

3D Neighborhood Relationships of Cellular Genetic Algorithms for the Tour Guide Assignment Problem

Lina Setiyani¹ and Takeo Okazaki²

¹ Graduate School of Engineering and Science, University of the Ryukyus 1 Senbaru / Nishihara, Okinawa 903-0213, Japan
tya.sachi@ms.ie.u-ryukyu.ac.jp

² Computer Science and Intelligent Systems, University of the Ryukyus 1 Senbaru / Nishihara, Okinawa 903-0213, Japan
okazaki@ie.u-ryukyu.ac.jp

* Corresponding Author: Takeo Okazaki

Received January 19, 2017; Revised March 20, 2017; Accepted April 6, 2017; Published June 30, 2017

* Regular Paper

* Extended from a Conference: Preliminary results of this paper were presented at the ITC-CSCC 2016. This paper has been accepted by the editorial board through the regular review process that confirms the original contribution.

Abstract: Management optimization is very important in tourism, especially when it is related to productivity. One of the problems in management optimization is tour guide assignment. Well-arranged tour guide assignment will increase productivity while maintaining service quality. A cellular genetic algorithm is one of the methods that can be used to solve this problem. Furthermore, previous study has shown that a cellular dimension increase can lead to promising benefits for certain problems. The objective of this research is to give a clear understanding of the advantages of increasing cellular dimensionality on the tour guide assignment problem by using a cellular genetic algorithm.

Keywords: 3D neighborhood, Tour guide assignment problem, Cellular genetic algorithm

1. Introduction

Good service quality with high productivity is a key factor in tourism service centers that attract more tourists and increase profits. For this reason, management optimization becomes very important for tourism. By maintaining good management optimization, a service center can provide good quality service while keeping operational costs as low as possible. One of the problems in tourism management is tour guide assignment.

In general, the assignment problem is a special type of linear programming problem that deals with assignment of various objects to various activities on a one-to-one basis in the most effective way to minimize the total assignment cost or consumed time needed to complete the tasks.

The genetic algorithm (GA) and the cellular genetic algorithm (cGA) are two methods that have been chosen and proven by researchers for solving many kinds of optimization problems in both classical and real-world settings [1].

In this research, we provide a clear understanding of the advantages of increasing neighborhood cellular

dimensionality on the tour guide assignment problem (TGAP) by using a cGA. The goal of this study is to compare the results of a 3D topology with a 2D topology in terms of efficiency and efficacy.

The rest of the paper is organized as follows. The next section describes the tour guide assignment problem; then, we explain the principle of the 3D neighborhood and how to introduce it into a cGA; after that, we show how the method was confirmed. Section 4 looks at the simulation experiment. Finally, we offer a conclusion from the results, and propose future work.

2. Tour Guide Assignment Problem

The assignment problem is usually recognized by two components of the assignments, which represent the unrevealed combinatorial structure. Each assignment problem has a table, or matrix, associated with it. In general, the rows represent the objects or people that we wish to assign, while the columns consist of the jobs or tasks that we assign to them. Consider the problem of

		Activity				
		A ₁	A ₂	...	A _n	
Resource	R ₁	C ₁₁	C ₁₂	...	C _{1n}	Available
	R ₂	C ₂₁	C ₂₂	...	C _{2n}	1
	⋮	⋮	⋮	⋮	⋮	⋮
	R _m	C _{m1}	C _{m2}	...	C _{mn}	1
Required		1	1	...	1	

Fig. 1. Matrix of Assignment Model.

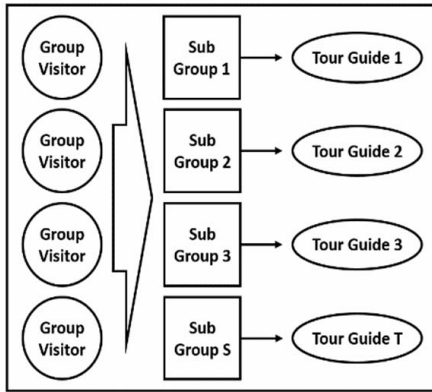


Fig. 2. The assigning of tour guides.

assigning n resources to m activities so as to minimize the overall cost or time under conditions where each resource will be assigned to handle one task only for each cycle. The cost matrix $C = (C_{ij})$ is represented in Fig. 1.

In a TGAP, the service center provides services where a number of tour guides will take different visitor groups. All guides have their own fixed guiding time to serve their group of visitors. To maintain good quality service, a guide should not be assigned to direct too many visitors at one time, especially on a weekend or holiday. To overcome this problem, when a visitor group is too large to handle, it will be divided into smaller subgroups before being served, where each subgroup will be handled by only one tour guide (Fig. 2). However, this situation will need more tour guides to handle every additional subgroup. Hiring a part time guide will be an economically reasonable solution, rather than recruiting a new full time employee. Since all part time guides will be paid based on their service time, an assignment of tour guides that provides lower total service time will be preferred. To reduce the total service time, additional preparation time should be reduced as much as possible.

From Fig. 2, we can define an evaluation function to measure the total guide time. An assignment that has the minimum total guide time will provide the best schedule:

For $i \in T, j \in S$

$$\text{Minimize } F = \sum_{i=1}^x \sum_{j=1}^y \{G_{ij}V_{ij} + p_i q_i\} \quad (1)$$

$$V_{ij} \in \{0, 1\} \forall i \in T, j \in S$$

$$q_i \in \{0, 1\} \forall i \in T$$

where G_{ij} represents actual guide time of the employees, and V_{ij} indicates a variable that takes a value of 1 for guide i when assigned to group j ; otherwise, it is 0. While p_i represents preparation time for the i^{th} tour, q_i will have a value of 1 when the service sequence number of the former subgroup is higher; otherwise, it is 0.

However, to measure performance, some modification is needed. The assignment that has the lowest total guide time will become the best schedule for assignment of subgroups to tour guides. The modified evaluation function is composed of three parts:

$$F^* = \sum_{i=1}^x \sum_{j=1}^y \{G_{ij}V_{ij} + p_i q_i + R_j o_j\}. \quad (2)$$

The first part is the actual guide time. In the second part, additional preparation time is considered. The final term in Eq. (2) is the product of $\{0,1\}$ variable o_j and the penalty function R_j , which is related to the relative importance of the visitor subgroups. The penalty function value for a more important subgroup is higher than other subgroups. The minimum total service time will be more effective and will lead to lower operational costs.

In addition, some important data are needed before running the assignment. These data can be divided into two parts:

- the visitor-related data, such as visitor group index number, names, and stay time
- the tour guide-related data, for example, service capacity and preparation time

3. 3D cGA Approach

The tour guide assignment problem itself is a combinatorial optimization problem that is NP-hard. For this reason, a heuristic approach has been applied to solve this problem. GA and cGA were previously shown and discussed to solve the TGAP within a short time [2, 3]. However, both of those studies focused on implementing the method in the 2D environment only. There is still a possibility to improve the performance of the algorithm. Here, we proposed to solve the TGAP in a 3D architecture.

Higher cellular dimensions have shown promising benefits in certain problems. Previous investigations into the 3D architecture showed improvements in the performance of the algorithm when compared with smaller grid dimensions [4, 5]. Increasing the dimensions of the neighborhood topology also gives a more efficient solution in terms of convergence [5].

Compared to other genetic algorithm methods, cGA has several advantages, such as a high diversity level, which can be maintained for much longer, in comparison to centralized ones. For that reason, we focused on solving the TGAP in a cGA.

In the cGA, the population is arranged in a special structure defined as a connected graph in a toroidal d-

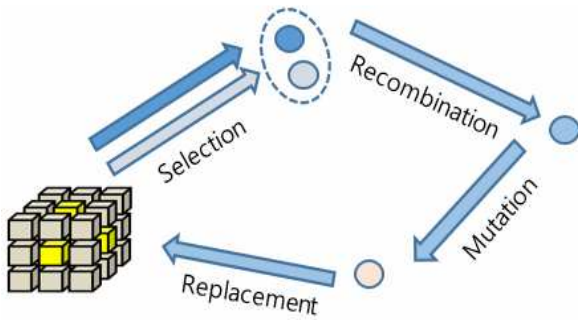


Fig. 3. General 3D CGA process.

1	1	2	2	3	3	4	4	5	5	Tour Guide ID
5	3	1	1	2	4	7	0	8	6	Sub Group ID

Fig. 4. Encoding the chromosomes.

dimension grid where each individual is restricted to communicating with only their nearest neighbors. Generally, as shown in Fig. 3, the cGA process will start by evaluating the fitness of the neighborhood genotype; then, the fittest genotype will be selected as the first parent, while the second parent is chosen from the current individuals. Through a crossover process between the first and second parents, the fittest offspring will be selected and mutated. The modified offspring are then evaluated to find the fittest value. The process will be repeated until termination.

The difference between a 2D- and a 3D-neighborhood cGA is the addition of the third dimension, which refers to the layers of the grid, with the position of individuals located on the z-axis. As a consequence, the 3D neighborhood is a denser neighborhood than one in 2D, leading to additional time for communications or computations, especially in a tournament selection system, since this system needs access to all individuals in the neighborhood.

Before we make the assignment, the important part is that we need to encode the chromosomes first. These chromosomes consist of genes that represent the tour guides and subgroups. There are chromosome limitations, which we used to make this assignment, as explained below.

1. Each tour guide can only be assigned to only one subgroup for each shift.
2. Tour guides are not allowed to handle a group of visitors exceeding their capacity.
3. Tour guides work for only 8 hours a day.
4. The number of tour guides is known.
5. Serving time and preparation time are fixed.

The chromosomes in the population include the ID number of the subgroup, as shown in Fig. 4. The repeating tour guide ID number represents the working shift: first and second. Each subgroup ID is then assigned to each tour guide for each working shift to complete the chromosomes. From this encoding process, we can assign the tour guide to the subgroup. As a result, the assignment

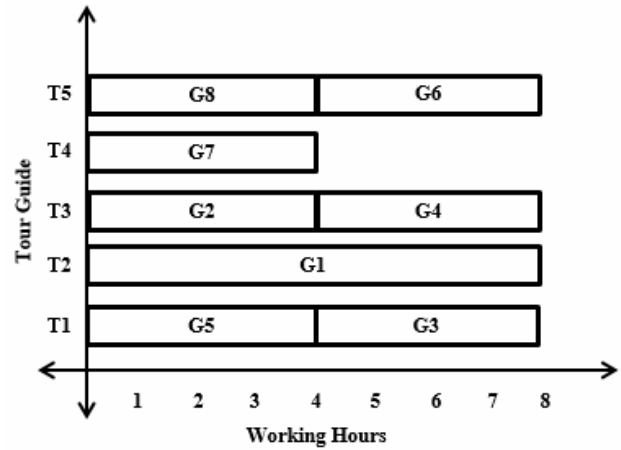


Fig. 5. Assignment Form.

	1	2	3	4	5	6
Chromosome 1	6	8	1	4	9	3
Chromosome 2	3	1	9	2	5	7
Offspring	6	8	9	2	5	3

Fig. 6. Crossover is done by a two-point crossover method.

gives the proposed schedule shown in Fig. 5.

Each tour guide will be assigned to only one subgroup in a certain time without exceeding the working hours of each tour guide.

In the crossover process, the offspring are produced through a two-point crossover method. After two points are randomly chosen, there are three fragments available in each chromosome. An offspring is then created by copying the first and third fragments from chromosome 1 with the second fragment from chromosome 2. As illustrated in Fig. 6, {6}, {8}, and {3} are copied from chromosome 1. The remaining numbers are {9}, {2}, and {5}. They are then copied to create the offspring. The procedure is quite straightforward, and it is not necessary for the number of each gene to be unique.

El Zoghby et al. [3] previously applied a cGA to a TGAP in a 2D neighborhood world. However, Sarma and De Jong found that the convergence rate of the algorithm depends on the shape and size of the grid and the neighborhood [6]; thus, increasing the dimensions of the neighborhood is expected to give a more efficient solution within a shorter time. In this research, the comparison to be learned based on a cellular genetic algorithm results when a 2D- or 3D-grid structure is applied to solve the tour guide assignment problem.

To increase the solution in a shorter computational time while keeping the quality of the solution with a 3D structure, Algorithm 1 is proposed. It starts by creating a random population (step 2), in which the fitness function of each individual will be evaluated (step 3), and it keeps updating individuals by using genetic operators until a

Algorithm 1. Pseudo code of 3D-cGA.

```

1. procedure cGA
2. Generate_initial_population(P(0));
3. Evaluation(P(0));
4. while ! stop_condition() do
5.   for i ← 1 to ROWS do
6.     for j ← 1 to COLUMNS do
7.       for k ← 1 to LAYERS do
8. Neighbors ← compute_neighbors (position(i,j,k));
9. Parent1 ← position(i,j,k);
10. Parent2 ← Local_selection (neighbors);
11. offspring ← Recombine (Pc, parent1, parent2);
12. offspring ← Mutate(Pm);
13. Evaluation ← Fitness(offspring);
14. Replace_if_better (position(i,j,k), offspring, P_aux(t));
15. end for;
16. end for;
17. P(t+1) ← P_aux(t);
18. t ← t+1;
19. end while;
20. end procedure cGA

```

termination condition is met (step 4). The current individual is then calculated, which can only interact with its neighbors (step 6). The first parent will be chosen from the current one (step 9). A binary tournament is then used as the local selection method in the neighborhood to choose the second parent (step 10). Two children with the best fitness will be delivered after one-point crossover is applied with probability P_c (step 11), followed by a bit-flip mutation with probability P_m (step 12).

The algorithm then calculates the fitness value (step 13), and replaces the current individual with the new one if it has a better fitness value (step 14). Each population then replaces the previous one on a synchronous update (step 17). The loop continues until the termination condition is met (step 4).

In the binary tournament selection, the population fitness in generation t will be distributed normally $N(\mu_{F,t}, \sigma_{F,t}^2)$. The individual with fitness F from the binary tournament ($s = 2$) will have the expected value as the tournament winner, which can be calculated as

$$\mu_{F,t+1} = \mu_{F,t} + \sigma_{F,t} \frac{1}{\sqrt{\pi}} \quad (3)$$

4. Simulation Experiment

To compare the quality of the solution and the calculation speed for the 3D neighborhood relationships in solving the TGAP, we made a comparative experiment simulation of 2D and 3D neighborhoods.

In this experiment, a conventional 2D and the proposed 3D cGA were used to solve the TGAP. A comparison of the obtained results from the two methods was done from the perspectives of solution quality and speed. All experimental data results were obtained based on 30 iterations, a two-point crossover and a two-point swapping mutation. Average, minimum, maximum, and variance

Table 1. Visitor-related Data.

	Description	Value
No. of Visitors	Number of visitors	Positive Integer
Stay Time	Time that visitors will choose for stay	4, 8 hours

Table 2. Tour Guide-related Data.

	Description	Values
Service Capacity	The tour guides have different capacities, depending on experience.	15, 20, 25
Preparation Time	This preparation time depends on service capacity of the tour guide.	1-2 hours

values were obtained from the experiment, which represent the central tendency of distribution, the best solution, the worst solution, and the stability of the algorithm, respectively.

In addition, some artificial data that are related to visitor and tour-guide data were input before the whole simulation process began. For this simulation, the number of tour guides and visitors was known. We set the data as in Tables 1 and 2.

The service capacity varied from an experienced to a beginner tour guide. The experienced guide has a greater service capacity than the others. The preparation time also varied between experienced and beginner tour guides, where the experienced tour guide requires less preparation time.

In the preliminary experiment, we try to find the most suitable condition for the TGAP. We ran simulation experiments with the same basic parameters but variations in some parameters, such as population size, crossover rate, and mutation rate, as shown in Table 3, to see the influence of these factors on the results and to calculate the best fitness over generations. The influence of crossover rate, mutation rate, and population size over the fitness values were tested to find the appropriate values of the parameters. The best assignment results can then be described in a timetable graph, as shown in Fig. 7.

A comparison of various crossover rates was done. The results showed that 3D gave a better result in almost all rates, compared to 2D, in terms of fitness value and stability, as shown by the minimum and variance values obtained, except for the 0.8 rate. In this case, although 3D has a better value for minimum fitness, the smaller variance value shows that 2D has better stability than 3D. In terms of fitness value and stability, the results show that the best case is when the crossover rate is set at 0.9 for both 2D and 3D. The variations of best fitness value from the influence of crossover rate in both 2D and 3D scenarios are shown in Table 4.

In a similar way, a comparison of mutation rates was done. The 0.9 crossover rate value from the previous experiment was chosen in this experiment in order to compare the results. Although there is almost no difference in the mutation process results from 2D and 3D at different rates, the results show that 3D still has a better fitness value and stability compared to the 2D neighborhood. Meanwhile, the best case was obtained at a rate of 0.05.

Table 3. Parameters for Simulation.

Parameters	2D-cGA	3D-cGA
Neighborhood	East, west, north, south	East, west, vertical north, south, horizontal north and south
Generation Number	100	100
Mutation Rate	0.01, 0.02, 0.03, 0.04, 0.05, 0.06	0.01, 0.02, 0.03, 0.04, 0.05, 0.06
Population Size	50, 100, 300, 400	50, 100, 300, 400
Crossover Rate	0.8, 0.9, 1.0, 1.1	0.8, 0.9, 1.0, 1.1
Parent Selection	Best neighborhood	Best neighborhood
Recombination	Two-point crossover method	Two-point crossover method
Mutation	Swap mutation	Swap mutation
Replacement	Replace if better	Replace if better

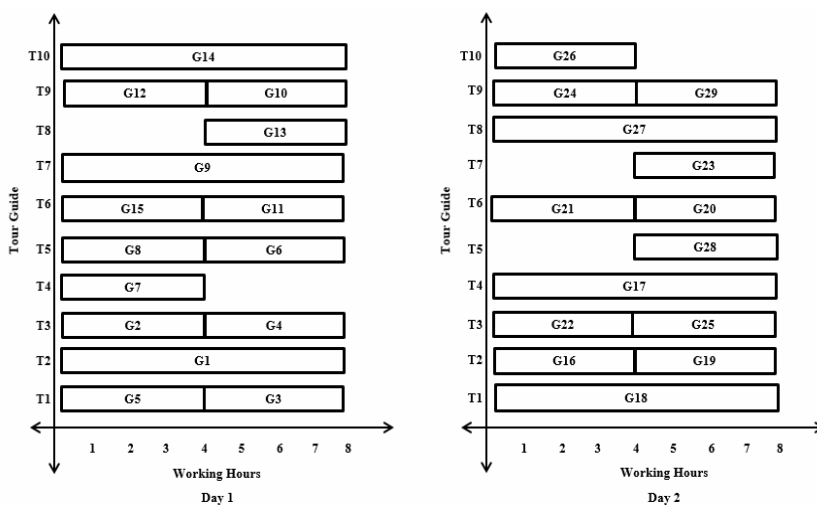


Fig. 7. Best Assignment Result.

Table 4. Comparison of Crossover Rate.

Crossover Rate	Neighborhood	Avg	Min	Max	Var
0.8	2D	159.39	158	164	3.84
	3D	154.81	153	165	4.72
0.9	2D	158.87	155	166	5.21
	3D	153.16	152	164	4.65
1.0	2D	160.75	158	165	5.60
	3D	155.56	154	166	4.72
1.1	2D	161.84	159	168	7.29
	3D	156.33	155	168	5.59

Table 5. Comparison of Mutation Rates.

Mutation Rate	Neighborhood	Avg	Min	Max	Var
0.01	2D	158.90	155	166	8.09
	3D	153.44	152	164	6.64
0.02	2D	158.12	157	163	4.16
	3D	154.70	153	162	4.05
0.03	2D	156.76	155	163	3.98
	3D	152.72	151	161	3.22
0.04	2D	154.7	153	160	3.41
	3D	150.4	150	159	3.04
0.05	2D	152.5	152	160	3.28
	3D	149.4	149	158	3.05
0.06	2D	154.6	154	161	3.70
	3D	150.4	150	160	3.31

The variations of best fitness value with the influence of mutation rate in both 2D and 3D scenarios are shown in Table 5.

We took another four additional cases to check the selected parameters with variations in population size at 50, 100, 300, and 400 to enhance the results. Interestingly, judging from the minimum and variance values, there is significant improvement in the results when the population size is 300. The results show that, at 300, better solution and stability were obtained in both 2D and 3D neighborhoods. Linearly, the results once again showed that at all rates, 3D also gave better results than a 2D

neighborhood in terms of fitness value and stability. The variations of best fitness values with the influence of population size in both 2D and 3D scenarios are shown in Table 6.

In addition, we made a comparison of the execution times needed to reach the convergence solution between 2D and 3D experiments. Although the difference is not too

Table 6. Influence of Population Size.

Population Size	Neighborhood	Avg	Min	Max	Var
50	2D	302.9	302	322	51.75
	3D	284.2	285	312	24.34
100	2D	265.9	265	295	49.03
	3D	243.7	244	261	18.13
300	2D	149.4	151	155	1.23
	3D	147.4	149	153	1.17
400	2D	192.4	191	203	12.64
	3D	183.9	183	192	5.38

Table 7. Comparison of Execution Time (ms).

Neighborhood	Average Time	Min	Max	Var
2D	184.07	181	185	1.51
3D	160.23	159	163	1.51

Table 8. Comparison of the Convergence Process based on Generation Cycle.

Neighborhood	Average	Min	Max	Var
2D	85.0	72	93	35.46
3D	73.5	59	83	22.78

great, 3D still has better execution time than 2D, as shown in Table 7.

Generally, the 3D topology has a complicated procedure, compared to 2D, which means that each generation needs a longer calculation time. However, for the same algorithm and parameters, we obtained an improved result with the 3D cGA over the 2D cGA in terms of execution and convergence time.

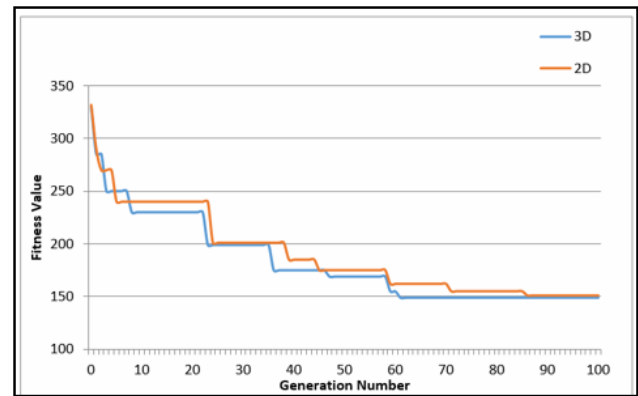
Furthermore, the experiment showed that 3D reached a faster convergence process than 2D, which is indicated by the smaller generation cycle needed to reach the convergence time, as shown in Table 8.

In addition, although in the crossover process, the 3D neighborhood will take a longer time than 2D, the mutation process will go faster, since the mutation only works over single individuals; therefore, no communications are needed among individuals in this process. Furthermore, the overlap of the neighborhoods provides an implicit mechanism for migration. This allows good solutions to spread quickly due to the shorter diameter, as well as diverse degrees of exploration and exploitation, giving longer genetic diversity preservation. As a result, the cGA with a 3D topology has better convergence time than a 2D cGA, as shown in Fig. 8.

5. Conclusion

This study analyzed the performance of cellular GAs with an increase in grid dimensions in order to find the best solution for minimizing the costs of the TGAP with a 3D cGA approach.

From the comparative experiments between the 2D and

**Fig. 8. Convergence Process of 2D and 3D-cGA.**

the 3D cGA, we can see that 3D has good features for solving the TGAP. The data obtained from the experiments showed that 3D gave a better solution in terms of quality and speed, compared to 2D. This approach has given a clear understanding of the advantages from increasing cellular dimensionality in solving the problem.

As future work, we intend to solve the assignment with multi-objective problems, and will try to address concerns about visitor demands.

References

- [1] Alba, E., Dorronsoro, B., "Cellular Genetic Algorithm", ser. Operations Research/Computer Science Interfaces, Springer-Verlag Heidelberg, 2008. [Article \(CrossRef Link\)](#)
- [2] Chen, R. and Wang, H., "Application of Genetic Algorithm to Scheduling of Tour Guides for Tourism and Leisure Industry", *INFOS*, Cairo-Egypt, 2008. [Article \(CrossRef Link\)](#)
- [3] El Zoghby, M., Badr, A. and Hegazy, A. E., "Scheduling of Tour Guides for Tourism and Leisure Industry using Cellular Genetic Algorithm", *IJCSNS*, 2011. [Article \(CrossRef Link\)](#)
- [4] Breukelaar, R. and Back, "Using a Genetic Algorithm to Evolve Behaviour in Multidimensional Cellular Automata", In Proceedings of GECCO'05, 107–114, 2005. [Article \(CrossRef Link\)](#)
- [5] Morales-Reyes, A., Al-Naqi, A., Erdogan, A. T., and Arslan, T., Towards 3D Architectures: "A Comparative Study on Cellular GAs Dimensionality". In Proceedings of the NASA/ESA Conference on Adaptive Hardware and Systems, San Francisco, California, USA, 223–229, 2009. [Article \(CrossRef Link\)](#)
- [6] Sarma, J. and De Jong, K., "An Analysis of the Effects of Neighborhood Size and Shape on Local Selection Algorithms", *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, 1996. [Article \(CrossRef Link\)](#)
- [7] E. Cantu-Paz. "Efficient and accurate Parallel Genetic Algorithms". Evolutionary Computation. Kluwer Academic Publishers, 1st Edition, 2000. [Article \(CrossRef Link\)](#)

- [8] E. Alba and J.M. Troya. “Cellular Evolutionary Algorithms: Evaluating the Influence of Ratio”. PPSN, pp. 29-38, 2000. [Article \(CrossRef Link\)](#)



Lina Setiyani is Ph.D Student of Information Engineering at University of the Ryukyus, Japan. She received her B.Comp. degree in Information Engineering from Janabadra University, Yogyakarta, Indonesia, in 2007. She recently graduated and received M.

Eng. degree in Information Engineering from University of the Ryukyus, Japan, in 2017. Her research interest is finding optimal solution of a problem with genetic algorithm using Java language program.



Takeo Okazaki is Associate Professor of Information Engineering at University of the Ryukyus, Japan. He received his B.Sci. and M.Sci. degrees in Algebra and Mathematical Statistics from Kyushu University, Japan, in 1987 and 1989, respectively and his Ph.D. in Information Engineering from

University of the Ryukyus in 2014. He was a research assistant at Kyushu University from 1989 to 1995. He has been an assistant professor at University of the Ryukyus since 1995. His research interests are statistical data normalization for analysis, statistical causal relationship analysis. He is a member of JSCS, IEICE, JSS, GISA, and BSJ Japan.