Evaluation of extended Stochastic Schemata Exploiter

T. Maruyama & E. Kita
Graduate School of Information Sciences, Nagoya University, Japan

Abstract

This paper describes the extended Stochastic Schemata Exploiter (ESSE), which has the improved algorithm of the Stochastic Schemata Exploiter (SSE). The algorithm of the ESSE is composed of the original SSE and the ESSE operations. There are seven ESSE algorithms. In the previous study, the authors compared seven ESSE algorithms in some test problems. The ESSE-c1 algorithm shows the best search performance among them. In this paper, the ESSE-c1 algorithm is compared with BOA and SSE in order to confirm the features.

1 Introduction

In most of the combinational optimization problems, the objective function spaces have so-called "big valley structure" [1]. In the problems with big valley structure, there is often the real (global) optimum solution near quasi-optimal solutions. The evolutionary algorithms are considered to be effective for such optimization problems [2–4].

Stochastic Schemata Exploiter (SSE) has been presented by Aizawa in 1994 [5]. In the traditional SGA, better individuals are selected as parents from a population and genetic operators generate new individuals from them. The SSE has a different algorithm than the SGA. In the SSE, the sub-populations are determined according to the descending order of the fitness of the individuals. Common schemata are extracted from the sub-populations and new individuals are generated from them. Selection and crossover operations are not necessary for the SSE. Since the SSE algorithm tends to search better solution near good solutions which are already found, it is adequate for the problems with function spaces of big valley structure. The SSE has two attractive features. First, the control parameters are relatively small because selection and crossover operators are not necessary. Second, the

WIT Transactions on The Built Environment, Vol 91,
www.witpress.com, ISSN 1743-3509 (on-line)
doi:10.2495/OP070051

convergence speed is faster than the SGA. Instead of them, in the SSE, the solution often converges to the local optimum. For overcoming this difficulty, the authors presented Extended Stochastic Schemata Exploiter (ESSE) in the previous study [6]. There are seven algorithms in the ESSE family. In the previous study, they are compared in some test problem. The results indicated that the ESSE-c1 algorithm is the best among them. So, in this paper, the ESSE-c1 is compared with the other algorithms in some test problems.

The remaining of the paper is organize as follows. In section 2, the background of the study is described. In section 3 and 4, the algorithms of the SSE and ESSE are explained. In section 5, their performance is compared in some numerical examples. Finally, the results are concluded in section 6.

2 Background

2.1 Genetic Algorithm (GA)

In SGA, the individuals are distributed over the solution domain. Better solutions can survive to generate their off-springs. The off-springs are generated by genetic operators such as crossover, mutation and so on. The processes are repeated until satisfying the convergence criterion.

2.2 Bayesian Optimization Algorithm (BOA)

A Bayesian network is constructed first as a model of promising solutions after selection. New candidate solutions are then generated by sampling the constructed Bayesian network. Finally, new solutions are incorporated into the population and next iteration is executed unless termination criteria are met.

The BOA has much faster convergence property than the simple genetic algorithm (SGA). It often reaches the local optimal solution than the global one.

2.3 Stochastic Schemata Exploiter (SSE)

In Stochastic Schemata Exploiter (SSE), the individuals are ranked according to the descending order of their fitness function. The sub-populations are generated from the whole population and the common schemata are extracted from the sub-populations. The new individuals are generated from the schemata.

2.4 Extended Stochastic Schemata Exploiter (ESSE)

The aim of an Extended Stochastic Schemata Exploiter (ESSE) is to improve the search performance of the SSE without sacrificing the convergence speed. After extracting schemata from sub-populations, the identical and similar schemata are extracted from the schemata list and the sub-populations are re-generated. Therefore, the search performance can be enhanced.

www.witpress.com, ISSN 1743-3509 (on-line)

3 Stochastic Schemata Exploiter

3.1 SSE algorithm

First, we will show the algorithm of the stochastic schemata exploiter (SSE).

1. Constructing initial population: An initial population is constructed by randomly generating M individuals.
2. Estimating fitness function of individual: The fitness function of individuals is estimated and the individuals are ranked according to the descending order of their fitness.
3. Convergence criterion: If the criterion is satisfied, the process stops.
4. Defining sub-populations: M sub-populations are generated according to the order of the individuals.
5. Extracting common schemata: Common schemata are extracted from the sub-populations.
6. Generating new individuals: The symbol "$*$" in the extracted schemata is randomly replaced by "0" or "1" to generate new individual.
7. Generation alternation: A time step is incremented and the process returns to 2.

The particular processes in the SSE are defining sub-populations, extracting common schemata, and generating new individuals. So, we would like to explain them in the followings.

3.2 Defining sub-populations

The sub-populations are generated according to the semi-order relation between the subpopulations.

The semi-order relation can be explained as follows.

The population P is composed of the individuals $c_1, c_2, \cdots, c_M$, which are numbered according to the descending order of their fitness. Therefore, the individual c_k denotes the $k-$th best individuals in P. The symbol S denotes the sub-population of the population P. When the individual c_k is excluded from S, the new population is represented as $S - c_k$. The operator $\cup$ denotes the union of sets.

When the number of the worst individual in the sub-population S is defined as $L(S)$, the following semi-order relation is held in the sub-populations of the population P;

- the average value of the fitness of the individuals in the sub-population S is higher than that in the sub-population $S \cup c_{(L(S)+1)}$;
- the average value of the fitness of the individuals in the sub-population S is higher than that in the sub-population $(S - c_{L(S)}) \cup c_{(L(S)+1)}$.

By using the semi-order relation, the schemata can be sorted as follows:

1. schema of c_1;
2. common schema between c_1 and c_2;
3. schema of c_2.

www.witpress.com, ISSN 1743-3509 (on-line)

4. common schema between $c_1; c_2$ and c_3.
5. common schema between c_1 and c_3;
6. $\cdots$.

M sub-populations are defined according to the schemata order.

4 Extended Stochastic Schemata Exploiter

The SSE algorithm often generates identical or very similar schemata from different sub-populations. If the identical schemata are excepted, the diversification of the new population can be improved. This is the basic idea of the extended SSE (ESSE) algorithm.

In the ESSE-c1 algorithm, the schemata list is revised when the extracted schemata A and B are identical. The algorithm is as follows.

The ESSE algorithm is composed of the original SSE algorithm and one or more of the above ESSE operations. The process in summarized as follows.

1. An initial population is generated.
2. Fitness function of individuals are estimated.
3. A convergence criterion is confirmed.
4. Sub-populations are generated and common schemata are extracted.
5. Schemata list is revised if the schemata A and B are identical.
   (a) A is kept and B is excluded.
   (b) A common schema is extracted from $S_A \cup S_B$. The symbol S_A denotes the sub-population from which the schema A is extracted.
6. New individuals are generated.

5 Numerical examples

A two-point crossover is adopted for the SGA and GA with MGG. The crossover rate is 1 (100%). The other parameters are selected as best ones from numerical experiments.

A maximum number of the generation is 40,000 in the deception problem and 10,000 in the knapsack problem. The number of the individuals in a population is $n_i = 10, 50$ or 100. 50 runs are performed for each problem from the different initial populations and their average values are shown.

The search performance of the algorithm can be measured with the average value of the final solutions. If the average value is high, the search performance of the algorithm is good. Since, in the evolutionary algorithms such as GA and SSE, the initial population is generated by randomly selected individuals, the final solutions often depend on the initial population. The standard deviation of the final solutions can measure the dependency of the algorithm on the initial population. If the standard deviation is small, we can consider that the algorithm dose not depend on the initial population very well.

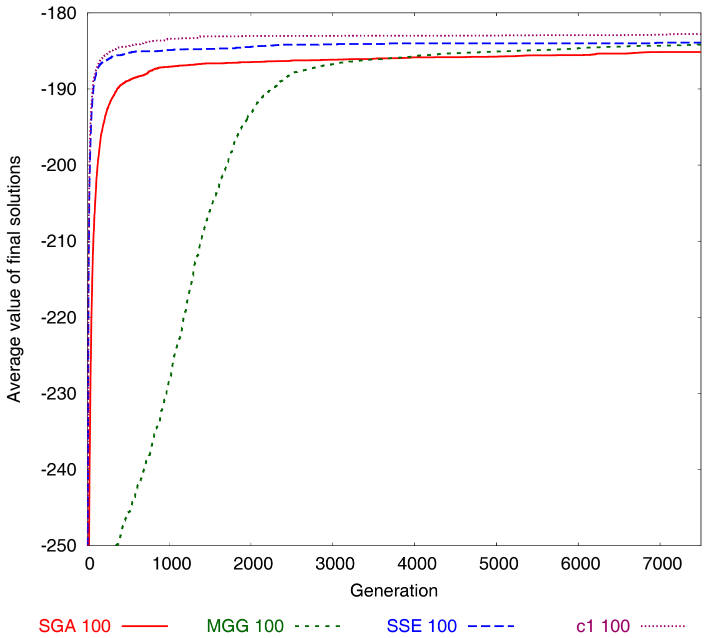

Figure 1: Average values of solutions of SGA, MGG, SSE and c1 on graph partitioning problem.

5.1 Comparison of ESSE-c1, SSE and GAs

5.1.1 Graph partitioning problem [7]

Given a graph G with the set of vertex V and the set of edges E that determines the connectivity between the nodes. The graph partitioning problem consists on dividing G into k disjoint partitions. The object is to minimize k and reduce the imbalance of the weight of the sub-domains. In this paper, the graph problem G124.08 is adopted.

When the graph G is divided into two sets L and RC the total numbers of vertex included in the sets are expressed as (| L | and | R |, respectively. The objective function of the problem is defined as

$$f_{graph}(L, R) = -c(L, R) \\ - \alpha(\mid L \mid - \mid R \mid)^2 \tag{1}$$

where the penalty parameter is taken as $\alpha = 0.1$.

Convergence history of the best fitness is shown in Figure 1. We notice that the convergence speed of the SSE and the ESSE-c1 is much faster than the MGG and that the ESSE-c1 can find the best solution among them.

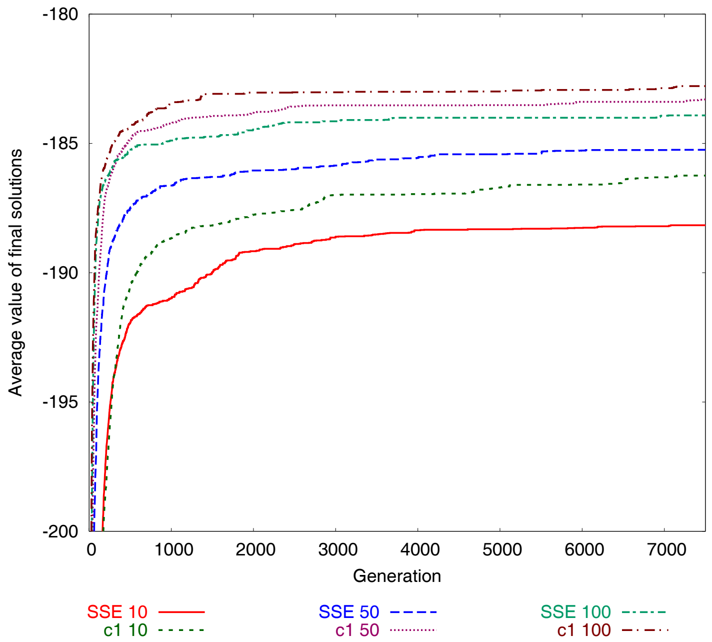

Figure 2: Average values of solutions of SSE and c1 on graph partitioning problem.

The ESSE-c1 is compared with the SSE in the different population sizes. Convergence history of the average value of the best individuals at every iterations is shown in Figure 2. We can recognize that the ESSE-c1 with 50 individuals can find much better solution than the SSE with 100 individuals although the population size is one-half.

5.2 Comparison of ESSE-c1, BOA and SSE

5.2.1 H-IFF Problem [8]

H-IFF problem is a hierarchical Building-Block test function for GAs.

The fitness of a string using H-IFF can be defined using the recursive function. This function interprets a string as a binary tree and recursively decomposes the string into left and right halves. If both child nodes are zeros, the parent node is zero. If both child nodes are ones, the parent node is one. In the other cases, the parent node is -(null).

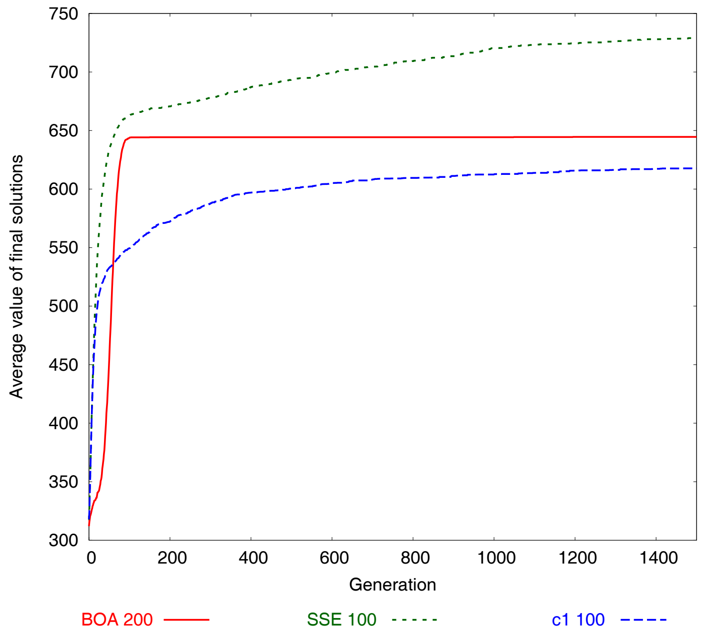

Figure 3: Average values of solutions of BOA, SSE and c1 on HIFF problem.

We shall consider the $k-$ary tree. Assuming the parent node B has the child nodes $b_1, \cdots, b_k$, we can define the problem as

$$t(a,b) = \begin{cases} 0 & \text{if } a = 0 \text{ and } b = 0 \\ 1 & \text{if } a = 1 \text{ and } b = 1 \\ null & \text{otherwise} \end{cases} \tag{2}$$

$$f(a) = \begin{cases} 1 & \text{if } a = 1 \text{ or } a = 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$T(B) = \begin{cases} B_1 & \text{if} |B| = 1 \\ t(T(B^1), \cdots, T(B^k)) & \text{otherwise} \end{cases}$$

$$F(B) = \begin{cases} f(B) & \text{if} |B| = 1 \\ |B| f(T(B)) + \sum_{i=1}^{k} F(B^i) & \text{otherwise} \end{cases}$$

where $|B|$ is the total number of leaves below the node B. Equations (2), (3), (4) and (4) give the state function at a parent node, the partial fitness function at each node, the function developing child nodes and the fitness function of H-IFF problem, respectively.

We will consider here the 16-bit H-IFF problem with a binary tree structure and therefore, $k = 2$.

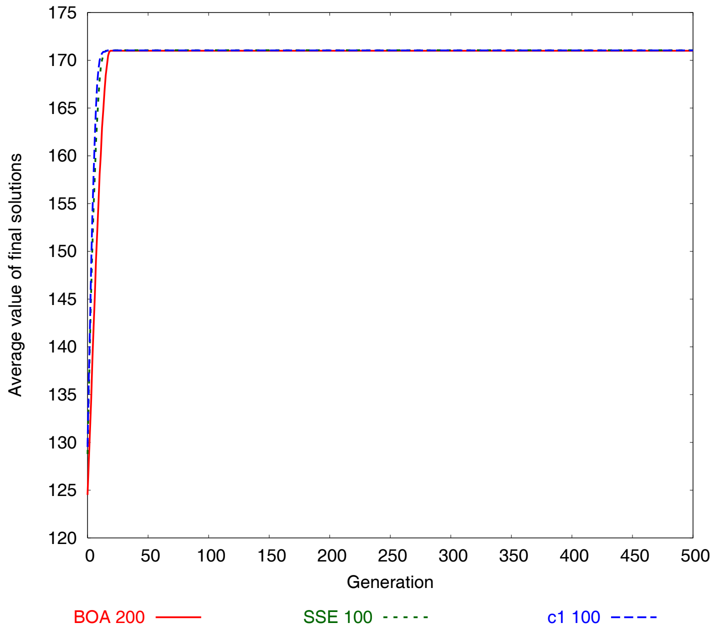

Figure 4: Average values of solutions of BOA, SSE and c1 on H-Trap problem.

The convergence histories of the average value of the fitness of the best individuals are shown in Figure 3. We notice from Figure 3 the following features. The BOA tends to converge the local optimal solution very fast. On the other hand, the SSE can find better solution than the BOA and the ESSE-c1 in this problem.

5.2.2 H-Trap problem [9]

The H-Trap problem is the H-IFF problem with three-ary tree structure; $k \geq 3$. All nodes except for leaves have deception function.

Assuming that total number of ones at child nodes and the total number of child nodes are referred to as u and k, respectively, we can define the fitness function as:

$$f_{trap}(u) = \begin{cases} f_{max} & u = k \\ f_{min} - u \cdot \dfrac{f_{min}}{k-1} & u \neq k \end{cases} \tag{4}$$

where

$$f_{\min} = 0.9, f_{\max} = 1 \quad \text{on root node}$$
$$f_{\min} = f_{\max} = 1 \quad \text{on the other node}$$

The convergence histories of the average value of the fitness of the best individuals are shown in Figure 4. All algorithms show the similar convergence property.

6 Conclusions

This paper described the extended Stochastic Schemata Exploiter (ESSE), which has the improved algorithm of the Stochastic Schemata Exploiter (SSE). The algorithm of the ESSE is composed of the SSE and the ESSE operations. There are seven algorithms in the ESSE family. In the previous study, the ESSE-c1 is the best among them. In this paper, the ESSE-c1 is compared with the others in some numerical examples. The results shows that the ESSE-c1 can find better solution than the original SSE without sacrificing the convergence speed. In the future, we would like to apply the ESSE to the actual optimization problems.

References

[1] D. Yoshizawa, H. Sakanoh, and S. Hashimoto. Coarse-graining newton's method for optimization. In *Trans. IPSJ, Mathematical Modeling and Problem Solving*, number 20, pages 21–24, 2003. In Japanese.

[2] J. H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, 1 edition, 1975.

[3] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1 edition, 1989.

[4] D. Whitly. The genitor algorithm and selection pressure: Why rank-based allocation of reproductive traials in best. In J. D. Shafer, editor, *Proc. 3rd Int. Conf. Genetic Algorithm*, pages 116–121. Morgan Kaufmann Pub., 1989.

[5] N. A. Aizawa. Evolving SSE: A stochastic schemata explointer. In *Proc. 1st IEE Conf. Evol. Comp.*, pages 525–529. IEEE, 1994.

[6] T. Maruyama and E. Kita. Estimation and extension of stochastic schemata exploiter. In *Data Mining VI, Data Mining, Text Mining and their Business Applications (Proceedings of Data Mining 2005)*, pages 45–54, 2005.

[7] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon. Optimization by simulated annealing : An experimental evaluation : Part I, graph partitioning. *Operations Research*, 37:865–892, 1989.

[8] R. A. Watson and J. B. Pollack. Hierarchically-consistent test problems for genetic algorithms. In *Proceedings of 1999 Congress on Evolutionary Computation (CEC 99)*, pages 1406–1413, 1999.

[9] M. Pelikan and D. E. Goldberg. Escaping hierarchical traps with competent genetic algorithms. In Lee Spector, Erik D. Goodman, Annie Wu, W. B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference 2001 (GECCO-2001, San Fransisco, CA)*, pages 511–518, 2001.

www.witpress.com, ISSN 1743-3509 (on-line)