World Automation Congress © 2010 TSI Press.

SIMULATED ANNEALING WITH SEARCH MECHANISMS OF INTERPOLATION AND EXTRAPOLATOLATION DOMAIN

††TOMOYUKI HIROYASU,†YUTA TOMATSU, †MASAHIRO NAKAO, ‡MIKI MITSUNORI, ††HISATAKE YOKOUCHI, ‡MASATO YOSHIMI

†Graduate School of Engineering, Doshisha University
††Department of Life and Medical Sciences, Doshisha University
‡Faculty of Science and Engineering, Doshisha University

Appropriate interpolation and extrapolation search methods are important in evolutionary computation. This paper presents a discussion of interpolation and extrapolation search methods in continuous optimization problems. First, we defined interpolation and extrapolation searches determined from the search domains, and introduce the stochastic multiple point search algorithm, which is based on simulated annealing, as an interpolation and extrapolation search algorithm. The effectiveness of interpolation and extrapolation search are discussed through numerical test functions. The results showed that interpolation search increases accuracy of the search for optimum solutions and extrapolation search increases the discovery rate of optimum solutions.
Key Words:

1. INTRODUCTION

Sakuma introduced the distance scale into the domain of design variables for genetic algorithms (GA), defined a crossover method for generating children between parents as interpolation crossover method, and defined a crossover method for generating children toward a direction away from parents as extrapolation crossover method [1].

Here, we propose simulated annealing (SA) with introduction of a mechanism of interpolation and extrapolation crossover, and introduce the two-point search algorithm, which is based on SA, as an interpolation and extrapolation search algorithm. The effectiveness of interpolation and extrapolation search was discussed through numerical test functions.

2. SIMULATED ANNEALING

By simulating annealing—a process employed to obtain a perfect crystal by the gradual cooling of a melted solid—SA obtains the minimum energy value. This energy is equivalent to the objective function value in conventional optimization problems.

The SA algorithm consists of three operations: generation, acceptance, and cooling. The generation operation changes the current solution x and generates the next solution x' using a probability distribution. The acceptance operation decides whether the change is acceptable. This acceptance is determined from the difference $\Delta E(=E'-E)$ of the current energy $\Delta E = f(x)$ and energy of the next solution $\Delta E = f(x')$ as well as the temperature parameter T. Metropolis *et al*. introduced a simple algorithm (shown in (1)) to provide efficient simulation.

$$P = \begin{cases} 1, & if(\Delta E < 0) \\ \exp(-\frac{\Delta E}{T}), & otherwise \end{cases} \tag{1}$$

That is, if $\Delta E < 0$, the change is accepted. Otherwise, the modification is accepted at a certain probability. The cooling operation generates the temperature of the next state from the temperature of the current state. If the temperature parameter T is large, the probability of accepting the solution with energy larger than that of the previous solution increases, while the probability of accepting the solution with smaller energy decreases if T is low. Therefore, at the beginning of the simulation, both the temperature and the acceptance levels must be high. As the simulation proceeds and temperature decreases, the search point attains the global optimum solution.

3. RELATIONSHIP BETWEEN LOCAL SEARCH AND GLOBAL SEARCH

It is important for neighborhood of SA to become large when search points exist near suboptimal solutions, and become small when search points exist near optimal solution. The design space can be divided into two domains: the interpolation domain and the extrapolation domain. Given a distance measure *d*, the interpolation domain *Din* and the extrapolation domain *Dex* are defined as (2) (3), where *S* denotes the design space. The interpolation domain and the extrapolation domain are illustrated in Fig.1.

$$D_{in} = \{s \in S \mid d(s,x_1) \le d(x_1,x_2) \text{ and } d(s,x_2) \le d(x_1,x_2)\} \quad (2)$$

$$D_{ex} = \{s \in S \mid d(s,x_1) > d(x_1,x_2) \text{ and } d(s,x_2) > d(x_1,x_2)\} \quad (3)$$

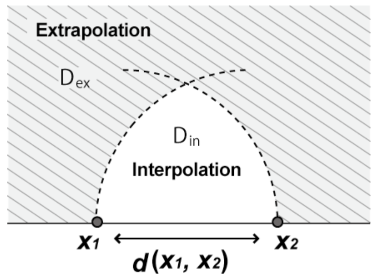


Fig. 1 Concept of interpolation and extrapolation domains

When a suboptimal solution lies between two search points, it is not possible to find the suboptimal solution unless we search between the two points creating an interpolation domain. On the other hand, When a suboptimal solution does not lie between two search points, it is not possible to find the suboptimal solution unless we search between the two points creating extrapolation domain. Therefore, it is necessary for SA to switch the search between interpolation and extrapolation domains according to the situation.

4. PROPOSED ALGORITHM

To define the interpolation and extrapolation domains in SA, we propose an algorithm that explores the design space with two search points. First, the algorithm sets two search points in the design variable domain at random. Second, the algorithm searches using the distance between the two points as a threshold value. We describe the search point with highest evaluation as the best search point, and the search point with lower evaluation as the sub-best search point. When the distance is shorter than the threshold value, the algorithm considers that the suboptimal solution does not exist, and then switches the search into the extrapolation domain. If the distance between two points is large, the algorithm considers that the suboptimal solution will exist and then switches the search into the interpolation domain. In this way, the algorithm switches the search between interpolation and extrapolation domains according to the situation.

4.1 Interpolation search

In this section, we discuss how to explore the design space using the interpolation search. The algorithm generates search points in the interpolation domain, which lies between two points as shown in Fig. 1. Then, we defined two interpolation domains: one domain considered the correlation between design variables, while the other did not. The former is a hypercube as in Fig. 2(a), while the latter is a hypersphere as in Fig. 2(b). The interpolation search is as follows.

[Algorithm of interpolation search]

Step 1: Generate *N* search points between the best search point and sub-best search point.

Step 2: The energy of the best point of *N* search points is compared with the energy of sub-best search point. In addition, the sub-best search point is selected if it meets the requirement of Metropolis. If there is no point meeting the requirement of Metropolis, the point at the shortest distance from the best search point is selected.

Step3: Return to Step 1 if the distance between the best search point and the sub-best search point is outside the threshold value *Z*.

Step4: Perform extrapolation search if the distance between the two search points is shorter than *Z.*

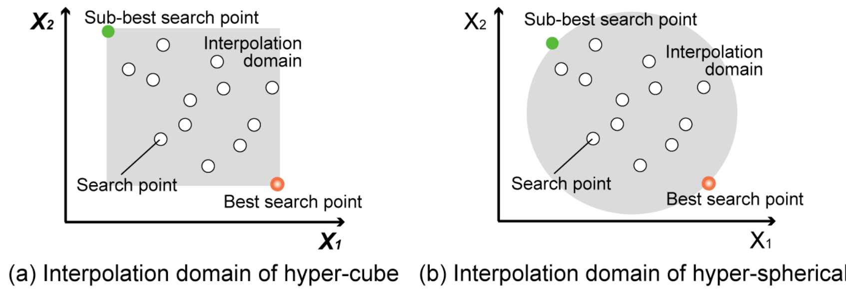


Fig. 2 Interpolation domain between two points

4.2 Extrapolation search

Extrapolation search proceeds in a direction away from a certain search point. Therefore, the search will restart if searching stops in interpolation search.

Extrapolation search is as follows.

[Algorithm of interpolation search]

Step 1: Generate M search points in extrapolation domain created by dividing design value in neighborhood R.

Step 2: The energy of the best point of M search points is compared with the energy of the best point among M search points. The sub-best search point is selected if it meets the requirement of Metropolis. If there is no point meeting the requirement of Metropolis, the point at the shortest distance from the best search point is selected.

Step 3: Return to Step 1 if annealing step is less S times.

Step 4: Perform interpolation search if annealing step is more S times.

----

5. PERFORMANCE VERIFICATION OF THE PROPOSED ALGORITHM

5.1 Optimization problems

Three functions were used as the optimization problems in this study: the Rastrigin function, Schwefel function, and Rosenbrock function. The Rastrigin function and Schwefel function are multi-peak functions with no correlations between design variables. The Rosenbrock function is a single-peak function in which there are correlations between design variables. We chose the Rastrigin function and Schwefel function to verify the global search performance of the proposed SA. We chose the Rosenbrock function to verify the local search performance of the proposed SA. These three functions are shown below.

○Rastrigin function

$$f(x) = 10n + \sum_{i=1}^{n}[x_i^2 - 10\cos\{2\pi \cdot x_i\}] \quad (-5.12 \le x_i \le 5.12)$$

$$\min(f(x)) = F(0,0,\ldots,0) = 0$$

○Schwefel function

$$f(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{|x_i|}) \quad (-512 \le x_i \le 512)$$

$$\min(f(x)) = F(420.97,\ldots,420.87) = -418.98 \cdot n$$

○Rosenbrock function

$$f(x) = \sum_{i=1}^{n-1}\{100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\} \quad (-2.048 \le x_i \le 2.048)$$

$$\min(f(x)) = F(1,1,\ldots,1) = 0$$

5.2 Numerical simulation

The parameters of the proposed SA are shown in Table I with reference to Ono[2]. We compared the proposed algorithm with "Simulated Annealing with Advanced Adaptive Neighborhood (SA/AAN)"[2], because we verified the performance of the proposed algorithm. The dimension number was 5. We performed 20 trials with the proposed SA and SA/AAN. The results of rearranging optimum solutions in ascending order are shown in Fig. 3. The vertical axis in Fig. 3 shows the values and the horizontal axis shows trial step.

We verified the search in the interpolation domain with a hypercube in Fig. 2(a) and hypersphere in Fig. 2(b) to verify the interpolation domain in interpolation search. The optimization functions were the Rastrigin function and Rosenbrock function in 5 dimensions. The results are shown in Fig. 4.

To verify the availability of the extrapolation search in the proposed algorithm, we evaluated two methods: (1) changed Step 4 in the algorithm of interpolation search to "random sub-best search point search," and (2) the proposed algorithm explained in section 4. We call (1) "interpolation and random search" and (2) "interpolation and extrapolation domain." The optimization function is each Rastrigin function and Schwefel function in 5 dimensions. The results are shown in Fig. 5.

Table I. Parameters of the proposed algorithm

Annealing steps	320000
Cooling steps	32
Max (initial) temperature	10
Min (final) temperature	0.01
Search number in interpolation domain	10
Threshold value	1.E-05
Step of search in extrapolation domain	10
Search number in extrapolation domain	10
Neighborhood in extrapolation search	5

6. RESULTS AND DISCUSSION

6.1 Verification of algorithm performance

With the exception of the Rosenbrock function, the proposed algorithm showed better performance than SA/AAN (Fig. 3). We consider that it is bad for the solution accuracy of the proposed algorithm in the Rosenbrock function to form a hypercube interpolation domain. The interpolation domain of the proposed algorithm is a hypercube in Fig. 2(a). Therefore, interpolation search is dependent on treatment of the coordinate axes. In fact, the solution accuracy of proposed algorithm was poorer in the Rosenbrock function because interpolation search does not consider correlations between design variables.

6.2 Determination of interpolation domain

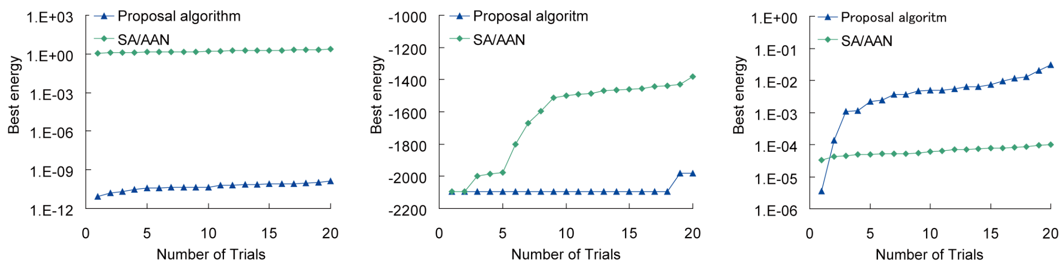
The solution accuracy of the hyperspherical domain was better than that of the hypercube domain shown in Fig. 4(a), and the solution accuracy of the hypercube domain was better in Fig. 4(b).

We consider that interpolation search with a hypercube domain is effective to search in the coordinate axes as the Rastrigin function because interpolation search searches each design variable in dimension. The Rosenbrock function, in which there are correlations between design variables, is effective to search design variables in all domains. As a result, the Rosenbrock function is effective to search the interpolation domain as shown in Fig. 2(b).

6.3 Effectiveness of extrapolation search

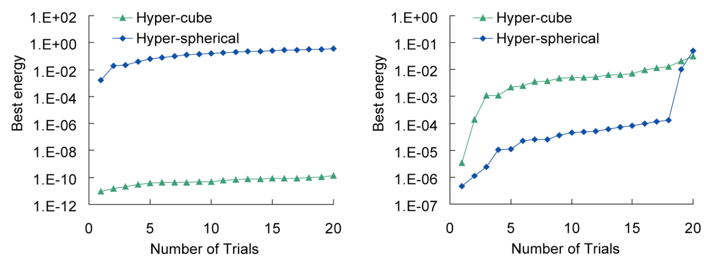
The solution accuracy of "interpolation and extrapolation search" was better than that of "interpolation and random search" as shown in Fig. 5(a), and the solution accuracy of "interpolation and random search" was better as shown in Fig. 5(b).

We consider that the differences in the results were caused by the differences in the search number in interpolation and extrapolation domains between the two methods. To verify this suggestion, we counted the search number in the interpolation and extrapolation domains. The results are shown in Table II.



(a) Rastrigin function (b) Schwefel function (c) Rosenbrock function

Fig. 3 Distribution of optimum solution with the proposed algorithm and SA/AAN



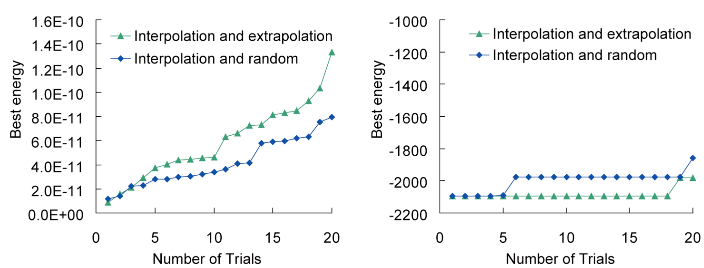
(a) Rastrigin function (b) Rosenbrock function

Fig. 4 Distribution of optimum solutions in hypercube domain and hyperspherical domain

"Interpolation and extrapolation domain" required fewer searches in the interpolation domain than "interpolation and random search" in the Rastrigin function (Table II). When we investigated the situation of search, two methods has found valley of optimal solution. The accuracy of the solutions obtained by the Rastrigin function in "interpolation and extrapolation search" was lower because of the reduction in number of searches in the interpolation domain that have local search capability.

The Schwefel function has optimal solutions separated from the sub-optimal solutions, and finding the turning point in the optimal solution was easy by performing a global search. The accuracy of the solutions obtained by the Schwefel function in "interpolation and extrapolation search" was better because of the extensive search in the extrapolation domain that has global search capability.

From the above observations, we consider the solution accuracy of the proposed algorithm to be better than those of SA/AAN in multi-peak functions shown in Fig. 3 because of the effective search in the extrapolation domain.



(a) Rastrigin function (b) Schwefel function

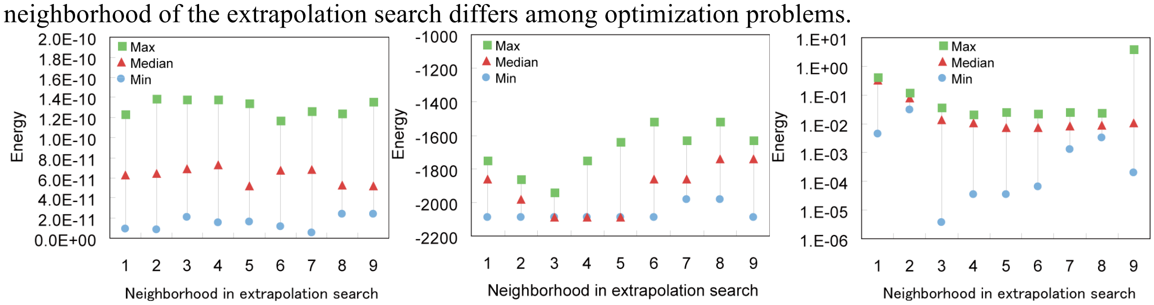
Fig. 5 Distribution of optimum solutions in
"interpolation and extrapolation search" and "interpolation and random search"

Table II. Search numbers in interpolation domain and extrapolation domain

	Rastrigin		Schwefel	
	Search number in interpolation	Search number in extrapolation	Search number in interpolation	Search number in extrapolation
Interpolation and extrapolation search	174100	125900	161880	138120
interpolation and random search	316963	3037	316479	3521

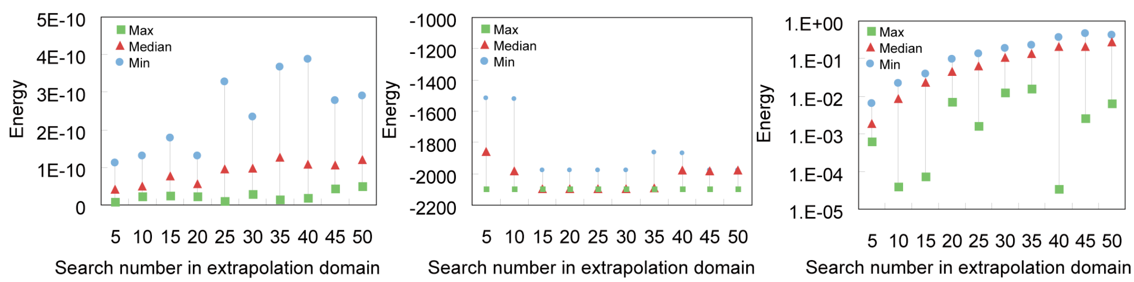
6.3 Extrapolation search parameters

In this section, we consider the parameters of extrapolation search: "Search number in extrapolation domain" and "Neighborhood in extrapolation search." The average, maximum, and minimum evaluated values of 20 trials are shown in Fig. 6 when we changed the parameter "search number in extrapolation domain" from 5 to 50.The solution accuracy of the proposed algorithm decreased as search number in extrapolation domain increased. This was due to a reduction in the search number in the interpolation domain because of the increased search number in the extrapolation domain. In the Schwefel function, the solution accuracy of the proposed algorithm was poorer with extrapolation search number below 10. This was because the proposed algorithm cannot find the turning point of optimal solution because the algorithm cannot perform a sufficient extrapolation search. On the other hand, the solution accuracy of the proposed algorithm with extrapolation search number of more than 15 was better. Therefore, it was efficient for functions with optimal solutions separated from sub-optimal solutions, such as the Schwefel function, to generate many extrapolation search points.The average, maximum, and minimum evaluated values of 20 trials are shown in Fig. 7 when we changed the parameter "Neighborhood in extrapolation search" from 1 to 9. Solution accuracy was related to "Neighborhood in extrapolation search" in the Rastrigin function as shown in Fig. 7. In addition, the solution accuracy of the proposed algorithm with a change in the neighborhood parameter from 5 to 3 was better in the Schwefel function (Fig. 7). In addition, the solution accuracy improved in the Rosenbrock function as the neighborhood increased (Fig. 7). Therefore, the neighborhood of the extrapolation search differs among optimization problems.



(a) Rastrigin function (b) Schwefel function (b) Rosenbrock function

Fig. 6 Results with changes in neighborhood in the extrapolation domain



(a) Rastrigin function (b) Schwefel function (b) Rosenbrock function

Fig. 7 Results with changes in search number in the extrapolation domain

7. CONCLUSION

In this paper, we proposed an SA with a mechanism of interpolation and extrapolation crossover. To verify the availability of the proposed algorithm, we performed comparisons with SA/AAN. The results indicated that functions with correlations between design variables are effective to search design variables in all domains.

REFERENCES

1. H. Sakuma and M. Yamamura. Extrapolation-directed crossover for real-coded GA: overcoming deceptive phenomena by extrapolative search, Proceedings of Congress on Evolutionary Computation, pp 553–560, 2001.
2. Mitsunori Miki, Tomoyuki Hiroyasu, Keiko Ono. Simulated Annealing with Advanced Adaptive Neighborhood, Journal of Information Processing Society of Japan, Vol.44, No.1, pp 1-6, 2003.