EVOLUTIONARY METHODS FOR DESIGN, OPTIMISATION AND CONTROL
K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou and T. Fogarty (Eds.)


# IMPLEMENTATION MODELS FOR DISTRIBUTED MEMORY ARCHITECTURE OF PARALLEL SIMULATED ANNEALING USING GENETIC CROSSOVER

**Maki Ogura, Tomoyuki Hiroyasu, Mitsunori Miki**
*Graduate School of Engineering and*
*Department of Knowledge Engineering,*
*Doshisha University*
*Kyo-tanabe, Kyoto 610-0321, Japan*
*e-mail: ogu@mikilab.doshisha.ac.jp*
*e-mail: tomo@is.doshisha.ac.jp*
*e-mail: mmiki@mail.doshisha.ac.jp*

**Yuko Okamoto**
*Department of*
*Theoretical Studies,*
*Institute for Molecular Science*
*Okazki, Aichi 444-8585, Japan*
*e-mail: okamotoy@ims.ac.jp*

**Abstract.** This paper examines implementation models for distributed memory architectures of a Parallel Simulated Annealing using Genetic Crossover (PSA/GAc). The PSA/GAc that was proposed by authors is the algorithm, where there are several processes of a simulated annealing working parallel. To exchange information between the solutions, the operation of genetic crossover is performed. We need new models to implement PSA/GAc to distributed memory architecture such as a PC cluster system, since PSA/GAc was designed only for shared memory architecture. We developed three types of implementation models of PSA/GAc. Each model was applied to a protein structure prediction problem that is one of the optimization problems. This paper makes a comparison and examination the effectiveness between the proposed models from two points of view; those are a computation time and a searching ability. Then, it is found that one of the proposed models are superior to the other models, since it can get more speed up and has high searching ability.

**Key words:** Parallel Distributed Algorithms, Simulated Annealing, Genetic Algorithm, Hybrid Algorithm, Protein Structure Prediction Problem, Optimization Problem.

## 1 Introduction

A Simulated Annealing (SA) is one of emergent calculation algorithms that solve optimization problems, and is an effective technique for solving combination optimization problems.[1] SAs have a guarantee of convergence to an optimum solution. However, SAs require huge computational costs. Specially, SA takes much time in finding an optimum solution in continuous problems. There are two solutions for this problem, performing SA in parallel and performing SA with other optimization

algorithms.

A Parallel Simulated Annealing using Genetic Crossover (PSA/GAc) is a hybrid SA using Genetic Algorithm (GA) operations and can find an optimum solution quickly compared to the conventional SA.

This paper examines implementation models of PSA/GAc for distributed memory architectures.

## 2 Parallel Simulated Annealing using Genetic Crossover

A Parallel Simulated Annealing using Genetic Crossover (PSA/GAc), proposed by authors, is a hybrid. In this algorithm, there are several processes of a simulated annealing (SA) working parallel. The flow of the concept of this algorithm is shown in Figure 1. In the proposed algorithm, there are plural processes and a sequential SA is operated in each process. After some steps, the genetic crossover is used to exchange the information between the solutions. In this paper, we call the total number of SA's search points "population size" and each SA's search point "individual".

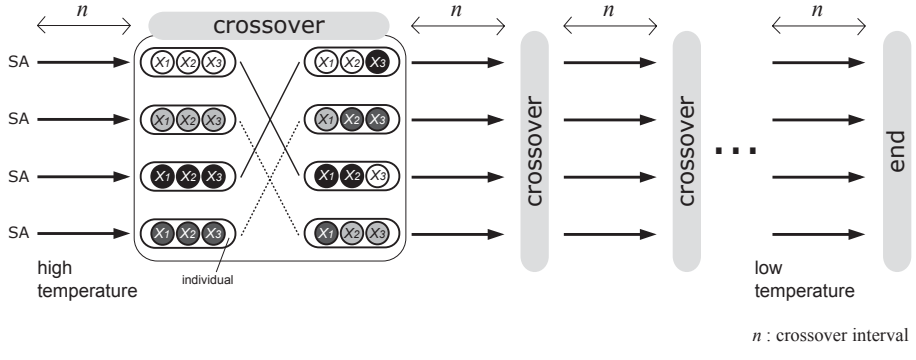


Figure 1: PSA/GAc

In this model, we select two individuals as parents randomly, and generate two children using genetic crossover when the solutions are exchanging between processes. This crossover is the only valid between the variables. Then, two individuals that have higher values of a fitness function among two parents and two children are selected. The two individuals become the next searching points. This mechanism of the operation can transmit the information of one population to the other population. To exchange information between the solutions, the operation of a genetic crossover is performed. This operation takes effect to get an optimum solution of the problem that has some minimum solutions in global and a lot of minimum solutions in local.

In the former study, PSA/GAc was applied to some test functions and we found out that the algorithm has a high searching ability. To find the effectiveness of PSA/GAc in real world problems, it was applied to a prediction of protein tertiary structure. Usually in predictions of protein tertiary structures, SAs are used for finding optimum solutions. It was also figured out that PSA/GAc could reduce a computation time and expand a searching ability compared with a conventional SA in a prediction of protein tertiary structure.[2]

## 3 Implementation Models for Distributed Memory Architecture

Since PSA/GAc that was proposed in the former study was designed only for shared memory architecture, we have to prepare new implementation models for distributed memory architectures to use the PSA/GAc on PC cluster systems. In this study, we developed three types of implementation models of PSA/GAc. We describe these models in the following sections.

### 3.1 Model 1: Dual individual model

In the model 1, the whole population is divided into sub populations called islands. In each island, there are two individuals. The islands are also divided into sub groups and each group is assigned to each processor of parallel computers. Sequential SA is operated in each island and each group. After the certain $n$ steps, the genetic crossover is performed between the two individuals in one island. After the crossover, one of the individuals is selected randomly and is moved to the other island. This operation is called "migration". This migration operation is performed synchronously for all islands. It needs communication between processors when the concerned islands are in the different processors. After the migration operation, the sequential SA is operated in each island and each group. These operations are repeated till the temperature becomes low and when the convergence condition is satisfied, the searching is terminated.

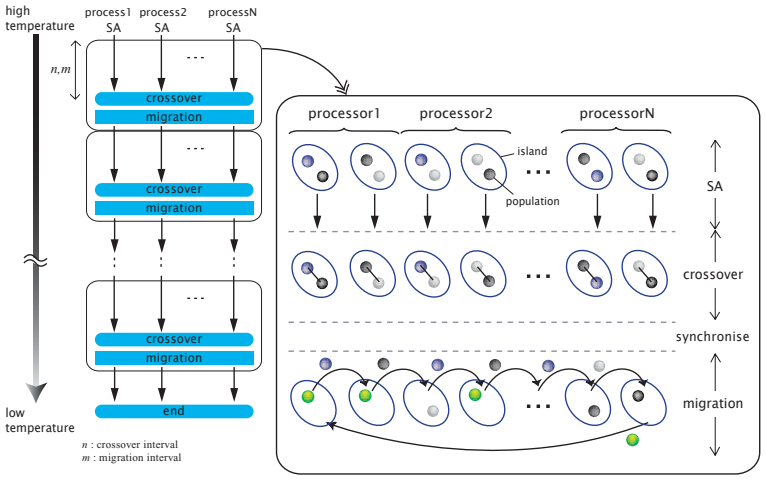


Figure 2: Model 1: Dual individual model

### 3.2 Model 2: Master slave model

In the model 2, there is a process that manages the operation of the crossover and processes that perform SA. In this model, the sequential SA is also performed for $n$ steps in each process. Then, the processes for SA send individuals to the process of the crossover. In the process for crossover, two individuals are chosen as parents and the genetic crossover is performed. The generated child is returned to the process for SA and SA is restarted. These operations are also repeated till temperature becomes low and the searching is terminated when the convergence condition is satisfied. In this paper, the operation of the crossover is performed synchronously. However, it can be also performed asynchronously in this model.

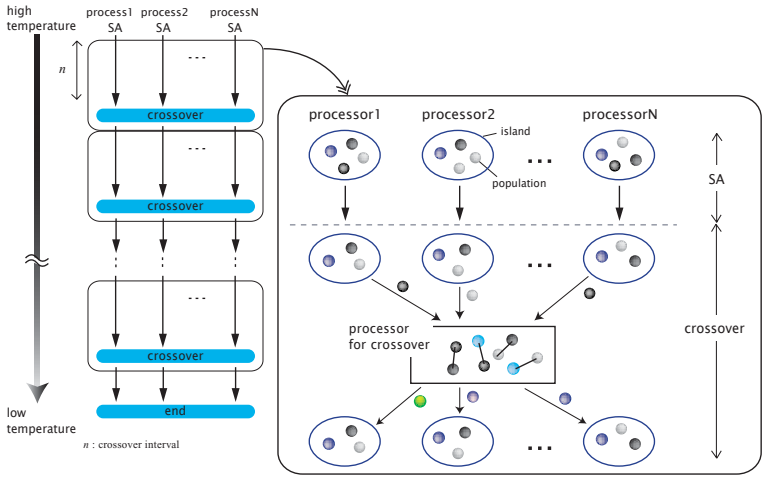


Figure 3: Model 2: Master slave model

### 3.3 Model 3: Island model

In the model 3, the whole population is divided into sub populations called islands. Sequential SA is operated in each island. After the certain $n$ steps, the genetic crossover is performed between the two individuals in one island. And, after the certain $m$ steps, the migration that selected two individuals randomly and moved them to the other island is performed. This migration operation is performed synchronously for all islands. It needs communication between processors. After the migration operation, the sequential SA is operated in each island. These operations are repeated till temperature becomes low and when the convergence condition is satisfied, the searching is terminated.

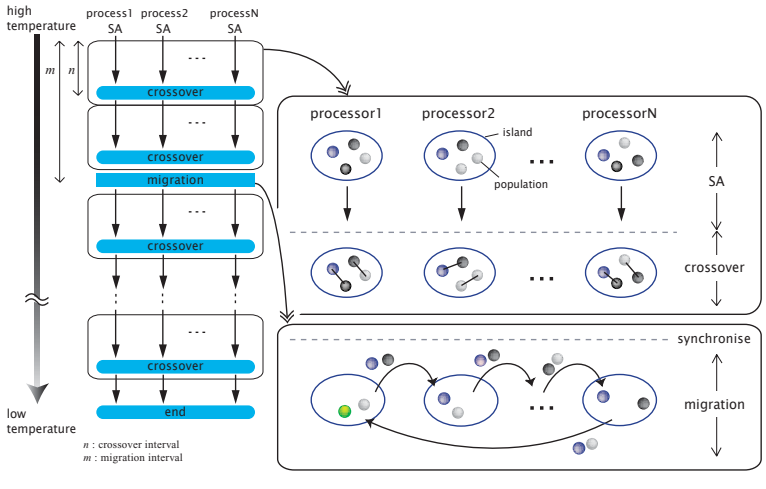


Figure 4: Model 3: Island model

## 4 Numerical Examples

### 4.1 Environment of Examples

To discuss the characteristics and the effectiveness of the proposed three models, the numerical experiments were performed. These experiments were carried out on IBM RS/6000 SP. We used 10 processors and SP switch for communication medium.

### 4.2 Prediction of Protein Tertiary Structure

Protein structure stays in the state where the energy of protein is minimum. Therefore, the predictions of protein tertiary structures can formulate as optimization problems. Usually, SA is used in predictions of protein tertiary structures. Okamoto et al. illustrated that the SA could find the optimum solution for small predictions of protein tertiary structures such as Met-enkephalin.[3] We also applied PSA/GAc to find an optimum solution of Met-enkepalin and found that this algorithm has higher searching ability compared with conventional SAs.[2]

In this numerical example, the proposed models of PSA/GAc were applied to Met-enkephalin to find the searching ability and the computation time. Met-enkephalin has 19 design variables.

The parameters that are given in Table 1 are used in this numerical example.

Table 1: parameter

| Parameter | Value |
|---|---|
| Population size | 24 |
| MCsweeps | 4000 |
| Initial temperature | 2.0 |
| Cooling rate | 0.999 |
| Crossover interval | 32 |
| Migration interval | 32 |

### 4.3 Results

In Figure 5, the searching ability of each model is shown. The searching ability can be measured by a success rate. The success rate means the probability of finding the optimum structure through the trial times. In this paper, we tried 40 times per each model. In Figure 6, the computation time of each model is shown with respect to the number of the used processors. Since the model 1 must have two individuals in each island and 24 individuals are used for searching, the trial with 8 processors cannot be executed.

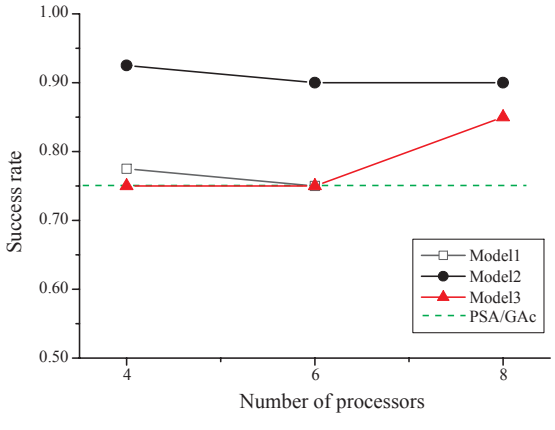


Figure 5: Success rate

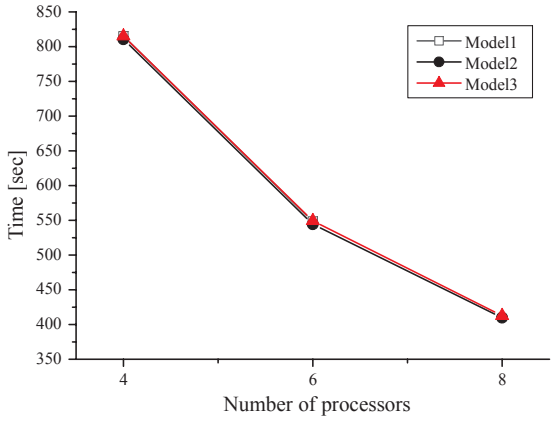


Figure 6: Computation time

From Figure 5, it is obvious that the searching ability of model 2 is higher than that of model 1 and model 3. The searching ability of all implementation models is equal to or higher than one processor model of PSA/GAc. From Figure 6, each model has a high parallel efficiency and there is few differences among models.

## 5 Conclusion

This study examines implementation models of PSA/GAc for distributed memory architecture from the side of a searching ability and a computation time. We prepared three kinds of models, these models were applied for solving the prediction of protein tertiary structure. We described that the model 2 is the best model because its searching ability is higher and a computation time is shorter. At the same time, model 2 can find an optimum solution with near 100% probability. We consider that model 2 can solve much large-scale predictions of protein tertiary structures.

## REFERENCES

[1] E. H. L. Aarts and J. H. M. Korst. *Simulated Annealing and Boltzmann Machines.* John Wiley & Sons, (1989).

[2] Maki OGURA, Tomoyuki HIROYASU, Mitsunori MIKI, Michiko SUMI, and Yuko OKAMOTO. Examination of Parallel Simulated Annealing using Genetic Crossover. *IPSJ SIG Notes,* Vol. 2001, No. 27, pp.57-60, (2001).

[3] Yuko Okamoto, Takeshi Kikuchi, and Hikaru Kawai. Prediction of Low-Energy Structures of Met-Enkephalin by Monte Carlo Simulated Annealing. *CHEMISTRY LETTERS,* pp.1275-1278, (1992).