Parallel Simulated Annealing with Adaptive Neighborhood Determined by GA*

Mitsunori Miki, Tomoyuki Hiroyasu, Toshihiko Fushimi
Knowledge engineering Dept., Doshisha University
1-3 Tatara Miyakodani, Kyo-tanabe, Kyoto, 610-0321, JAPAN
mmiki@mail.doshisha.ac.jp

Abstract – *Simulated annealing (SA) is an effective general heuristic method for solving many optimization problems. This paper deals with the two problems in SA. One is the long computational time of the numerical annealings, and the solution to it is the parallel processing of SA. The other one is the determination of the appropriate neighborhood range in SA, and the solution to it is the introduction of an adaptive mechanism for changing the neighborhood range. The multiple SA processes are performed in multiple processors, and the neighborhood range in the SA processes are determined by a genetic algorithms. The proposed method is applied to solve many continuous optimization problems, and it is found that the method is very useful and effective.*

Keywords: Simulated Annealing, Genetic Algorithm, Adaptive Neighborhood Range, Continuous Optimization Problems.

1 Introduction

There is a strong incentive to parallelize the computation for optimization problems since it requires many iterations of analysis. Especially, simulated annealing, which are very effective for solving complicated optimization problems with many optima, requires tremendous computational power. Consequently, parallelization of the method is very important.

It was Kirkpatrick et al. who first proposed simulated annealing, SA, as a method for solving combinatorial optimization problems[3]. It is reported that SA is very useful for several types of combinatorial optimization problems and also for continuous optimization problems. One of the most advanced continuous optimization problems solved by SA is the prediction of the tertiary structure of Protein[5].

Some difficulty in using SA for continuous optimization problems exists for the determinations of an appropriate temperature schedule and neighborhood range. For discrete or combinatorial optimization problems, the neighborhood structure is uniquely determined by the generation method of a new solution from the current one and it is difficult to control. However, the neighborhood structure for continuous optimization problems is very simple, and it is easily controlled by the neighborhood range, or the scaling parameter of the search step.

For the control of the neighborhood range, Corana proposed an adaptive neighborhood mechanism, where the neighborhood range is so adjusted that the acceptance ratio is maintained to be 0.5 [1]. However, the validity of the target value of 0.5 is not certain, and Miki et al. found that the appropriate acceptance ratio for an adaptive neighborhood is 0.1-0.2 [6].

This type of adaptive neighborhood method is very effective and useful in SA for continuous optimization problems, but the target acceptance ratio should be determined experimentally. In order to reduce such preliminary experiments, a new adaptive neighborhood mechanism where the neighborhood range is optimally determined by a genetic algorithm (GA) is proposed in this paper.

2 Effect of Neighborhood Range

For continuous optimization problems, the neighborhood range in SA has a significant effect on the accuracy of the solutions. In order to find this effect, some numerical experiments were carried out with various fixed neighborhood range. The parameters of the experiments are shown in Tables 1 and 2. The distribution for generating neighbors is uniform in this experiment. That is, a neighbor is generated by the following equation.

$$x_{next} = x_{current} + rR, \quad (-1 \leq r \leq 1) \tag{1}$$

where r is a random value and R means the neighborhood range.

Figure 1 shows the effects of the fixed neighborhood range on the energy of the optimum solutions for the Rastrigin [9], Griewank [9] and the Rosenbrock functions [9], which are typical ones among standard mathematical test functions for continuous optimization problems. It can be seen from these results that the neighborhood range has a significant effect on the perfor-

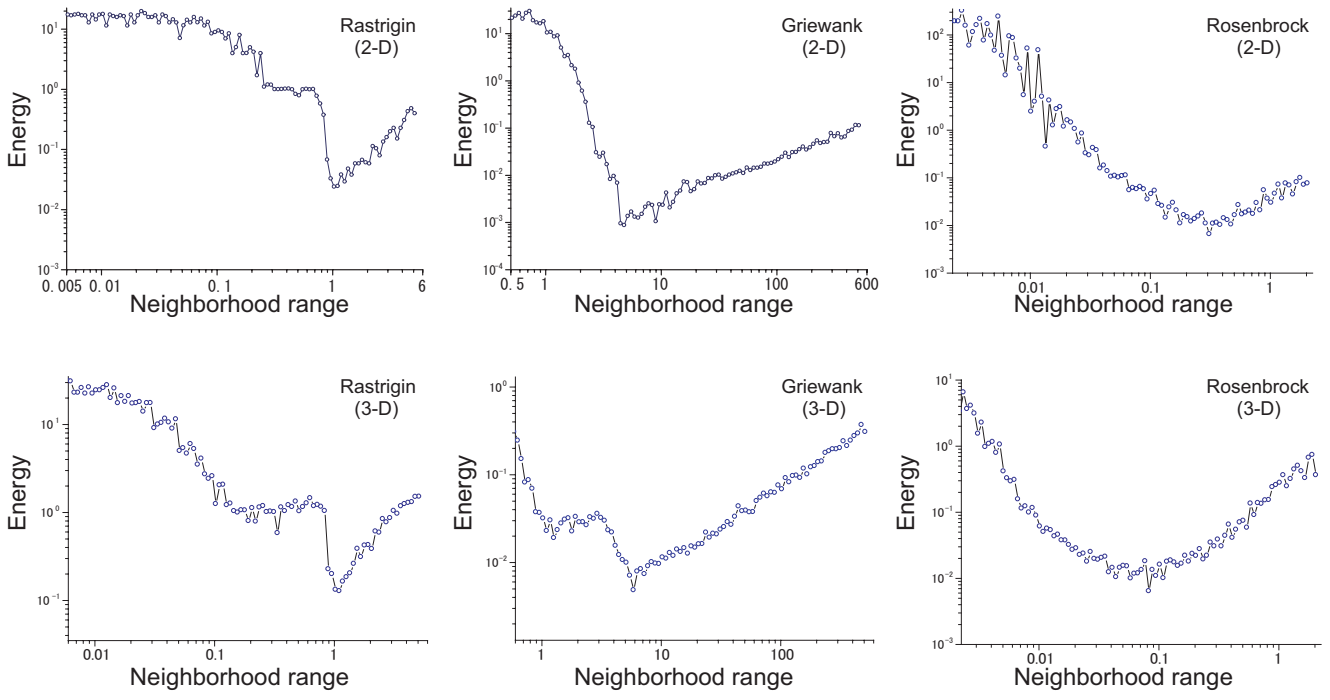

Figure 1: Effect of neighborhood range (n-D means n dimensional)

Table 1: Parameters used (Rastrigin, Griewank)

Function	Rastrigin	Griewank
Max.(Initial) temperature	10.0	20.0
Min.(Final) temperature	0.01	0.001
Cooling rate	0.8	0.726
Markov length	10240	30720

Table 2: Parameters used (Rosenbrock)

Function	Rosenbrock
Max.(Initial) temperature	1.0
Min.(Final) temperature	0.001
Cooling rate	0.81
Markov length	307

mance of SA. For the Rastrigin function, the appropriate neighborhood range is 1.0 in 2 and 3 dimensional variable spaces. For the Griewank function, the approproate neighborhood range is around 5.0 in 2 and 3 dimensional variable spaces. That is, the appropriate neighborhood range depends on problems to be solved. Furthermore, the appropriate neighborhood range also depends on the dimension of problems. For the Rosenbrock functions the appropriate neighborhood range is around 0.3 in 2 dimensional variable space and around 0.08 in 3 dimensional variable space.

From these results, it is found that the neighborhood range has very large effect on the accuracy of the obtained solutions and the appropriate neighborhood range is very difficult to find.

For continuous optimization problems, the neighbor is generated by using a particular distributions, such as uniform distributions, Normal distributions, and some special distributions used in the Fast Simulated Annealing (FSA) [8] and the Very Fast Simulated Annealing (VFSA) [2]. These distributions can be divided broadly into two categories, such as uniform and center-weighted, as shown in Fig. 2. The center-weighted distributions can be represented by a triangle distribution.

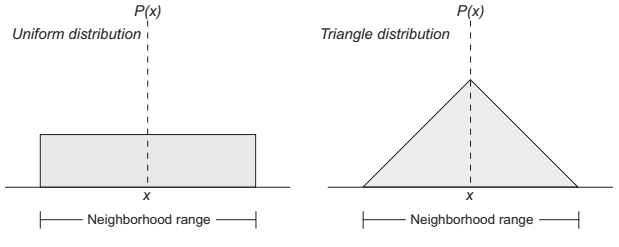

Figure 2: Typical distribution types

The effect of the distributions for generating neighbors is examined by using these two distributions. The performance of SA with these distributions is shown in Fig. 3, and the triangle distributions show a better performance for all the test functions used. It is found that the center-weighted distribution is better than the uni-

form distribution for generating neighbors. Therefore, the triangle distributions are used hereafter.

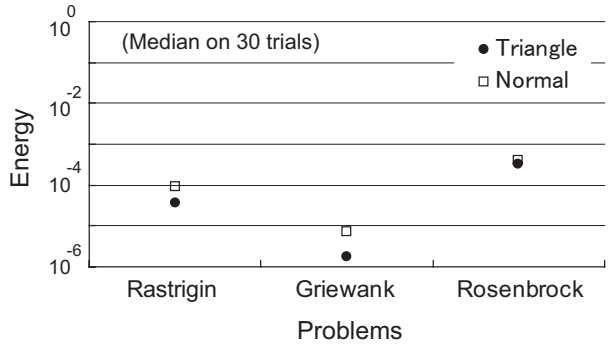

Figure 3: Effect of the distributions on the accuracy of the converged solutions

3 Parallel Simulated Annealing with Adaptive Neighborhood Determined by Genetic Algorithm

3.1 Concept of PSA/ANGA

We found that the appropriate neighborhood range in SA exists for continuous optimization problems, but such appropriate neighborhood range is problem-dependent and it is difficult to find the appropriate neighborhood range in advance. Therefore, we consider an adaptive mechanism for determining an appropriate neighborhood range for parallel SA (PSA). This method is called the parallel simulated annealing with adaptive neighborhood range determined by genetic algorithms (PSA/ANGA). Each neighborhood range of PSA is determined by a conventional GA (genetic algorithm). The schematic of the proposed method is shown in Fig. 4. It should be noted that the neighborhood range can be evolved since multiple SA processes are performed in parallel, and the population of the neighborhood ranges can be constructed.

3.2 Algorithm of PSA/ANGA

Figure 5 shows the algorithm of PSA/ANGA, where the initial neighborhood ranges are generated with random numbers and multiple SA processes start with these neighborhood ranges. After the prescribed annealing steps the neighborhood ranges are evolved by using GA operators, and new neighborhood ranges are assigned to the multiple SA processes.

The features of the method is as follows.

1. Generation, Acceptance criterion, Transition
   The generation of a new solution, the acceptance criterion, and the transition of the solution are the same as conventional SA.

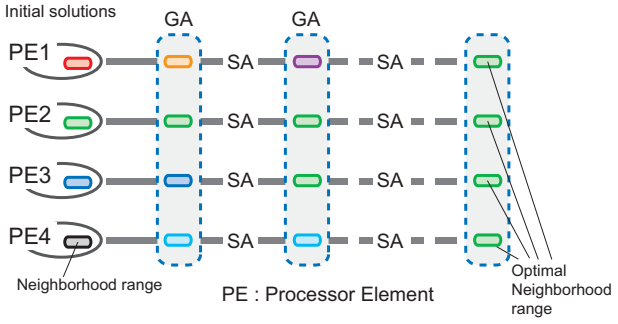

Figure 4: Schematic of the parallel SA with adaptive neighborhood range

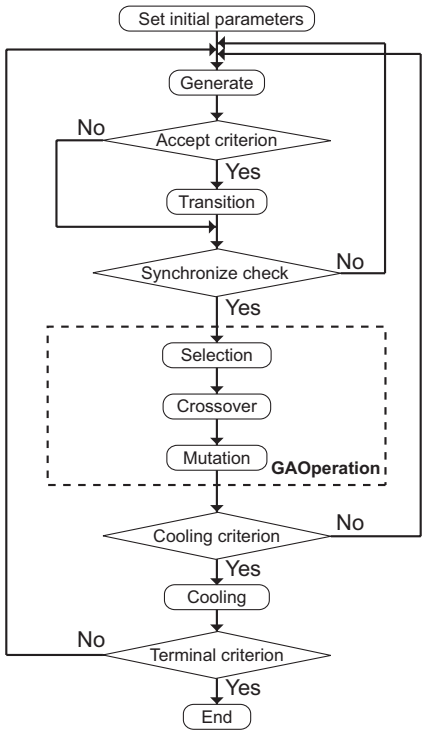

Figure 5: Algorithm of PSA/ANGA

2. Neighborhood range is determined by GA
   The synchronization of all the processes is done with a certain period, and the neighborhood ranges are determined by a GA based on the fitness value calculated from the energy of the solutions.

   The neighborhood ranges of the multiple SA processes are changed by crossover, mutation, and selection. That is, the neighborhood ranges that gives good solutions survive. Thus, all the neighborhood ranges are expected to be converged to an appropriate neighborhood range for a problem.

The coding of the neighborhood range is represented by the following equation.

$$NeighborhoodRange = MaxRange \cdot 10^{X}, \quad (-3 \leq X \leq 0) \tag{2}$$

where X is the design variable and it is a 10-bit Grey coded binary. Therefore, the range of neighborhood can be very wide.

The 1-point crossover is used and the mutation is carried out by the 1-bit flip for the 10-bit string. The crossover rate is 0.6.

3.3 Fitness value

The fitness values for the multiple neighborhood ranges for the selection of better neighborhood range are the minimum energies in multiple SA processes in the interval of the GA operations.

$$Fitness = \frac{1}{Energy} \tag{3}$$

4 Numerical Experiments

4.1 Outline of experiments

To verify the validity and the effectiveness of the proposed method, PSA/ANGA, a parallel SA with optimum fixed neighborhood range, PSA/FN, a temperature parallel SA, TPSA, and the TPSA with Corana's adaptive neighborhood, TPSA/AN, are applied to solve three typical continuous optimization problems.

The parameters used are shown in Tables 3 and 4. The number of neighborhood ranges is the same as the number of the SA processes. The maximum temperature is determined so that the acceptance rate for the worst solutions becomes 0.5 in the preliminary experiment. The minimum temperature is determined according to the required accuracy[7].

Table 3: Parameters used for PSA/ANGA (Rastrigin, Griewank)

Function	Rastrigin	Griewank
Max.(Initial) temperature	10.0	20.0
Min.(Final) temperature	0.01	0.001
Cooling rate	0.8	0.726
Markov length	102400	307200
Synchronise cycle	1600	4800
Number of processors	32	32

For the TPSA, the temperatures are determined by the conventional method mentioned in [4], and the interval for exchange solutions are the same as the temperature change interval shown in Tables 3 and 4.

Table 4: Parameters used for PSA/ANGA (Rosenbrock)

Function	Rosenbrock
Max.(Initial) temperature	1.0
Min.(Final) temperature	0.001
Cooling rate	0.81
Markov length	3072
Synchronise cycle	48
Number of processors	32

4.2 Parallel computer used

The parallel computer used is a PC cluster (Cambria cluster system) with 256 processor elements shown in Fig. 6, and 32 nodes are used for the experiments. The detail of the computer system is shown in Table 5.

Figure 6: PC cluster used

Table 5: Detail of the PC cluster used

CPU	Pentium3 800MHz(256CPU)
Memory	256MB ×256
Network	FastEthernet
OS	Debian GNU/Linux 2.4

4.3 Experimental results

Figure 7 shows a typical example of the histories of the neighborhood ranges in 32 SA processes in solving the Rastrigin function. From this figure, it is found that the appropriate neighborhood range varies between 0.2 and 2 at the beginning, but it gradually decreases to around 0.1, and it finally converges to 0.01. That is, the appropriate neighborhood range varies from large to small, while the optimum fixed neighborhood range is 1.0.

Figure 8 shows a typical history of the neighborhood range of the SA process that gives the best solution in TPSA/AN. It can be seen that this type of adaptive mechanism provides a very small neighborhood range from the beginning, and the solution is trapped into a local optimum. However, the TPSA alleviates this problem since multiple SA processes with various temperatures is running simultaneously and the solutions

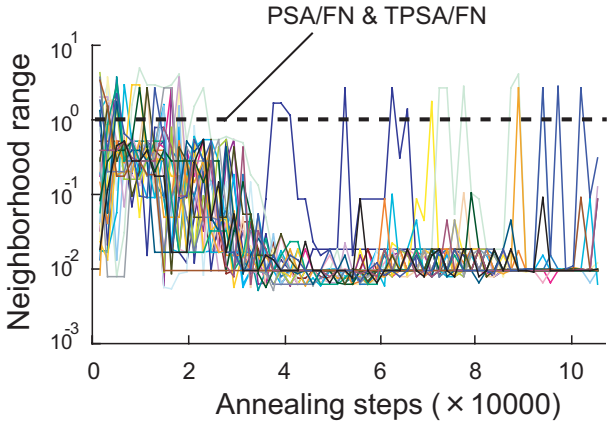

Figure 7: History of the neighborhood range in 32 SA process in solving the Rastrigin function

are interchanged among those SA processes, while single sequential SA with Corana's adaptive neighborhood mechanism provides a local optimum at all times[7].

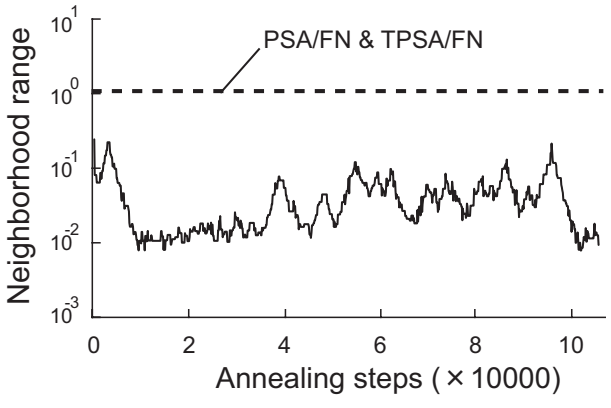

Figure 8: History of the neighborhood range of the SA process that gives the best solution in TPSA/AN

For the Rastrigin function, the minimum neighborhood range to escape a local optimum is 1.0, therefore the optimum fixed neighborhood range becomes 1.0. However, once the search region is within a global optimum region, the neighborhood range should be decreased to obtain high accuracy. The proposed method can realize it. Furthermore, the population of the neighborhood ranges has a certain diversity as shown in Fig. 7, and the escape from a local minimum is easy.

Figure 9 shows the performance of various methods. These plots represent the median of 30 trials. The reason why the median is used is that the median is very reliable for comparing the converged energy values. The converged energy values sometimes vary in a wide range and the average value is affected by one extreme value.

From this figure, neither PSA/FN nor TPSA/FN give

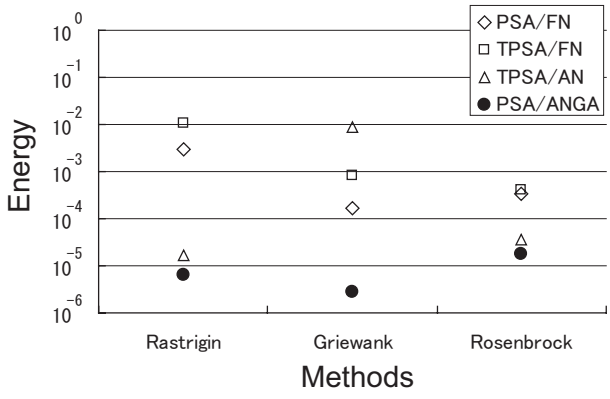

Figure 9: Comparison of the converged energies

good performance in finding the minimum energy, while the proposed method, PSA/ANGA, provides the best performance of all. On the other hand, TPSA/AN provides good performance for the Rastrigin and the Rosenbrock functions, but it does not give good performance for the Griewank function. This is due to the enormous local optima in the function.

Figure 10 shows the typical histories of the energies of 32 SA processes. The proposed method, PSA/ANGA, shows fast decrease in the energy and very low minimum value, while other methods do not. In TPSA/FN and TPSA/AN, only the best history is effective, but, the best history does not outperform the result obtained by the proposed method. It can be seen that many other SA processes except for the best process do not greatly contribute in finding the optimum solution.

Consequently, the proposed mechanism for determining the appropriate neighborhood by GA is found very effective, and PSA/ANGA can be considered to be a useful parallel SA method for continuous optimization.

It should be noted that the neighborhood range adaptation mechanism adopted here can be realized with parallel SA since the criterion for selecting " good " neighborhood range can be established from the relative value of the energies of the multiple SA processes. From this standpoint of view, parallelization of SA will give another new insight to optimization research field as well as speedup.

5 Conclusions

A new parallel simulated annealing method with adaptive neighborhood range mechanism is proposed here. The conclusions are as follows.

1. It is not easy to determine an appropriate neighborhood range for a continuous optimization problem in simulated annealing (SA).

2. The effect of the neighborhood range on the performance of SA is investigated, and the appropriate

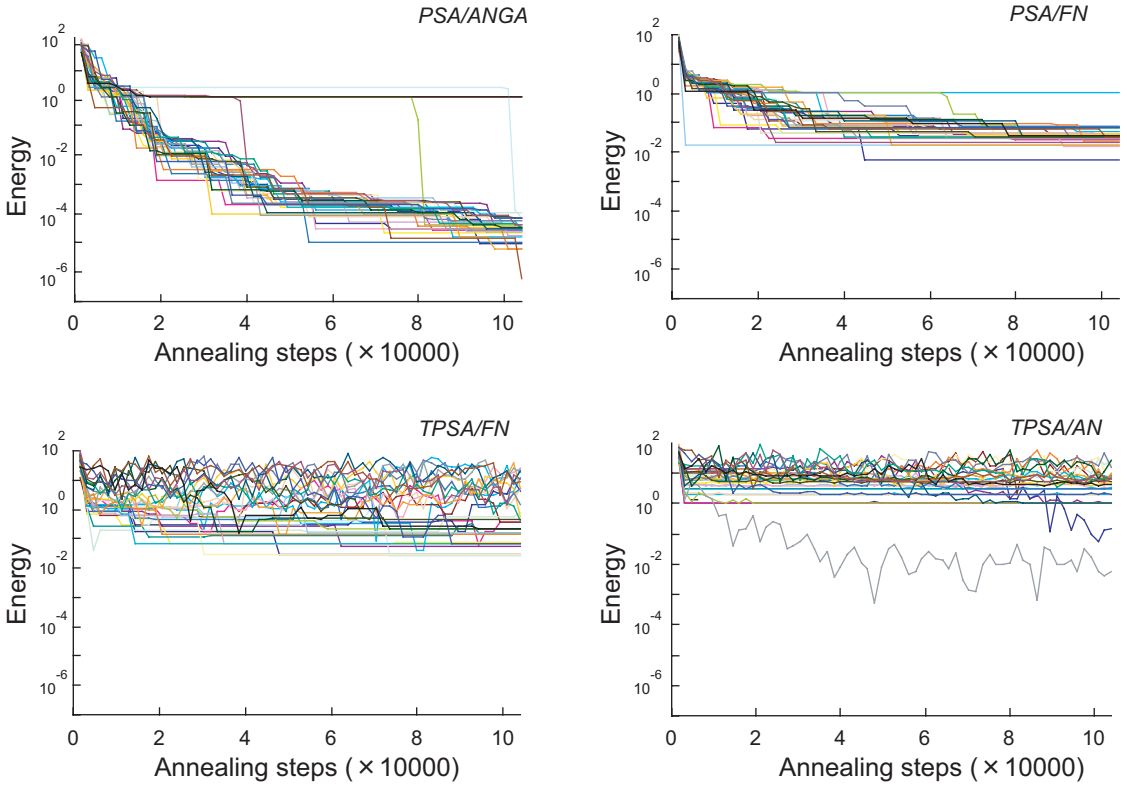

Figure 10: Typical history of energies of 32 SA processes

neighborhood range is problem-dependent, and it varies during the annealing.

3. It is found that the neighborhood range of parallel SA processes can be optimized using GA and the above characteristics.

4. A new parallel SA with the above adaptation mechanism is proposed, and the effectiveness and the usefulness of the proposed method are shown clearly for the three standard test problems. This method is called PSA/ANGA and the method is very easy to use since we do not have to determine the neighborhood range, and it gives very good solutions as well.

References

[1] Corana, A., Marchesi, M., Martini, C. and Ridella, S.:Minimiaing Multimodal Functions of Continuous Variables with the "Simulated Annealing" Algorithm, ACM Trans. Mathematical Software, Vol.13, pp.262-280,1987.

[2] Ingber, L. Genetic Algorithms and Very Fast Simulated Reannealing. A Comparison, Mathematical and Computer Modeling, 1992.

[3] Kirkpatrick, S., Gelett Jr. C.D and Vecchi, M.P.: Optimazation by Simulated Annealing, Scince, Vol.220, No.4598, pp.671-680, 1983.

[4] Konishi, K., Taki, K., and Kimura, K.: Temperature-Parallel Simulated Annealing and Its Evaluation, Information Processing Society of Japan, Vol36, No.4, pp.797-807, 1995.

[5] Minoru Kanehisa: Introduction to the genome information, Kyoritsu publishing, 1996.

[6] Mitsunori Miki, Tomoyuki Hiroyasu, Keiko Ono: Computatinal Intelligence and Applications (Proceedings of the 2nd International Workshop on Intelligent Systems Design and Applications: ISDA-02), pp.48-50, 2002.

[7] Mitsunori Miki, Tomoyuki Hiroyasu, Masayuki Kasai, Motonori Ikeuchi: Temperature Parallel Simulated Annealing with Adaptive Neighborhood for Continuous Optimization Problem, the IASTED Internationnal Conference Parallel and Distributed Computing and Systems, pp.302-316, 1999.

[8] Szu, H., Hartley, R.: Fast Simulated Annealing, Physics Letters A, 1987.

[9] Whitley,D.,Mathias,K.,Rana,S.,Dzubera,J.Evaluating Evolutionary Algorithms. Artificial Intelligence, 1996.