on $S(t)$. SEDA selects the network B by maximizing the measure of the quality of networks, which is the Bayesian Dirichlet (BD) metric [5], [7]. The BD metric $p(D, B|\varepsilon)$ for network B, data set D and the background information ϵ is defined as

$$
\begin{aligned}
p(D, B|\varepsilon) = \\
& p(B|\varepsilon) \prod_{i=0}^{l-1} \prod_{\pi_{X_i}} \frac{m'(\pi_{X_i})!}{(m'(\pi_{X_i}) + m(\pi_{X_i}))!} \\
& \times \prod_{x_i} \frac{(m'(x_i, \pi_{X_i}) + m(x_i, \pi_{X_i}))!}{m'(x_i, \pi_{X_i})!}, \qquad (1)
\end{aligned}
$$

Table 1 Operation of processes in SEDA.

step	process
1	Set $t = 0$, randomly generate initial population $P(0)$, clear the switching flag.
2	Select a set of promising individuals $S(t)$ from $P(t)$.
3	Construct the Bayesian network B on $S(t)$.
4	Generate a set of new individuals $O(t)$ according to distribution $p(X)$ based on B.
5	Create a new population $P(t + 1)$ by replacing some individuals from $P(t)$ with $O(t)$, set $t = t + 1$, go to step 9.
6	Select a set of promising individuals $Q(t)$ from $P(t)$.
7	Create a new population $Q'(t)$ by applying crossover to $Q(t)$.
8	Create a new population $P(t+1)$ by applying mutation to $Q'(t)$, set $t = t + 1$.
9	If the termination criterion is not satisfied, go to step 10, else stop.
10	If the switching flag is cleared and the switching criterion (EDA→GA) is satisfied, set the switching flag. If the switching flag is set and the switching criterion (GA→EDA) is satisfied, clear the switching flag.
11	If the switching flag is cleared, return to step 2. If the switching flag is set, return to step 6.

Manuscript received April 16, 2010.

[†] The authors are with the Department of Electrical and Electronic Engineering, Faculty of Engineering Science, Kansai University, Suita-shi, 564–8680 Japan.

a) E-mail: miyoshi@kansai-u.ac.jp

DOI: 10.1587/transinf.E93.D.3108

where l is the problem size and $p(B|\epsilon)$ is the prior probability of the network B. $m(\pi_{X_i})$ denotes the number of instances with variables Π_{X_i} (the parents of X_i) instantiated to π_{X_i} in D. $m(x_i, \pi_i)$ denotes the number of instances with X_i instantiated to x_i, as well as Π_{X_i} instantiated to π_{X_i}, in D. $m'(\pi_i)$ and $m'(x_i, \pi_i)$ stand for the values of $m(\pi_i)$ and $m(x_i, \pi_i)$ as prior information, respectively. $m'(\pi_i)$ and $m'(x_i, \pi_i)$ are obtained from the prior probability of the network. In this letter, we assume that there is no prior information. Thus we set $m'(\pi_{X_i}) = m'(x_i, \pi_{X_i}) = 0$ in Eq. (1). Accordingly, we use the following BD metric:

$$p(D, B|\varepsilon) = \prod_{i=0}^{l-1} \prod_{\pi_{X_i}} \frac{1}{m(\pi_{X_i})!} \prod_{x_i} m(x_i, \pi_{X_i})!. \tag{2}$$

In step 4, SEDA generates a set of new individuals $O(t)$ according to the distribution $p(X)$ based on network B constructed in step 3. The distribution $p(X)$ is obtained as

$$p(X) = \prod_{i=0}^{l-1} p(X_i|\Pi_{X_i}) \tag{3}$$

[5]. In step 5, SEDA creates a new population $P(t + 1)$ by replacing some individuals of $P(t)$ with $O(t)$.

Steps 6 to 8 are processes corresponding to GA. In step 6, SEDA selects a set of promising individuals $Q(t)$ from $P(t)$. We can adopt various selection methods, for example, roulette wheel selection and elitist selection [8]. In step 7, SEDA creates a new population $Q'(t)$ by applying crossover to $Q(t)$. We can adopt various crossover methods, for example, two-point crossover and uniform crossover [8]. In step 8, SEDA creates a new population $P(t + 1)$ by applying mutation to $Q'(t)$.

Steps 2 to 8 correspond to the standard EDA and GA. Steps 10 and 11 are particularly important because they are unique to SEDA.

In step 10, SEDA judges whether the process should be switched or not according to *switching criteria*. The *switching criteria* consists of those for EDA→GA and GA→EDA.

This judgment is indicated by using the switching flag. Here, let us suppose that the switching flag is cleared. Then, the process is EDA, and hence SEDA judges whether the switching criterion for EDA→GA is satisfied. If the criterion is satisfied, then SEDA sets the switching flag. Conversely, let us suppose that the switching flag is set. Then, the process is GA, and hence SEDA judges whether the switching criterion for GA→EDA is satisfied. If the criterion is satisfied, then SEDA clears the switching flag.

It is very important how we design the *switching criteria*. First, we explain the switching criterion for EDA→GA. EDA can search in depth for an optimum. However, EDA sometimes converges to a local optimum. The reason is that EDA generates individuals comparable to each other, since it uses the single probabilistic model for generating them. In other words, EDA cannot search the global optimum widely enough since it loses the diversity of the population. To solve this problem, we hope to quantify the reduction of diversity and to use that quantity as the switching criterion for EDA→GA. However, it is difficult to quantify the reduction of diversity. Therefore, we design the switching criterion for EDA→GA as the following: if the best solution in the population is not improved within a certain number of generations, then SEDA switches the process from EDA to GA because the diversity is reduced in this case. We call this period the switching generation period for EDA→GA.

Next, we explain the switching criterion for GA→EDA. We use GA in SEDA in order to recover the diversity of the population. Thus, we hope to quantify the recovery of diversity and to use that quantity as the switching criterion for GA→EDA. We expect that if the process with GA continues for some period, then the diversity is recovered. Therefore, we design the switching criterion for GA→EDA as the following: if the process by GA continues for a certain number of generations, then SEDA switches the process from GA to EDA. We call this period the switching generation period for GA→EDA.

In step 11, SEDA switches the process according to the judgment made in step 10. That is, if the switching flag is cleared, SEDA proceeds to step 2 corresponding to EDA. If the switching flag is set, SEDA proceeds to step 6 corresponding to GA.

3. Experiments and Discussion

We used the deceptive problem, the additively decomposable problem, for experiments [5], [9]. The deceptive problem has a whole fitness function that is composed of several trap functions. The whole fitness function is defined as

$$f(X) = \sum_{i=1}^{m} f_{i,k}(u_i), \tag{4}$$

where X is an individual consisting of $\{0, 1\}$. The length of X is l. m is the number of subproblems, $f_{i,k}(u_i)$ is the subfunction of the ith subproblem. The subfunction $f_{i,k}(u_i)$ is the trap function defined as

$$f_k(u) = \begin{cases} k - 1 - u & \text{if } u < k \\ k & \text{otherwise} \end{cases} \tag{5}$$

where u is the number of '1' in the k-length subproblem. We have omitted i because the same trap functions are used as subfunctions in all subproblems. In the trap function, $u = k$ (that is 111... 1) is the partial optimum. This partial optimum is an isolated point. Moreover, the search proceeds in the opposite direction as this partial optimum because the function value increases as u decreases. Therefore, in each subproblem, partial solutions tend to converge to the partial local optimum corresponding to $u = 0$. This implies the following. The whole fitness function $f(X)$ is maximized when each partial solution obtains $u = k$ in all subproblems, that is, when the whole solution attains $X = 111 \ldots 1$.

The deceptive problem has strong dependences among the variables in subproblems. In many real-world problems, however, dependences are more complicated and overlapped. Therefore, letting each subproblem overlap, we can incorporate the real-world concept into the deceptive problem [10].

We performed experiments on the non-overlapped deceptive problem and the overlapped deceptive problem. All problems have the same whole problem size l = 125 and the same subproblem size k = 5. Here, 'overlapped' means that 1 bit of adjacent subproblems is overlapped with each other.

The performances of GA, EDA and SEDA were compared for each problem. The population size for each algorithm is 200. The best fitness in each generation was calculated until the 10,000 th generation. The average value of fitness in 40 independent runs was used for performance comparison.

First, for GA, two-point crossover with the crossover rate of 100% was used. Roulette wheel selection and elitist selection with the elitist selection rate of 10% were also used. The mutation rate was also set to 4% becauce it is usually set to k/l for decomposable problems such as the deception problem [10]. Second, for EDA, Bayesian networks with the restricted number of incoming edges of 2 were used for the stochastic model. Elitist selection with the elitist selection rate of 10% was also used. Lastly, for SEDA, the settings of GA and EDA were the same as the above settings. The switching generation period for EDA→GA was 10 and that for GA→EDA was 5.

Figures 1 and 2 show the performances of GA, EDA and SEDA on the non-overlapped and the overlapped deceptive problem, respectively. Here, 'Error' represents '1 - Fitness', and 'Fitness' is the normalized fitness with a maximum that corresponds to 1. Error = 0 is not shown because the vertical axis of each figure is a log scale. However, the lines of SEDA that reach the horizontal line means Error = 0, that is, the global optimum, is achieved.

Figures 1 and 2 show that SEDA produced the global optimum irrespective of overlap. On the other hand, GA and EDA were not able to produce the global optimum. Figure 1 shows that the behavior of SEDA corresponded to that of EDA until around the 40 th generation. After that, SEDA achieved the global optimum by around the 2,000 th generation. On the other hand, EDA was not able to achieve the global optimum by the 10,000 th generation. Figure 2 shows that the behavior of SEDA corresponded to that of EDA until around the 20 th generation. After that, EDA had Error smaller than that of SEDA until around the 90 th generation. Subsequently, SEDA had Error smaller than that of EDA. SEDA consequently achieved the global optimum by around the 2,000 th generation, whereas EDA was not able to achieve the global optimum by the 10,000 th generation.

The above results are because of the *switching criteria*. That is, SEDA avoids converging to a local optimum and searches the global optimum in depth by switching the process from EDA to GA before losing the diversity of the population. Accordingly, SEDA reduces the possibility of being trapped to a local optimum. Moreover, SEDA can escape the local optimum and search the global optimum. From Figs. 1 and 2, we see that the property of SEDA is superior irrespective of overlap.

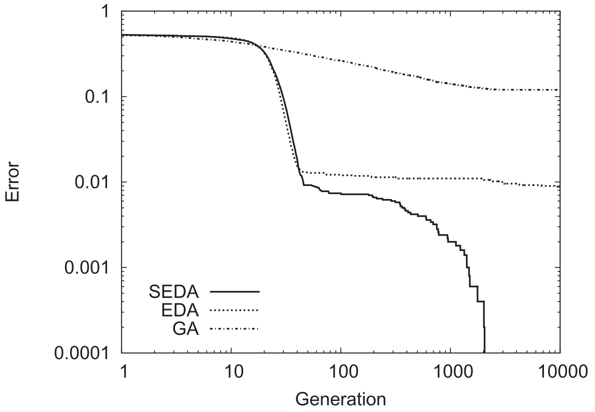

Fig. 1 Performances of GA, EDA and SEDA for the non-overlapped deceptive problem.

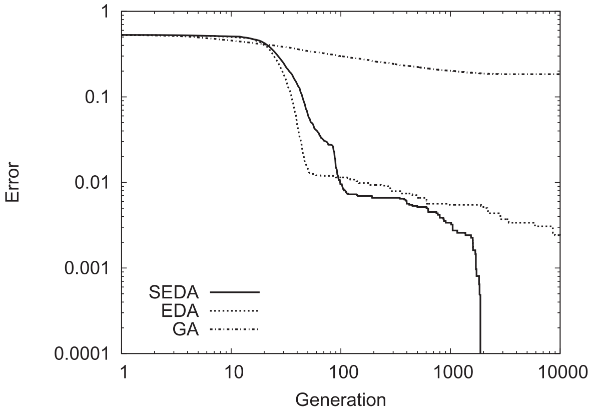

Fig. 2 Performances of GA, EDA and SEDA for the overlapped deceptive problem.

4. Conclusion

We have proposed an "estimation of distribution algorithm" incorporating switching (SEDA), in which the process is switched from the estimation of distribution algorithm (EDA) to the genetic algorithm (GA), or vice versa, according to *switching criteria*. SEDA has the advantages of both EDA – using the dependences between variables – and GA – making a global search for optima. We have conducted experiments with small-scale deceptive problems. As a result, SEDA was found to obtain the global optimum early in deception problems compared with GA and EDA. In this study, we decided the switching criterion heuristically. A future task is to introduce mathematically or theoretically based *switching criteria* into SEDA.

Acknowledgments

This research was partially supported by the Ministry of Education, Culture, Sports, Science, and Technology of Japan, with a Grant-in-Aid for Scientific Research 21500228 and the Kansai University Grant-in-Aid for progress of research in graduate course, 2010.

References

[1] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.

[2] J.H. Holland, "Genetic algorithms," Scientific American, vol.267, pp.66–72, July 1992.

[3] H. Mühlenbein and G. Paaß, "From recombination of genes to the estimation of distributions I. binary parameters," Parallel Problem Solving from Nature-PPSN IV, Lect. Notes Comput. Sci., vol.1411, pp.178–187, 1996.

[4] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference, Morgan Kaufmann Publishers, San Mateo, California, 1988.

[5] M. Pelikan, D.E. Goldberg, and E. Cautü-Pas, "BOA: The Bayesian optimization algorithm," IlliGAL Report no.99003, Jan. 1999.

[6] H. Iba, "Estimation of distribution algorithm and programming," Proc. 2006 Workshop on Information-Based Induction Sciences, pp.9–16, Osaka, Oct. 2006.

[7] D. Heckerman, D. Geiger, and M. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," Mach. Learn., vol.20, no.3, pp.197–243, 1994.

[8] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Kluwer Academic Publishers, Boston, MA, 1989.

[9] K. Deb and D.E. Goldberg, "Analyzing deception in trap function," in Foundations of Genetic Algorithms 2, ed. L.D. Whitley, pp.93–108, Morgan Kaufmann Publishers, 1993.

[10] M. Pelikan, K. Sastry, M.V. Butz, and D.E. Goldberg, "Hierarchical boa on random decomposable problems," MEDAL Report no.2006001, Missouri Estimation of Distribution Algorithms Laboratory, March 2006.