社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

Cross-Generational Elitist Selection SSEの収束特性について

丸山　崇[†]　北　栄輔[††]

確率的スキーマどん欲法には，速い収束速度と制御パラメータが少ないという特徴がある．その一方で，しばしば局所解に収束してしまう欠点がある．この欠点を解決するために，本研究ではcross-generational elitist selection SSE (cSSE) を提案する．この方法では，cross-generational elitist selectionを利用することで，SSEの個体集団の多様性を改善し，大域的探索性能を改善する．数値例において，cSSEをminimum generation gap (MGG) を用いた遺伝的アルゴリズムやBayesian Optimization Algorithm (BOA)，SSEと比較する．計算結果により，cSSEは他のアルゴリズムよりも，短い計算時間で同程度以上の精度の解を求められることを示す．

Convergence Analysis of Cross-Generational Elitist Selection SSE

TAKASHI MARUYAMA[†] and EISUKE KITA[††]

Stochastic Schemata Exploiter (SSE) has interesting features such as very fast convergence speed and only one control parameter. It sometimes converges to local optimum solution. In this paper, one describes cross-generational elitist selection SSE (cSSE). Since the use of the cross-generational elitist selection enhances the diversity of the individuals in the population, the global search performance is improved. In the numerical examples, cSSE is compared with genetic algorithm with minimum generation gap (MGG), Bayesian Optimization Algorithm (BOA), and original SSE. The results show that cSSE can find better solutions at shorter CPU time than the other algorithms.

1. Introduction

The search performance of the simple Genetic Algorithm (SGA) depends on the early convergence and evolutionary stagnation[1),2)]. The early convergence means that all individuals gather to same local optimum solutions at early generation and therefore, the global optimum solution can not be found. The evolutionary stagnation means that the convergence speed slows down. For overcoming these problems, a new generational alternation model, Minimal Generation Gap (MGG), was presented by Sato et.al.[3)].

Bayesian Optimization Algorithm (BOA) is also one of evolutionary computations[4)]. BOA searches a solution by using Bayesian network learned from the information of the better solutions in the population. Since offspring are generated from the stochastic model, evolutionary operations in SGA such as selection, crossover, and mutation are not necessary.

Stochastic Schemata Exploiter (SSE) was presented by Aizawa[5)]. Individuals are ranked according to the descending order of their fitness. Sub-populations are generated according to the semi-order relation of sub-populations. Common schemata are extracted from the individuals in each sub-population. New individuals are generated from the schemata. Since SSE can spread better schemata over the whole population faster than the GA, the convergence speed of SSE is also faster than that of the GA. However, SSE sometimes converges to local optimum solution.

The aim of this study is to improve the search performance of SSE without sacrificing the convergence speed. For this purpose, one presents cross-generational elitist selection SSE (cSSE), in which the cross generational elitist selection[6)] is introduced to the original SSE. The use of the cross-generational elitist selection enhances the diversity of the population and therefore, the global search

† 前名古屋大学大学院生 (ソネット・エンターテイメント (株))
†† 名古屋大学

performance is improved. In the numerical examples, cSSE is compared with genetic algorithm with minimum generation gap (MGG), Bayesian Optimization Algorithm(BOA), and the original SSE in some numerical examples.

2. cSSE Algorithm

2.1 Process of cSSE

The cSSE process is summarized as follows.

(1) An initial population is constructed by randomly generating M individuals.

(2) The fitness function of individual is estimated and individual is ranked according to the descending order of their fitness.

(3) If the criterion is satisfied, the process stops.

(4) M sub-populations are generated according to the order of the individuals.

(5) Common schemata are extracted from the individuals in sub-populations.

(6) M offspring are generated from M schemata.

(7) The cross generational elitist selection makes a new population by the individuals selected from parent population and offspring population.

(8) A generation is incremented and the process returns to 2.

2.2 Defining sub-populations

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 $c_1, c_2, \cdots, c_M$. The individuals are numbered according to the descending order of their fitness. The sub-population of the population P is referred as S. When the individual c_k is excluded from S, the new sub-population is as $S - c_k$. The operator $\cup$ denotes the union of sets. The rank of the worst individual in the sub-population S is represented as $L(S)$. Since the worst individual in the sub-population S is $c_{L(S)}$, $c_{(L(S)+1)}$ denotes the worse one by one rank than $c_{L(S)}$ in the whole population P.

We can find the following semi-order relation among the sub-populations of the population P;

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

For example, the semi-order relation shows the order of the schemata as follows:

(1) Schema of c_1; i.e, chromosome of c_1 itself.

(2) Common schema between c_1 and c_2.

(3) Schema of c_2.

(4) Common schema between c_1 and c_3.

(5) $\cdots$.

2.3 Extracting Common Schemata

After defining the sub-populations, the common schemata are determined as the product set of the chromosomes of the individuals in the sub-populations.

2.4 Generating New Individuals

The extracted schemata are composed of three characters; "0"s, "1"s, and "*"s. The new individuals are defined by randomly replacing "*" by "0" or "1".

2.5 Cross Generational Elitist Selection

Algorithm of cross generational elitist selection is written as follows[6].

(1) At generation $t-1$, offspring are generated from individuals in the population.

(2) Populations of parents and offspring are indicated with $P(t-1)$ and $O(t-1)$, respectively.

(3) $P(t-1)$ and $O(t-1)$ are merged to new population $P'(t-1)$.

(4) Individuals in $P'(t-1)$ are ranked according to their fitness.

(5) The population $P(t)$ is generated by selecting best M individuals from $P'(t-1)$.

3. Numerical Examples

cSSE is compared with GA with MGG, BOA, SSE, and cSSE in some problems[7].

In GA with MGG, two-point crossover of crossover rate $= 1$ is employed and all individuals are replaced at every generations. In all algorithms, the best mutation rates are determined from numerical experiments.

Maximum generation is 40,000 for deception problem and 15,000 for knapsack problem, respectively. Population size is specified as $N_i =$

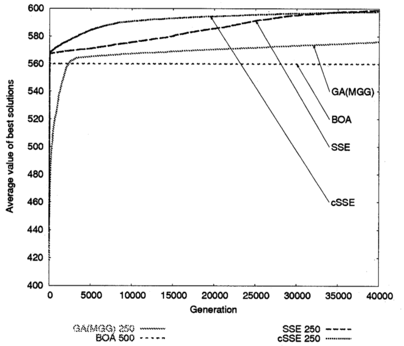

図 1 だまし問題における最良解の適応度平均値
Fig. 1 Average fitness value of solutions on deception problem

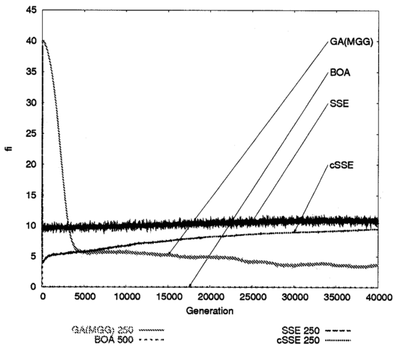

図 2 だまし問題における多様性パラメータ f_i
Fig. 2 Diversity parameter f_i in deception problem

10, 50, 100 or 250 for GA with MGG, SSE and cSSE and $n_i = 20, 100, 200$ or 500 for BOA, respectively. Since BOA replaces half population size at every generations, computational cost of fitness function is half as much as the other algorithms. For equalizing the computational cost of all algorithms, the population size of BOA is twice as many as the other algorithms. Simulations are performed 50 times from different initial populations. The average fitness values of the best individuals are compared.

3.1 Deception Problem

The deception problem is defines as the summation of the 4-bit deception sub-problem[7]. The objective function f is defined as

$$f_{deception} = \sum_{i=1}^{n} f_d(x_i) \tag{1}$$

where n denotes the number of 4-bit deception problem and $n = 10$. The design variable x_i of the problem are defined as $x_i \in 0000, 0001, \cdots, 1111$.

The history of average fitness of the best individual is shown in Fig.1. The abscissa and the ordinate denote the generation and the fitness value, respectively. We notice from Fig.1 that the convergence speed of MGG is the slowest and that the cSSE is the fastest among them. Although BOA has very fast convergence speed, it may be attracted to a local optimum solution because the final BOA solution is worse than the other.

Next, we will discuss the diversity of solutions in deception problem. For that purpose, we will define the diversity parameter f_i as follows.

$$f_i = \frac{\sum_{i=1}^{M-1} \sum_{j=i+1}^{M} \mid p_i - p_j \mid}{{}_M C_2} \tag{2}$$

where M and p denote population size and an individual, respectively. $\mid p_i - p_j \mid$ denotes the hamming distance between p_i and p_j. Therefore, the diversity of the individuals in the population depends on the magnitude of the parameter f_i.

Figure 2 shows the history of the parameter f_i. While the parameter in BOA decreases to zero immediately, the parameters in the others are relatively big till the final generation. We can recognize that this is the reason why the BOA is caught to the local solution.

3.2 Knapsack Problem

When there are n baggage in a knapsack, the knapsack problem is defined as the maximization of the value of the knapsack without exceeding the weight limit b. The problem is defined as

$$\begin{aligned} &\max_{\{x_i\}} \sum_{i=1}^{n} c_i x_i \\ &\quad \text{subject to} \sum_{i=1}^{n} a_i x_i \leq b \\ &\quad x_i \in 0, 1 \qquad (i = 1, \cdots, n) \end{aligned} \tag{3}$$

where a_i and c_i denote the weight and the value of the baggage i, respectively. They are randomly taken within $1 \leq a_i, c_i \leq 100$ and $b = 10000$ and $n = 400$.

The history of the average fitness value of the best individuals is shown in Fig.3. Figure 3 illustrates that the convergence speed of MGG is the

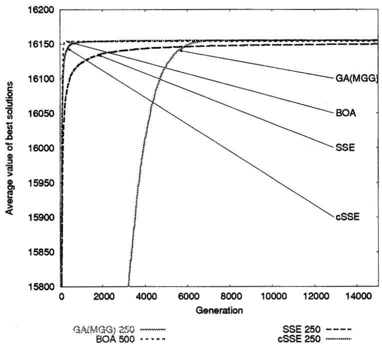

図 3 ナップザック問題における各世代の最適解の平均適応度
Fig. 3 Average fitness value of solutions on knapsack problem

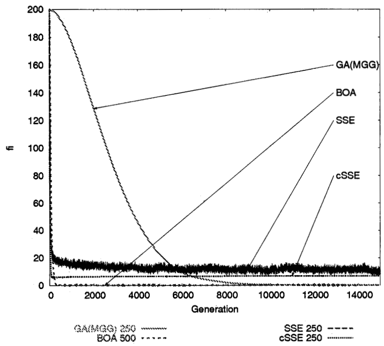

Fig. 3 Diversity parameter f_i in knapsack problem

slowest among them.

Figure 3.2 shows the history of the parameter f_i. Also in this case, the parameter in BOA decreases to zero immediately. On the other hand, the parameters in the others are relatively big till the final generation.

4. Conclusions

In this study, we described the cross generational elitist selection SSE (cSSE) in which the generation alternation model of original SSE was changed to cross generational elitist selection model. The presented algorithm was compared with GA with MGG, BOA, and SSE.

Numerical results shows the following features of cSSE.

In comparing search performance of algorithms, note that cSSE can find slightly or much better solution than the others in all examples. Specially, cSSE shows very good performance in the deception problem. In the problem, the average fitness value of the final cSSE solutions is better than that of GA with MGG by about 5%.

In comparing the convergence speed of algorithms, note that cSSE is much faster than GA with MGG and similar to BOA and original SEE in the deception. Although, in the knapsack problem, cSSE is slightly slower than BOA, their difference is small.

Finally, we can conclude that cSSE is better than the others from both viewpoints of the search performance and the convergence speed.

参 考 文 献

1) J.E. Baker. Reducing bias and inefficiency in the selection algorithm. *Proceedings of the Second International Conference on Genetic Algorithms*, pp. 14–21, 1987.
2) 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*, pp. 116–121. Morgan Kaufmann Pub., 1989.
3) H. Satoh, I. Ono, and S. Kobayashi. Minimal generation gap model for gas considering both exploration and exploitation. *Proc. IIZUKA'96*, pp. 494–497, 1996.
4) M. Pelikan, D. E. Goldberg, and E. Cantu-Paz. Boa: The bayesian optimization algorithm. In W. Banzhaf, Daida J, A. E. Elben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference 1999 (GECCO-1999, San Fransisco, CA)*, pp. 525–532. Morgan Kaufmann., 1999.
5) N. A. Aizawa. Evolving SSE: A stochastic schemata explointer. In *Proc. 1st IEE Conf. Evol. Comp.*, pp. 525–529. IEEE, 1994.
6) L. J. Eshelman. The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. In G.J.E. Rawlins, editor, *Foundations of Genetic Algorithms 1991 (FOGA 1)*, pp. 265–283. Morgan Kaufmann., 1991.
7) L. D. Whitley. Fundamental principles of deception in genetic search. In G. J. E. Rawlins, editor, *Foundations of Genetic Algorithms 1991 (FOGA 1)*, pp. 221–241. Morgan Kaufmann, 1991.