

Effects of Using Two Neighborhood Structures in Cellular Genetic Algorithms for Function Optimization

Hisao Ishibuchi, Tsutomu Doi, and Yusuke Nojima

Department of Computer Science and Intelligent Systems, Graduate School of Engineering,
Osaka Prefecture University, 1-1 Gakuen-cho, Naka-ku, Sakai, Osaka 599-8531, Japan
hisaoi@cs.osakafu-u.ac.jp, doi@ci.cs.osakafu-u.ac.jp
nojima@cs.osakafu-u.ac.jp
http://www.ie.osakafu-u.ac.jp/~hisaoi/ci_lab_e

Abstract. We implement a cellular genetic algorithm with two neighborhood structures following the concept of structured demes: One is for interaction among individuals and the other is for mating. The effect of using these two neighborhood structures on the search ability of cellular genetic algorithms is examined through computational experiments on function optimization problems. Experimental results show that good results are obtained from the combination of a small interaction neighborhood and a large mating neighborhood. This relation in the size of the two neighborhood structures coincides with many cases of biological evolution in nature such as plants and territorial animals. It is also shown that the search ability of cellular genetic algorithms is deteriorated by the opposite combination of the two neighborhood structures.

1 Introduction

Cellular algorithms are one of the most popular models of spatially structured evolutionary algorithms [2], [3]. Since early studies in the late 1980s [6], [12] and the early 1990s [16], [17], cellular algorithms have been an active research area in the field of evolutionary computation (i.e., see [1]-[3], [5]). In cellular algorithms, each individual is spatially fixed in a cell of a lattice (typically a two-dimensional grid-world). A new offspring in a cell is generated from individuals in its neighboring cells. The main feature of cellular algorithms is the use of local selection based on a neighborhood structure. It was shown in the literature [7], [13], [14] that the size of the neighborhood structure has a large effect on the behavior of cellular algorithms.

A single neighborhood structure has been usually used in cellular algorithms in the literature. There are, however, many cases where biological evolution is based on two different neighborhood structures. For example, most plants have two neighborhood structures. Neighboring plants fight with each other for water and sunlight in an interaction neighborhood, which is much smaller than a mating neighborhood where they can disperse their pollen. Another example is territorial animals. The interaction neighborhood (i.e., territory) of a territorial animal is much smaller than its mating neighborhood. Evolution of altruism in the two neighborhood structures was actively studied under the name of structured demes in the late 1970s [4], [15], [18], [19].

The effect of the two neighborhood structures on the evolution of cooperative behavior was examined in spatial Iterated Prisoner’s Dilemma (IPD) games [8]-[11]. In [9]-[11], an individual in each cell played against only its neighbors in the interaction neighborhood. A new individual in each cell was generated from its neighbors in the mating neighborhood. It was shown under random pairing that cooperative behavior was evolved only when the interaction neighborhood was very small and the mating neighborhood was small.

In this paper, we examine the effect of using the two neighborhood structures on the search ability of cellular genetic algorithms through computational experiments on function optimization problems. In Section 2, we implement a cellular genetic algorithm following the concept of structured demes. In our cellular genetic algorithm, a rank is assigned to each individual according to the ranking of its fitness among its neighbors in the interaction neighborhood. The selection of parents for generating a new individual in each cell is performed in the mating neighborhood. In Section 3, we examine the effect of using the two neighborhood structures through computational experiments using two-dimensional grid-worlds of various sizes. Experimental results show that good results are obtained when the mating neighborhood is larger than the interaction neighborhood. Finally we conclude this paper in Section 4.

2 Cellular Genetic Algorithms with Two Neighborhood Structures

In this section, we implement a cellular genetic algorithm with two neighborhood structures following the concept of structured demes [4], [15], [18], [19]. Our cellular genetic algorithm is similar to the spatial IPD model in [8]-[11].

2.1 Two-Dimensional Grid World with Two Neighborhood Structures

We use a two-dimensional grid-world where a single individual is spatially fixed in each cell. Thus the number of cells is the same as the number of individuals as in other studies on cellular genetic algorithms. We assume the torus structure of the two-dimensional grid-world. In computational experiments, we examine three specifications of the size of the two-dimensional grid-world: 11×11 , 21×21 and 31×31 .

As we have already explained, we use two neighborhood structures in our cellular genetic algorithm. One is for interaction among individuals. This neighborhood structure determines the neighbors against which each individual competes. We denote the interaction neighborhood of the i th cell as $N_{\text{Compete}}(i)$ in order to clearly show that the individual in the i th cell competes against its neighbors in $N_{\text{Compete}}(i)$. The other neighborhood structure is for mating. This neighborhood structure determines the neighbors from which an offspring is generated for each cell. We denote the mating neighborhood of the i th cell as $N_{\text{Select}}(i)$ in order to clearly show that parents are selected from the neighbors in $N_{\text{Select}}(i)$ to generate an offspring for the i th cell.

We show three neighborhood structures in Fig. 1. In each plot, open circles show the neighbors of the closed circle individual. In computational experiments, we examine five specifications of the neighborhood size for each of the interaction and mating

neighborhood structures: 5, 9, 25, 49 and the unstructured case. The neighborhood with 49 neighbors is defined by the 7×7 square as in Fig. 1 (b) and Fig. 1 (c). All individuals are included in the neighborhood in the unstructured case. A neighborhood structure with three neighbors is not used in this paper due to its single-dimensional property (see [9], [11] for the use of such a neighborhood structure).

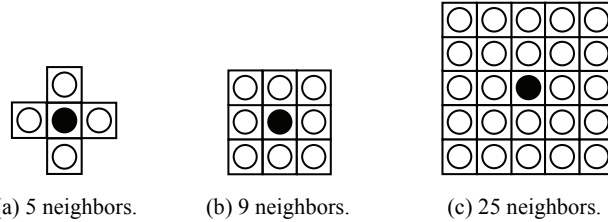


Fig. 1. Three neighborhood structures. In each plot, open circles show the neighbors of the closed circle individual. The number of the neighbors includes the closed circle individual.

2.2 Competition in the Interaction Neighborhood

In our cellular genetic algorithm, each individual competes against its neighbors in the interaction neighborhood. We assign a rank to each individual according to the ranking of its fitness among its neighbors in the interaction neighborhood.

Let us assume that our task is to find an optimal or near optimal solution of the minimization problem with the objective function $f(\mathbf{x})$. We denote the individual in the i th cell by \mathbf{x}_i . In order to assign a rank to \mathbf{x}_i , we sort the objective values of its neighbors (including \mathbf{x}_i itself) in $N_{\text{Compete}(i)}$ in ascending order. When multiple neighbors have the same objective value, we tentatively use random tiebreak to give them tentative rankings. Then a rank is assigned to \mathbf{x}_i according to the ranking of its objective value. If \mathbf{x}_i has the best objective value, rank 1 is assigned. If \mathbf{x}_i has the second best objective value, rank 2 is assigned. When multiple neighbors including \mathbf{x}_i have the same objective value, the average value of their rankings is assigned to \mathbf{x}_i . For example, when four neighbors including \mathbf{x}_i have the second best objective value, rank 3.5 is assigned to \mathbf{x}_i because their rankings are 2nd, 3rd, 4th, and 5th. In this manner, a rank is assigned to each individual according to the ranking of its objective value among its neighbors in the interaction neighborhood.

It should be noted that we do not have to sort all neighbors in $N_{\text{Compete}(i)}$. The sorting of all neighbors in the above-mentioned procedure is just for the simplicity of explanation. We only need the ranking of the objective value of \mathbf{x}_i and the number of its neighbors with the same objective value as \mathbf{x}_i in $N_{\text{Compete}(i)}$.

We handle all individuals with rank 1 as elites, which survive in their current cells to the next generation with no modifications. Such an elite is the best individual among its neighbors in the interaction neighborhood. If we use a small interaction neighborhood such as Fig. 1 (a) with five neighbors, the total number of elites is large. The number of elites decreases as the size of the interaction neighborhood increases.

2.3 Selection in the Mating Neighborhood

When the individual in the i th cell is an elite, it survives in the same cell. That is, the next generation has the same individual in the i th cell as the current generation. When the individual in the i th cell is not an elite, a new individual for the i th cell is generated from its neighbors in the mating neighborhood $N_{\text{Select}}(i)$. The standard binary tournament selection procedure is applied to the neighbors in $N_{\text{Select}}(i)$ to choose a pair of parents. A new individual for the i th cell is generated by crossover and mutation from the selected pair of parents. The point of our local selection scheme is that each individual is not evaluated by its raw objective value but its rank.

In computational experiments, we examine two versions of our local selection scheme. In the first version (Version I), two parents are selected from the neighbors as we have just explained. In the second version (Version II), the current individual in each cell is always used as one parent. Its mate is selected from its neighbors by the binary tournament selection procedure based on their ranks.

2.4 Cellular Genetic Algorithm

The outline of our cellular genetic algorithm can be written as follows:

- Step 1: Randomly generate an initial population. A single individual is located in each cell of the two-dimensional grid-world.
- Step 2: Assign a rank to each individual according to the ranking of its objective value among its neighbors in the interaction neighborhood. Individuals with rank 1 are handled as elite individuals.
- Step 3: Replace the non-elite individual in each cell with an offspring generated by selection, crossover and mutation from its neighbors in the mating neighborhood. Elite individuals stay in their current cells with no modifications.
- Step 4: Return to Step 2 if the prespecified stopping condition is not satisfied.

In computational experiments, the total number of generations is used as the stopping condition. The execution of our cellular genetic algorithm on each test problem is terminated at the 500th generation.

3 Computational Experiments

In this section, we examine the effect of using the two neighborhood structures on the search ability of our cellular genetic algorithm through computational experiments.

3.1 Conditions of Computational Experiments

As test problems, we use the following five minimization problems. The number of decision variables (i.e., n) is always specified as 10 in all the five test problems. That is, our test problems are 10-dimensional minimization problems.

F1 (Sphere Function):

$$f_1(x_j | 1 \leq j \leq n) = \sum_{j=1}^n x_j^2, \quad x_j \in [-5.12, 5.11]. \quad (1)$$

F2 (Rastrigin Function):

$$f_2(x_j | 1 \leq j \leq n) = 10n + \sum_{j=1}^n [x_j^2 - 10 \cos(2\pi x_j)], \quad x_j \in [-5.12, 5.11]. \quad (2)$$

F3 (Schwefel Function):

$$f_3(x_j | 1 \leq j \leq n) = 418.9829n + \left[\sum_{j=1}^n \left(-x_j \sin(\sqrt{|x_j|}) \right) \right], \quad x_j \in [-512, 511]. \quad (3)$$

F4 (Griewangk Function):

$$f_4(x_j | 1 \leq j \leq n) = \left[\sum_{j=1}^n \frac{x_j^2}{4000} \right] - \left[\prod_{j=1}^n \cos \frac{x_j}{\sqrt{j}} \right] + 1, \quad x_j \in [-512, 511]. \quad (4)$$

F5 (Rosenbrock Function):

$$f_5(x_j | 1 \leq j \leq n) = \sum_{j=1}^{n-1} \left[100(x_{j+1} - x_j^2)^2 + (1 - x_j)^2 \right], \quad x_j \in [-2.048, 2.047]. \quad (5)$$

Using a gray code, each decision variable x_j is coded as a binary string of length 10 (F1, F2, F3 and F4) or 12 (F5). We examine three specifications of the grid-world: 11×11 , 21×21 and 31×31 . In each grid-world, we examine five specifications of the neighborhood structure: 5, 9, 25 neighbors in Fig. 1, 49 neighbors in the 7×7 square, and the unstructured case. These five specifications are used for interaction and mating. That is, we examine 25 combinations of the two neighborhood structures.

Our cellular genetic algorithm is applied to each test problem using the following parameter specifications:

Population size: 121 (11×11), 441 (21×21), 961 (31×31),

Crossover probability (One-point crossover): 1.00,

Mutation probability (Bit-flip mutation): 0.05,

Termination condition: 500 generations.

3.2 Experimental Results: Version I

In this subsection, we report experimental results using our Version I algorithm where two parents are selected from the mating neighborhood to generate an offspring for each cell by crossover and mutation. In the next subsection, we report experimental results by our Version II algorithm where the current individual in each cell is used as one parent and its mate is selected from the mating neighborhood.

We examine the 25 combinations of the two neighborhood structures for the three grid-worlds. That is, we examine 75 different settings. Using each setting, we apply our Version I algorithm to each test problem 100 times. In each run, we record the best (i.e., smallest) objective value over all the examined individuals. Then the average of the best objective values is calculated over 100 runs.

Experimental results are shown in Figs. 2-6. In each figure, the left, center and right plots are results with the 11×11 , 21×21 and 31×31 grid-worlds, respectively. In each plot, 121, 441 and 961 denotes the unstructured case (i.e., the total number of cells). The diagonal five bars from the bottom-left to the top-right corner of each plot correspond to the cases where the two neighborhood structures are the same (e.g., $|N_{\text{Compete}}(i)| = |N_{\text{Select}}(i)| = 5$ at the bottom-left corner). On the other hand, the other 20 off-diagonal bars correspond to the cases with two different neighborhood structures. In Figs. 2-5, we can observe the improvement in the search ability of our cellular genetic algorithm by the use of a small interaction neighborhood $N_{\text{Compete}}(i)$ and a large mating neighborhood $N_{\text{Select}}(i)$. On the contrary, the search ability of our cellular genetic algorithm is degraded by the opposite combination: a large interaction neighborhood $N_{\text{Compete}}(i)$ and a small mating neighborhood $N_{\text{Select}}(i)$. In Fig. 6, the effect of using the two neighborhood structures is not clear. The search ability is somewhat degraded around the bottom-right corner in Fig. 6 (especially Fig. 6 (c)).

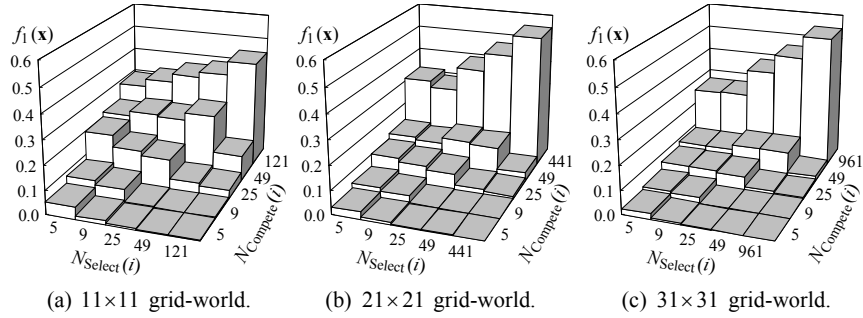


Fig. 2. Experimental results by our Version I algorithm on F1 (Sphere Function).

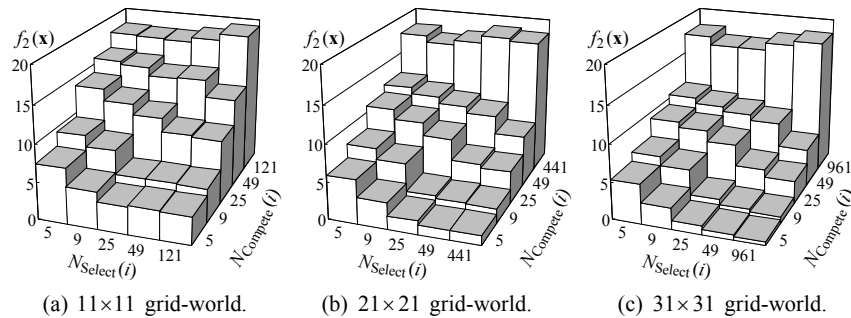


Fig. 3. Experimental results by our Version I algorithm on F2 (Rastrigin Function).

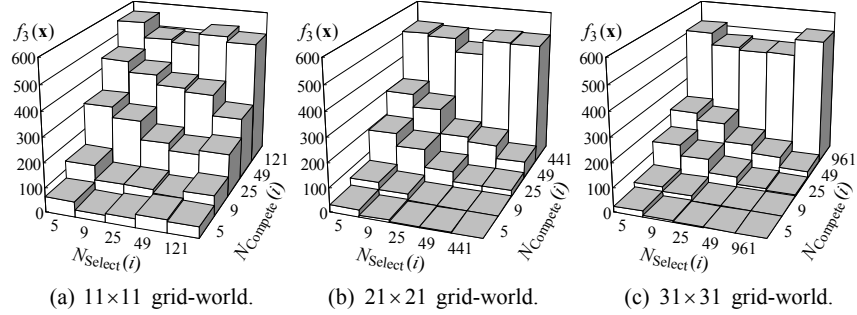


Fig. 4. Experimental results by our Version I algorithm on F3 (Schwefel Function).

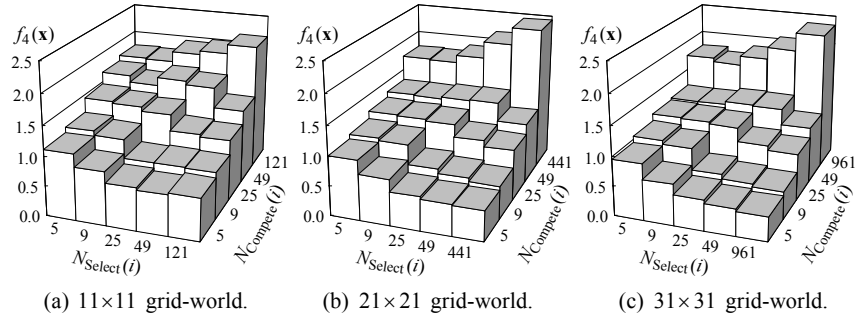


Fig. 5. Experimental results by our Version I algorithm on F4 (Griewangk Function).

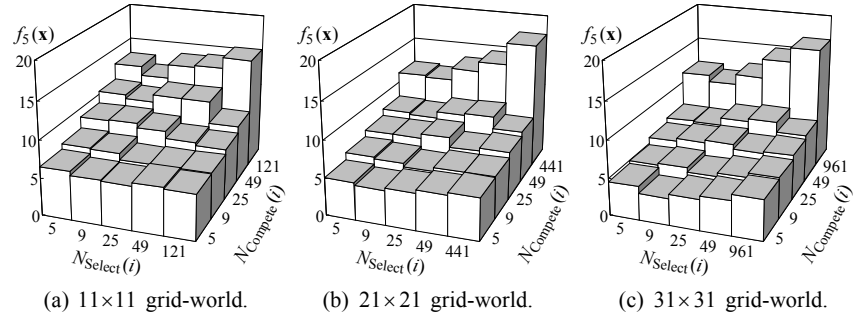


Fig. 6. Experimental results by our Version I algorithm on F5 (Rosenbrock Function).

3.3 Experimental Results: Version II

In this subsection, we report experimental results using our Version II algorithm. In the same manner as Figs. 2-6, average results over 100 runs are shown in Figs. 7-11. From the comparison between Figs. 2-6 and Figs. 7-11, we can see that our Version II algorithm is inferior to our Version I algorithm (the scale of the vertical axis is not the

same between the corresponding figures, e.g., see Fig. 2 and Fig. 7). Nevertheless, we can obtain almost the same observation from Figs. 7-11 as Figs. 2-6. That is, the search ability of our cellular genetic algorithm is improved around the bottom-right corner in Figs. 7-11 by the use of a small interaction neighborhood $N_{Compete}(i)$ and a large mating neighborhood $N_{Select}(i)$, and degraded by the opposite combination: a large interaction neighborhood and a small mating neighborhood.

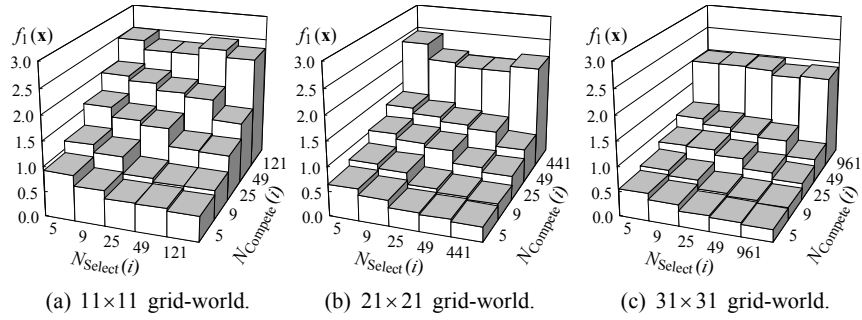


Fig. 7. Experimental results by our Version II algorithm on F1 (Sphere Function).

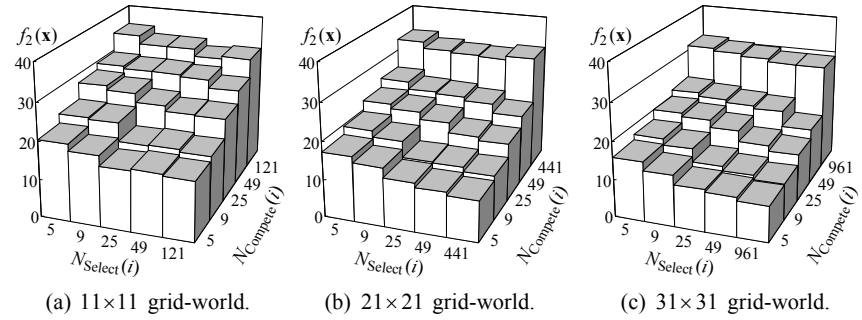


Fig. 8. Experimental results by our Version II algorithm on F2 (Rastrigin Function).

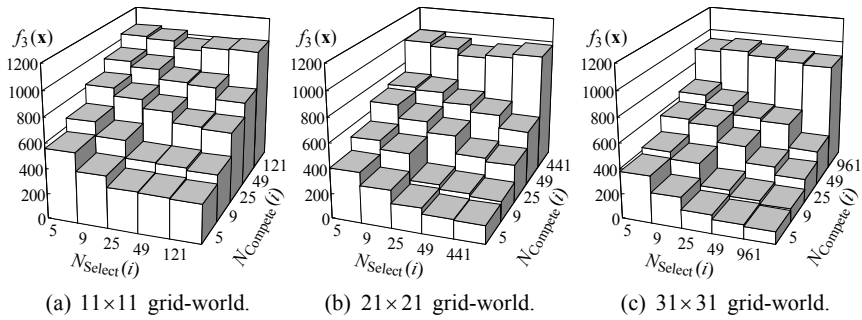


Fig. 9. Experimental results by our Version II algorithm on F3 (Schwefel Function).

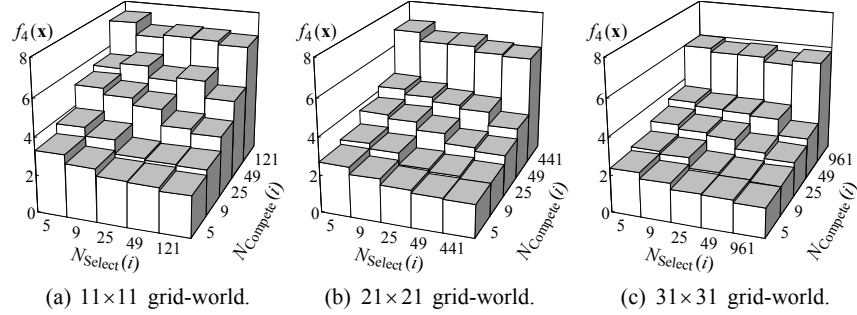


Fig. 10. Experimental results by our Version II algorithm on F4 (Griewangk Function).

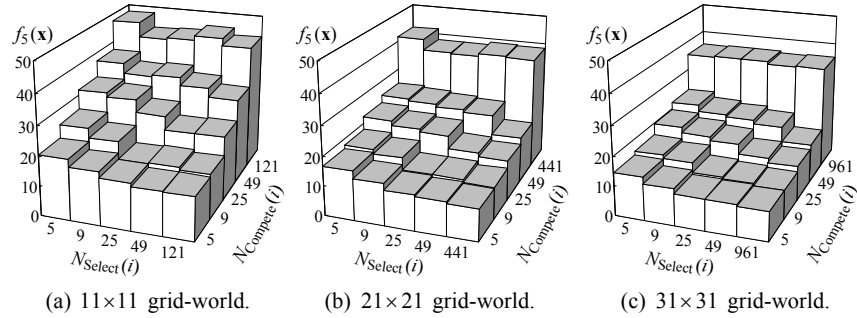


Fig. 11. Experimental results by our Version II algorithm on F5 (Rosenbrock Function).

4 Conclusions

In this paper, we examined the effect of using two neighborhood structures on the search ability of cellular genetic algorithms: One is for interaction among individuals and the other is for mating. We first implemented a cellular genetic algorithm with the two neighborhood structures following the concept of structured demes. Then we examined the effect of the two neighborhood structures through computational experiments on function optimization problems. Experimental results showed that the search ability of our cellular genetic algorithm was improved by using a small interaction neighborhood and a large mating neighborhood. Such a combination of the two neighborhood structures coincides with many cases in nature such as plants and territorial animals. It was also shown that the opposite combination degraded the search ability of our cellular genetic algorithm. In our current implementation, the number of elite individuals depends on the size of the interaction neighborhood. In future work, we will compare different specifications of the neighborhood size under the same condition with respect to the number of elite individuals.

This work was partially supported by Japan Society for the Promotion of Science (JSPS) through Grand-in-Aid for Scientific Research (B): KAKENHI (17300075).

References

1. Alba, E., Dorronsoro, B.: The Exploration/Exploitation Tradeoff in Dynamic Cellular Genetic Algorithms. *IEEE Trans. on Evolutionary Computation* 9 (2005) 126-142
2. Alba, E., Tomassini, M.: Parallelism and Evolutionary Algorithms. *IEEE Trans. on Evolutionary Computation* 6 (2002) 443-462
3. Cantu-Paz, E.: *Efficient and Accurate Parallel Genetic Algorithms*. Springer, Berlin (2000)
4. Charlesworth, B.: A Note on the Evolution of Altruism in Structured Demes. *The American Naturalist* 113 (1979) 601-605
5. Giacobini, M., Tomassini, M., Tettamanzi, A. G. B., Alba, E.: Selection Intensity in Cellular Evolutionary Algorithms for Regular Lattices. *IEEE Trans. on Evolutionary Computation* 9 (2005) 489-505
6. Gorges-Schleuter, M.: ASPARAGOS: An Asynchronous Parallel Genetic Optimization Strategy. *Proc. of 3rd International Conference on Genetic Algorithms* (1989) 422-427
7. Gorges-Schleuter, M.: A Comparative Study on Global and Local Selection in Evolutionary Strategies. *Lecture Notes in Computer Science*, Vol. 1498: Parallel Problem Solving from Nature - PPSN V. Springer, Berlin (1998) 367-377
8. Ifti, M., Killingback, T., Doebeli, M.: Effects of Neighborhood Size and Connectivity on the Spatial Continuous Prisoner's Dilemma. *Journal of Theoretical Biology* 231 (2004) 97-106
9. Ishibuchi, H., Namikawa, N.: Evolution of Iterated Prisoner's Dilemma Game Strategies in Structured Demes under Random Pairing in Game Playing. *IEEE Trans. on Evolutionary Computation* 9 (2005) 552-561
10. Ishibuchi, H., Namikawa, N.: Evolution of Cooperative Behavior in the Iterated Prisoner's Dilemma under Random Pairing in Game Playing. *Proc. of 2005 Congress on Evolutionary Computation* (2005) 2637-2644
11. Ishibuchi, H., Namikawa, N., Ohara, K.: Effects of Spatial Structures on Evolution of Iterated Prisoner's Dilemma Game Strategies in Single-Dimensional and Two-Dimensional Grids. *Proc. of 2006 Congress on Evolutionary Computation* (2006) (in press)
12. Manderick, B., Spiessens, P.: Fine-Grained Parallel Genetic Algorithms. *Proc. of 3rd International Conference on Genetic Algorithms* (1989) 428-433
13. Sarma, J., De Jong, K.: An Analysis of the Effects of Neighborhood Size and Shape on Local Selection Algorithms. *Lecture Notes in Computer Science*, Vol. 1141: Parallel Problem Solving from Nature - PPSN IV. Springer, Berlin (1996) 236-244
14. Sarma, J., De Jong, K.: An Analysis of Local Selection Algorithms in a Spatially Structured Evolutionary Algorithm. *Proc. of 7th International Conference on Genetic Algorithms* (1997) 181-186
15. Slatkin, M., Wilson, D. S.: Coevolution in Structured Demes. *Proc. of the National Academy of Sciences* 76 (1979) 2084-2087
16. Spiessens, P., Manderick, B.: A Massively Parallel Genetic Algorithm: Implementation and First Analysis. *Proc. of 4th International Conference on Genetic Algorithms* (1991) 279-286
17. Whitley, D.: Cellular Genetic Algorithms. *Proc. of 5th International Conference on Genetic Algorithms* (1993) 658
18. Wilson, D. S.: Structured Demes and the Evolution of Group-Advantageous Traits. *The American Naturalist* 111 (1977) 157-185
19. Wilson, D. S.: Structured Demes and Trait-Group Variation. *The American Naturalist* 113 (1979) 606-610