Evolutionary Algorithm Based on Schemata Exploiter

Takashi Maruyama † and Eisuke KITA†

Stochastic Schemata Exploiter (SSE), which has been presented by Aizawa et. al., is one of the evolutionary algorithms based on the stochastic schemata operation. In this paper, the SSE algorithm is explained firstly and the improved algorithms of the SSE, which is named as "Extended SSE", are introduced. The SSE, the ESSE and the simple genetic algorithms (SGA) are evaluated and compared by some test problems. The results indicate that some of the ESSE algorithms have better convergence property than the SSE.

1. Introduction

Most of the combinational optimization problems have so-called "big valley structure"[1]. The evolutionary algorithms are considered to be effective for such optimization problems[2)~4)]. Stochastic Schemata Exploiter (SSE) is one of the evolutionary algorithms.

The Stochastic Schemata Exploiter (SSE) has been presented by Aizawa et. al. in 1994[5]. In the traditional simple genetic algorithms (SGA), the better individuals themselves survive at the next generation. In the SSE, the schemata extracted from better individuals in the population are ranked and the higher-ranked schemata can survive at the next generation. In this paper, the SSE algorithm is explained first evaluated by the solution search of Rastrigin, Rosenbrock, Griewank, Ridge, and Schwefel functions[6]. After that, we present the extended SSE (ESSE) algorithm which is composed of the original SSE and new ESSE operations. The ESSE and the original SSE are compared in the Knapsack problem.

2. SSE Algorithm

We will explain the SSE algorithm here[5].

At the time step t, the individuals in the population P_t are ranked in the descending order of their fitness values, which are named as $c_1, c_2, \cdots, c_M$. The chromosome of the individual c_n is referred as to x_n. Considering the fitness function $f(x)$, the fitness values of the schemata are ranked in the descending order as

$$f(x_1) > f(x_2) > f(x_3) > \cdots. \tag{1}$$

We shall define that the operator $\Lambda(s(H,t))$ denotes the highest ranked schema among the schemata class H at the time t. The highest ranked schema in the population P_t can be defined as $\Lambda(\{c_1\})$, which is equal to the chromosome x_1 of the individual c_1. Considering that

$$\frac{f(x_1)+f(x_2)}{2} > f(x_2), \tag{2}$$

the second highest ranked schema is the schema $\{c_1, c_2\}$, which denotes the common scheme in the individuals c_1 and c_2. As a result, the ranking of the schemata included in the individuals of the population P_t can be re-defined that the ranking of the common schemata $\{c_1\}, \{c_1, c_2\}, \cdots$ is ranked in descending order of their average fitness values.

The SSE algorithm is composed of two subprocesses; the extraction and the ranking of the better schemata from the individuals of the population, and the generation of the new individuals from the extracted schemata.

2.1 Extraction and Ranking of Schemata

We shall firstly introduce the operator $L(S)$ and the symbol $C_{L(S)}$. The operator $L(S)$ means the rank of the sub-population $S(\neq \phi)$ of the population P_t. Besides, the symbol $C_{L(S)}$ denotes the individual of which the rank is the highest among those of the sub-population $L(S)$.

The algorithm to extract the first to Mth highest ranked schemata from the individuals in the population is summarize as follows.

(1) The list $\boldsymbol{a}_{list}$ of the size M is prepared;

$$\boldsymbol{a}_{list}=\{a_{list}(1), a_{list}(2), a_{list}(3), \cdots, a_{list}(M)\}.$$

(2) $i \leftarrow 1$ and $a_{list}(1) \leftarrow c_1$.

(3) At the step i, the sub-populations $S_i \vee$

† 名古屋大学情報科学研究科
Graduate School of Information Sciences, Nagoya University

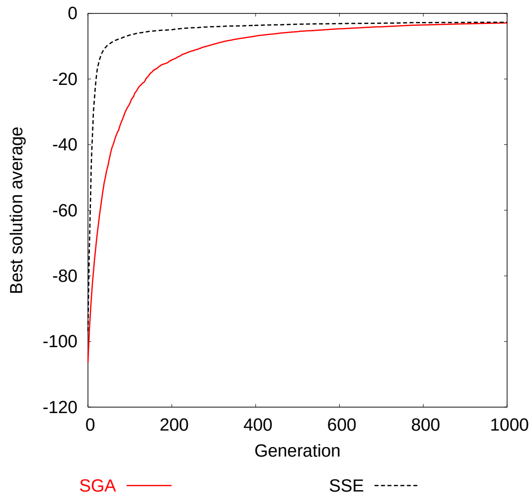

Fig. 1 Comparison of SGA and SSE (Rastrigin function)

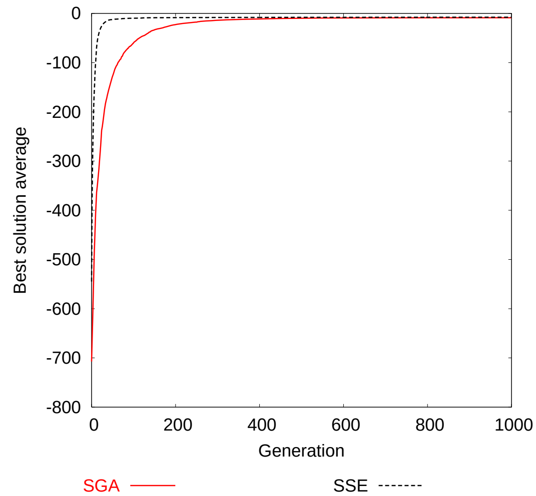

Fig. 2 Comparison of SGA and SSE (Rosenbrock function)

Table 1 Average values of final solutions

Function	SGA	SSE
Rastrigin	-2.8947	-2.6624
Rosenbrock	-8.4694	-7.7780
Griewank	-0.3197	-0.2331
Ridge	-111.97	-6.24

Table 2 Best values of final solutions

Function	SGA	SSE
Rastrigin	-0.0594	-0.0396
Rosenbrock	-0.1909	-0.7185
Griewank	-0.133	-0.0752
Ridge	-3.0	-2.0

$C_{L(S_i)+1}$ and $(S_i - C_{L(S_i)}) \vee C_{L(S_i)+1}$ are generated if $L(S_i) < M$.

(4) The new sub-populations and the list $\{a_{list}(i), \cdots, a_{list}(M)\}$ are re-ranked and added to the $\boldsymbol{a}_{list}$.

(5) If $i < (M-1)$, then $i \leftarrow i+1$ and the process goes to the step 3.

2.2 Generation of New Individuals from Schemata

The individuals of the population P_{t+1} at the next generation t are generated as follows.

(1) The M individuals are generated randomly from the schemata in the list $\boldsymbol{a}_{list}$.

(2) The mutation operation, which is the same as that in the SGA, is applied to the new individuals.

3. Comparison of SGA and SSE

The SGA and SSE are applied to the following problems in order to their search performance; solution search of Rastrigin, Rosenbrock, Griewank, and Ridge functions.

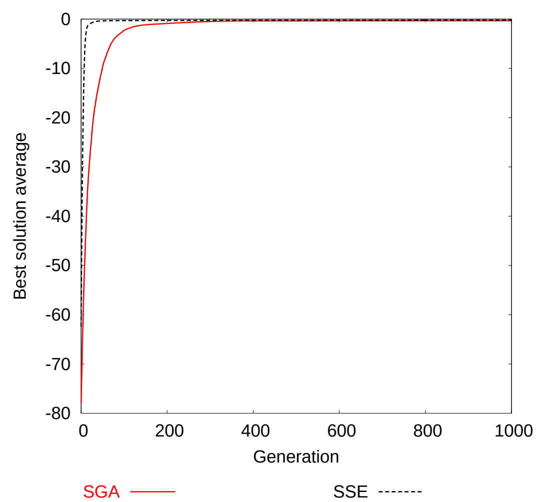

Fig. 3 Comparison of SGA and SSE (Griewank function)

Table 3 Standard deviation of final solutions

Function	SGA	SSE
Rastrigin	1.834	2.1621
Rosenbrock	9.1184	2.4296
Griewank	0.1060	0.0912
Ridge	147.57	3.77

At each problem, 100 runs are performed from the different initial populations. Figures 1 to 4 illustrate the average fitness of the best individuals at the runs at the generation. The abscissa and the ordinate denote the genera-

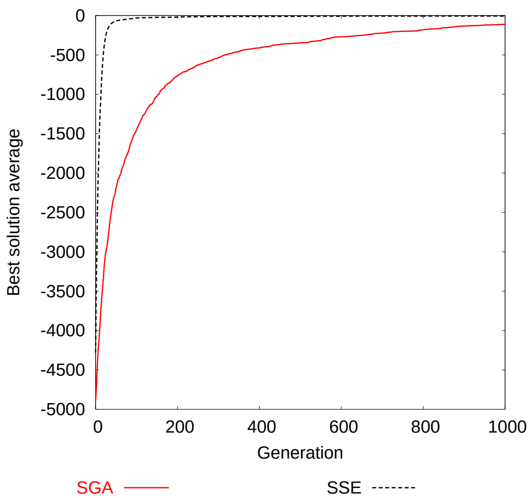

Fig. 4 Comparison of SGA and SSE (Ridge function)

tion number and the average fitness value of the best individuals, respectively. Tables 1, 2, and 3 shows the average and the best values and the standard deviation of fitness values of the final best individuals, respectively.

The comparison of the SSE and SGA solutions indicates the following points;

- The convergence speed of the SSE is higher than that of the SGA.
- The standard deviations of the SSE solutions are less than or equal to those of the SGA solutions in all problems, which indicates the SSE solutions depend on the selection of the initial populations less strongly than the SGA.
- In comparison of the average values and the standard deviation of the final solutions, the SSE solutions are better than the SGA.

4. ESSE Algorithm

The extended SSE (ESSE) algorithm is composed of the original SSE and three kinds of new operations. The reason why the operations are added is to improve the diversity of the individuals. The operations are added individually or complicatedly to the original SSE algorithm.

Consider that the schema A and B are extracted from the different sub-population $S(A)$ and $S(B)$, respectively. We would like to introduce the following ESSE operations which are defined from the relationship between the schema A and B.

4.1 ESSE operation 1

The operation 1 is performed when the schema A and B are identical.

Table 4 Definition of ESSE algorithms

Name	Ope.1	Ope.2	Ope.3
c1	Yes		
c2	Yes	Yes	
c3		Yes	
c4		Yes	Yes
c5			Yes
c6	Yes		Yes
c7	Yes	Yes	Yes
SSE			

(1) The new sub-population $S(C)$ is generated from the sub-population $S(A)$ and $S(B)$ and the schema C is extracted from $S(C)$

(2) $L(S(A))$ is re-calculated from $L(S(C))$.

The operation 1 deletes the identical schemata from the list $\boldsymbol{a}_{list}$ to use them more efficiently than the original SSE and moreover, the use of the operation 1 enables to estimate the fitness value of the schemata accurately.

4.2 ESSE operation 2

The operation 2 is performed when the schema A is included in B.

(1) The new sub-population $S(C)$ is generated from the sub-population $S(A)$ and $S(B)$ and the schema C is extracted from $S(C)$.

(2) If $f(A) \geq f(B)$, $L(S(A))$ and $L(S(C))$ are re-calculated and ranked.

(3) If $f(B) \geq f(A)$, $L(S(B))$ and $L(S(C))$ are re-calculated and ranked.

4.3 ESSE operation 3

The operation 3 is performed when there is the common schema between the schema A and B.

(1) The common schema C is extracted from the schema A and B.

(2) If $f(A) \geq f(B)$, $L(S(A))$ and $L(S(C))$ are re-calculated and ranked.

(3) If $f(B) \geq f(A)$, $L(S(B))$ and $L(S(C))$ are re-calculated and ranked.

4.4 ESSE Algorithms

The ESSE algorithms are defined by adding three ESSE operations to the original SSE individually or complicatedly. So, we can have the seven ESSE algorithms as shown in Table 4.

5. Comparison of SSE and ESSE

For comparing the SSE and the ESSE, we will consider the knapsack problem defined as

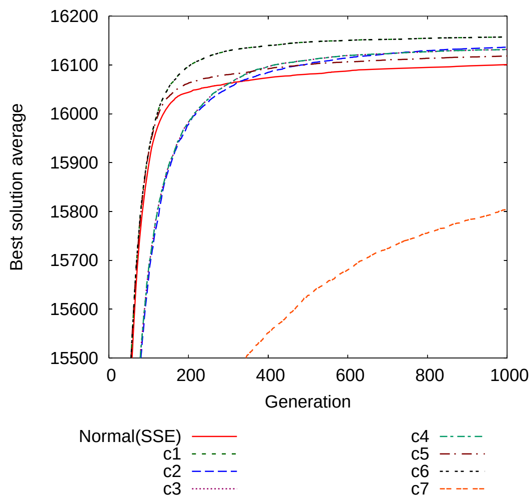

Fig. 5 Comparison of SSE and ESSE

$$\max_{\{x_i\}} \sum_{i=1}^{n} c_i x_i \quad x_i \in 0, 1 \quad (i = 1, \cdots, n),$$

$$\text{subject to} \quad \sum_{i=1}^{n} a_i x_i < b,$$

where the values a_i and c_i denote the weight and the price of the knapsack i. The weight a_i is determined by uniform random numbers from the range $0 \leq a_i \leq 100$. The price c_i is determined by uniform random numbers from the range $0 \leq a_i \leq 100$. The upper limit of the load is $b = 10000$ and the total number of the knapsacks is $n = 400$.

Figure 5 illustrates the convergence property of the best individuals. The abscissa and the ordinate denote the generation number and the average values of the best individuals, respectively. The comparison of the final best solutions indicates that the SSE and the ESSE algorithms are better in the order as

c1, c6, c2, c3, c4, c5, SSE, and c7.

In the convergence speed, they are ranked as

c1, c6, c5, SSE, c3, c4, c2, and c7.

We can conclude that the ESSE operation 1 is the most effective among three operations for improving the performance of original SSE algorithm.

6. Conclusions

Firstly, the Stochastic Schemata Exploiter (SSE) was introduce and evaluated by the solution search of Rastrigin, Rosenbrock, Griewank, Ridge, and Schwefel functions. As the results, we could confirm that the convergence speed of the SSE is faster than that of the SGA.

Next, we described the extended SSE (ESSE) algorithm which is composed of the original SSE and three ESSE operations. The ESSE and the SSE are compared in the Knapsack problem. The results show that some of a family of ESSE algorithms are better than the SSE.

Acknowledgement

This work was performed as the part of 21st century Center of Excellence Project "Frontier of Computational Sciences" under the auspices of Japanese Ministry of Education, Culture, Sports, Science and Technology.

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) D. Whitly, K. Mathias, S. Rana, and J. Dzubera. Building better test functions. In L. J. Eshelman, editor, *Proc. 6th Int. Conf. Genetic Algorithm*, pages 239–246. Morgan Kaufmann Pub., 1995.