Performance Analysis of Localization Strategy for Island Model Genetic Algorithm

Alfian Akbar Gozali
Graduate School of IPS
Waseda University
Wakamatsu-ku, Fukuoka 808-0135 Japan
Email: alfian@fuji.waseda.jp

Shigeru Fujimura
Graduate School of IPS
Waseda University
Wakamatsu-ku, Fukuoka 808-0135 Japan
Email: fujimura@waseda.jp

***Abstract*—Genetic algorithm (GA) is one of the standard solutions to solve many optimization problems. One of a GA type used for solving a case is island model GA (IMGA). Localization strategy is a brand-new feature for IMGA to better preserves its diversity. In the previous research, localization strategy could carry out 3SAT problem almost perfectly. In this study, the proposed feature is aimed to solve real parameter single objective computationally expensive optimization problems. Differ with an issue in previous research which has a prior knowledge and binary, the computationally expensive optimization has not any prior knowledge and floating type problem. Therefore, the localization strategy and its GA cores must adapt. The primary goal of this research is to analyze further the localization strategy for IMGA's performance. The experiments show that the new feature is successfully modified to meet the new requirement. Localization strategy for IMGA can solve all computationally expensive functions consistently. Moreover, this new feature could make IMGA reaches leading ratio 0.47 among other current solvers.**

***Index Terms*—genetic algorithms, island model genetic algorithm, localization strategy, computationally expensive optimization**

I. Introduction

Genetic algorithm (GA) is one of the common solutions to solve many optimization problems. From real world case such as university course timetabling [1] and wind farm layout optimization [2] to theoretical like boolean satisfiability problem [3], [4] and knapsack [5]. All of them using a different kind of GA according to each case. One of a GA type commonly used for solving a case is island model GA (IMGA). IMGA is often used to address the problems from theoretical such as boolean satisfiability problem [3], [4] to a scalable real-world case such as university course timetabling problem [1].

The main idea of IMGA is splitting computation into separate populations (multi-population). By dividing its computational process, there will be more than one GA core pursuing to get the best fitness. Each of populations is communicating each other by migration. In a period (usually generation), the best individual from one island will migrate to another. This mechanism will help IMGA to preserve its population diversity intrinsically. In IMGA, the populations are usually called as islands. The term island and population are interchangeable.

There are several models of IMGA such as AIMGA [1], IMGA for 3SAT [3], IMGA for Job Shop [6], Parallel Quantum GA [7], and IMGA with Localization Strategy [8]. From all those IMGA variations, IMGA with localization strategy has a real potential because it is successful to solve a classic 3SAT problem with the right result. Localization strategy is a mechanism to preserve diversity further of IMGA so it can avoid genetic drift.

Localization strategy tries to make every island in IMGA has a different environment. In localization strategy, a different environment can be different configuration parameters or even different GA core. Localization strategy used to solve 3SAT consists of three islands. Each of them has different GA core. It used standard GA (SGA), pseudo GA (PGA), and informed GA (IGA). By compositing these three of GA core, localization strategy could produce a significantly better result [8].

Localization strategy for IMGA succeeded to overcome the 3SAT problem also because of IGA role. IGA is the slowest island, but it can direct the IMGA into better fitness. That is because IGA uses greedy initialization and directed mutation from prior knowledge of 3SAT. We can see it as a white box problem which can be analyzed previously how the GA operation should be carried out. This prior knowledge which is being exploited by IGA to help IMGA find better fitness than previous research.

Real parameter single objective computationally expensive optimization (computationally expensive optimization) is a mathematical problem which is using an actual parameter and requires expensive computation to achieve a single aim (usually minimization). There are many mathematical functions which are concluding as this problem. Those functions can be unimodal, simple multimodal, hybrid, or composite functions. This issue is often used to test durability, stability, and consistency of proposed algorithm. Functions concluded in this problem type are such as Bent Cigar, Discus, Weierstrass, Schwefel's, Katsuura, HappyCat, and HGBat function. These functions are commonly used to test proposed algorithm from research [9]–[12] to competition purpose [13], [14]. They are usually treated as we only know the proper input and output but without any knowledge of its internal workings.

This research will analyze further the localization strategy

978-1-5090-5504-3/17/$31.00 ©2017 IEEE
SNPD 2017, June 26-28, 2017, Kanazawa, Japan

for IMGA's performance. Previously, IMGA with localization strategy is used to solve a classical NP-complete problem. However, in this research, it will be forced to solve several computationally expensive optimization problems. By treating it like this, we could analyze the performance of localization strategy for IMGA more comprehensive.

II. Localization Strategy for Island Model GA

Localization strategy for IMGA is an approach to treating an island as a single living environment for its individuals. Because of this, each of islands configuration which can be its parameters value or core engine (algorithm) might be different. The configuration differences will tend to be different evolution tracks which can be its speed or pattern. An island can evolve faster or more efficient from other islands. The harmony of island's difference will lead IMGA preserving its diversity to reach better fitness. In previous research, this mechanism was implemented to solve 3SAT [8]. This research will analyze further localization strategy for IMGA performance to solve more complex optimization problem.

In localization strategy for IMGA, there are two types of islands: *master* and *slave*. Master island takes a role in controlling migration by setting the IMGA parameter configuration and decides which island's individual that will be migrated to an island. In the other hand, slave island takes a role in computing or running core genetic algorithm to produce better individual generation to generation. Commonly there is only one master island to control IMGA, but there are more than one slave islands to do core GA computation.

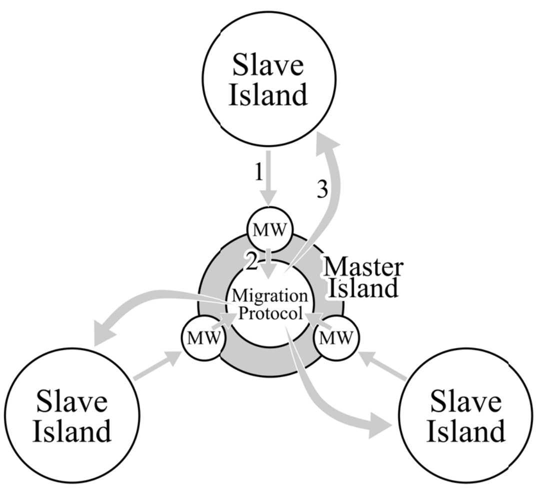

Fig. 1. Interaction between Master and Slave Islands

In localization strategy, the slave islands could have different GA core. As mentioned in section I, in the previous research, Gozali et. al. [8] uses three separate kinds of GA cores: standard GA (SGA), pseudo GA (PGA), and informed GA (IGA). These three types of GA cores are chosen based on classification as standard, speed, and performance GA. Figure 1 illustrates the interaction between master and slave islands. MW is abbreviation from *Migration Window* which is a buffer in master to keep the best individual from every island. The best individual placed in MW is called as a migrant. The three islands are communicating each other under control of master island using *migration protocol*. The pseudocode of migration protocol is explained in Algorithm 1.

Algorithm 1 Migration Protocol Pseudocode

```
Require: an island P send the best individual (p_i^t)
  if B_s^t ≥ θ then
    if migrant Windows ≠ null then
      if number of migrants > 1 then
        find migrant with furthest HD from p_s^t
        migrate that migrant to island P
      else
        migrate migrant to island P
      end if
    end if
  end if
```

Migration protocol is a mechanism to control migration by master island using predefined migration protocol parameter configuration. Previous research adapted *Bias* value from forking genetic algorithm [15] to check diversity of current island's population. The Bias value B_s^t is defined as the measure of genotypic diversity of population P^t and $0.5 \leq B^t \leq 1.0$ when $P_{i,j}^t$ is its allele. The formulation is shown in equation 1.

$$B_s^t = \frac{1}{N \times L} \sum_{j=1}^{L} \left(\left| \sum_{i=1}^{N} p_{i,j}^t - \frac{N}{2} \right| + \frac{N}{2} \right) \tag{1}$$

III. Real Parameter Single Objective Computationally Expensive Optimization

Real parameter single objective computationally expensive optimization (so called computationally expensive optimization) is a bound constrained numerical optimization problem which is using real parameters. Because of its complexity, this problem requires expensive computation in common to achieve a single objective (usually minimization). This research uses computationally expensive optimization problem as had been used in CEC competition 2015 [14]. These problems are treated without any prior knowledge. Table I shows the summary of this problems.

All test functions are minimization problems defined as follows:

$$minf(x), x = [x_1, x_2, ..., x_D]^T \tag{2}$$

where D is the problem dimension. Each function has a predefined shift data for its global optimum ($*F_i$) which is randomly distributed in $[-80, 80]^D$. All test functions are shifted and scalable. The same search ranges (x) are defined for all test functions as $[-100, 100]^D$.

TABLE I
SUMMARY OF THE CEC 2015 EXPENSIVE OPTIMIZATION TEST PROBLEMS

Categories	No	Functions	$^*F_i$
Unimodal functions	TF1	Rotated Bent Cigar Function	100
	TF2	Rotated Discus Function	200
Simple Multimodal functions	TF3	Shifted and Rotated Weierstrass Function	300
	TF4	Shifted and Rotated Schwefels Function	400
	TF5	Shifted and Rotated Katsuura Function	500
	TF6	Shifted and Rotated HappyCat Function	600
	TF7	Shifted and Rotated HGBat Function	700
	TF8	Shifted and Rotated Expanded Griewanks plus Rosenbrocks Function	800
	TF9	Shifted and Rotated Expanded Scaffers F6 Function	900
Hybrid functions	TF10	Hybrid Function 1 (N=3)	1000
	TF11	Hybrid Function 2 (N=4)	1100
	TF12	Hybrid Function 3 (N=5)	1200
Composition functions	TF13	Composition Function 1 (N=5)	1300
	TF14	Composition Function 2 (N=3)	1400
	TF15	Composition Function 3 (N=5)	1500

IV. IMPLEMENTATION FOR ISLAND MODEL GA

The implementation of localization strategy for IMGA in carrying out computationally expensive optimization problem is different from previous research. The major difference is in the chromosome encoding and the prior knowledge existences. Localization strategy for IMGA to solve 3SAT using binary encoding chromosome and computationally expensive optimization has the range with floating type value. Bias formulation and hamming distance measurement would apply the same treat. Prior knowledge of 3SAT which can be well exploited with IGA previous research cannot be obtained from a computationally expensive problem set. That is because the problem set involves many variations, modification, and composition from one to five types of mathematical function. Therefore, this research must quite radically modify the localization model.

A. Fitness function

The fitness function using for this research is derived from the objective function in equation 2. The optimization problem for this case is minimization with zero as the best fitness value.

B. Chromosome Representation

We use floating type encoding for chromosome. Its length will be as wide as the maximum number of input variable which is equal to the dimension number. The dimension number for the problem set is 10 or 30. This research uses chromosome structure which is encoded according to equation 3.

$$chromosome = [x_1, x_2, x_3, ...x_D], D = (10, 30) \tag{3}$$

C. Bias Value

The current bias formulation is formulated to measure diversity for binary type individual. Because we use floating type instead, the bias formulation has to be modified. This research uses bias formulation as explained in equation 4. Where UB is upper bound of range value which is 100 in this case.

$$B^t = \frac{1}{N \times L} \sum_{j=1}^{L} \left(\left| \sum_{i=1}^{N} \left[\frac{p_{i,j}^t + UB}{2 \times UB} \right] - \frac{N}{2} \right| + \frac{N}{2} \right) \tag{4}$$

D. Hamming Distance

Same as bias, the hamming distance (HD) needs to be modified from binary to floating type input variables. Equation 5 shows the modification of HD used in this research. Where L is the length of chromosome and UP is upper bound of range value.

$$HD(x', x'') = \sum_{i=1}^{L} \left| \left[\frac{x_i' + UB}{2 \times UB} \right] - \left[\frac{x_i'' + UB}{2 \times UB} \right] \right| \tag{5}$$

E. Slave Islands

The main idea of localized strategy is placed on the way how to choose different living environment. Previously, GA variant for slave island's choosing uses classification as speed and performance. By considering it, pseudo GA (PGA) and informed GA (IGA) were chosen. For computationally expensive optimization, PGA [16] is still implementable because of its flexibility. Contrary, IGA is a variant which depends on the case's prior knowledge. So, this research needs another approach to modify IGA into a more general case to accommodate this problem. We will explain more specification detail of slave islands later.

F. Standard Genetic Algorithm

The standard GA (SGA) used in this research applies full GA operation from roulette wheel selection, crossover, mutation, to elitism recording. Mutation operator used in SGA has to be changed from flipping (negates one to zero and contrary) into bound constrained randomize within the range $[-100, 100]^D$. The SGA parameters configuration of the slave islands uses consideration from [8]. According to it, the parameter configuration is $\mu = 20 - 40$, $P_c = 50\% - 75\%$, and $P_m = 2\%$.

G. Pseudo Genetic Algorithm

PGA has a quite similar process with SGA, but it implements no roulette wheel selection and mutation. PGA uses complementary chromosome [16] which is static-dynamic for initialization and crossover instead. This mechanism avoids PGA from incest breeding by creating the complement of a parent chromosome to be its couple. For the parameter's value, PGA will use the same configuration with SGA.

Because of floating type of allele, the complementary process must be modified. Previously, complementing is just a negation of all alleles in a binary chromosome. To map the range which is symmetric [-100,100] into [0,1], the range must be divided into [-100,0) and [0,100]. So that, complementary of allele x is -x and vice versa.

H. Informed Genetic Algorithm

The GA core which has to be radically modified is IGA. As mentioned previously, this GA variant needs prior knowledge for doing greedy initialization and directed mutation. We need to make these processes more general to handle a problem such as computationally expensive optimization. Therefore, we make several modifications to direct individual evolution for better fitness from generation to generation.

Modified steps of greedy initialization (execute sequentially):

1) for half of the population (scatter individuals distributive):
    a) Generate quartile values between [LB, UB].
    b) Generate population with individual's allele sparsity within those quartile values.
2) for another half of the population (randomly sparse individuals):
    a) Generate population with individual's allele between [0,1] randomly.
    b) De-normalize allele within [LB, UP] range.

A gene will be treated as a vector which has a direction to direct the mutation. We adapt this approach from particle swarm optimization [17] with simplification. A gene will have value and direction whether UP or DOWN. Figure 2 is the illustration of this chromosome modification. A gene which has UP value will be added by δ if it is mutated and will be subtracted by δ if otherwise. If the mutation result is worse than before, the direction will be changed from UP into DOWN and contrary. Algorithm 2 shows the modified steps of directed mutation (for a chromosome). Where δ is a value which shows how far an allele will be mutated and DMAX is its maximum value ($0 < \delta <$ DMAX).

Fig. 2. Chromosome modification for directed mutation

V. Experimental Result

This research uses function dataset defined previously in Section III. 15 functions are consisting of four types (two unimodal, seven simple multi-modal, three hybrids, and three composite functions). The experiment's main goal is to analyze the performance of localization strategy for IMGA to solve a computationally expensive optimization problem. This main purpose is divided into three experiments:

1) Prove of concept
   This experiment implements modified localization strategy design for IMGA. It will prove that the modification can find a reasonable optimum for the first two unimodal functions (TF1 and TF2). The evaluation parameters for this experiment are the score (minimum value) and execution time. Every function will be executed in ten times to get the average. Additional parameter configuration: migration bias threshold is 0.65 and DMAX is 20.
2) Localization strategy flexibility test
   This experiment is conducted to prove that localization strategy for IMGA can produce reasonable optimum for all datasets (TF1-TF15). The evaluation parameter for this experiment is score and execution time. All test will be run in ten times to get the average. We use a similar configuration parameter for this experiment.
3) Comparative study
   The last experiment objective is to analyze where is the place of localization strategy for IMGA between current solvers. These comparators are MVMO as CEC 2015 competition winner, TunedCMAES, CMAS-ES_QR, iSRPSO, and HumanCog. The result data is extracted from [18]. The comparison evaluation parameter is an only score (minimum value).

The result of first experiment is shown in Table II and III.

Algorithm 2 Directed Mutation Pseudocode

Require: a chromosome x
  set all genes direction to up {initialization}
  for all gene g in chromosome x **do**
    $v_0 \leftarrow$ fitness(x)
    $r \leftarrow$ random(0,1)
    if $r \leq P_m$ **then**
      $\delta \leftarrow$ random(0,DMAX)
      if direction(x_g) = up **then**
        $x_g \leftarrow x_g + 1$
        $v_1 \leftarrow$ evaluate(x)
        if v_0 is better than v_1 **then**
          direction(x_g) $\leftarrow$ $down$
        end if
      else
        $x_g \leftarrow x_g - 1$
        $v_1 \leftarrow$ evaluate(x)
        if v_0 is better than v_1 **then**
          direction(x_g) $\leftarrow$ UP
        end if
      end if
    end if
  end for

TABLE II
EXPERIMENT 1 TF1 RESULT

No	Dimension 10		Dimension 30	
	Score	Time(s)	Score	Time(s)
1	5.39E+07	60	7.85E+07	189
2	1.39E+07	56	8.50E+07	200
3	1.99E+07	58	1.41E+08	201
4	2.39E+07	57	2.84E+07	207
5	2.03E+07	59	9.62E+07	194
6	1.24E+07	57	2.36E+07	201
7	7.14E+06	57	8.41E+07	204
8	1.47E+07	60	3.78E+07	205
9	5.88E+06	58	2.46E+07	205
10	1.03E+07	57	3.70E+07	205
Average	1.82E+07	57.90	6.37E+07	201.10

TABLE III
EXPERIMENT 1 TF2 RESULT

No	Dimension 10		Dimension 30	
	Score	Time(s)	Score	Time(s)
1	1.57E+04	63	8.39E+04	198
2	1.39E+04	65	1.01E+05	206
3	1.39E+04	61	6.53E+04	203
4	9.39E+03	63	8.40E+04	202
5	2.32E+04	60	1.02E+05	203
6	3.44E+04	60	9.05E+04	224
7	1.70E+04	60	8.80E+04	206
8	2.57E+04	61	7.73E+04	202
9	1.63E+04	62	9.46E+04	204
10	3.24E+04	63	8.95E+04	204
Average	2.02E+04	61.80	8.76E+04	205.20

They show the localization strategy for IMGA score result of TF1 and TF2 functions with 10 and 30 dimensions. It was obtained from first ten runs, and the highlighted cell means the best result among repetitions. Both of them show that localization strategy can handle a unimodal problem with reasonable result and execution time (about a minute for D10 and 3 minutes for D30). Moreover, Figure 3 shows the consistency of algorithm. It produced minimum score for TF1 D10 but not for D30. The higher complexity of D30 problem is the reason.

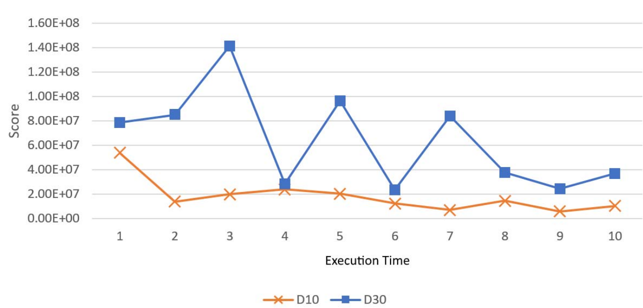

Fig. 3. Consistency of Localization Strategy for IMGA in TF1-D10 and D30

The second experiment result will be represented as line chart in Figure 4. It shows that localization strategy for IMGA could compute all computationally expensive functions (TF1-TF15) well. Figure 4 also explains that 10-dimension problem those have the highest difficulty among all is TF1 (unimodal rotated bent cigar function). Localization strategy for IMGA fails to get a minimum score less than the average score in this problem. For the rest of 10 dimension problems, the localization strategy can successfully utilize IMGA reach a valid minimum score. Furthermore, for the 30-dimension problem, there are three functions which are above average. These are TF1, TF8, and TF10. Even though 30 dimension problems get a worse result rather than 10 ones, the score differences are not too large in average.

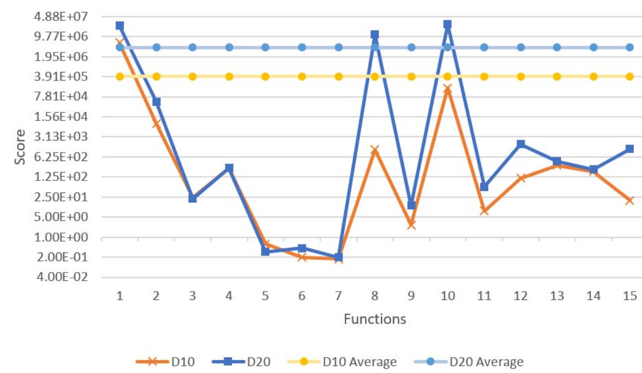

Fig. 4. Result of Localization Strategy for IMGA for All Functions

The last experiment is conducted to compare localization strategy for IMGA result among current solvers for this case. Table IV shows the comparison between localization strategy for IMGA (LS-IMGA) and other five solvers. The gray cell means that solver gets the best score among others. As can be seen in that table, localization strategy for IMGA can lead the minimum score 14 times (eight in D10 and six in D30). Overall, the success ratio of localization strategy beating current solver is 0.47. This ratio is higher than MVMO (0.30) and CMAS-ES_QR (0.13). This number means localization strategy can bring IMGA into significantly better performance among all current solver with great consistency.

VI. CONCLUSION

This research aims to analyze the performance of localization strategy for IMGA further. It implements this new feature into a significantly different problem from previous studies. Real parameter single objective computationally expensive optimization is the addressed problems for this research. The computationally expensive is a minimum optimization problem which hasn't any prior knowledge and has floating type. However, this research succeeds to modify localization strategy to fit in the case. The feature modifications are in its bias function, hamming distance, even GA cores (SGA, PGA, and IGA).

The experiment shows that the new modified version of localization strategy for IMGA could solve a computationally expensive optimization problem with a great result. Localization strategy is proven to have excellent performance and consistency by experiment. Its modified version could tackle almost all 15 functions in 10 and 30 dimensions. It produces

TABLE IV
THE RESULTS COMPARISON OF ALL SOLVERS

No	LS-IMGA		MVMO		TunedCMAES		CMAS-ES_QR		iSRPSO		humanCog	
	D10	D30	D10	D30	D10	D30	D10	D30	D10	D30	D10	D30
1	5.88E+06	2.36E+07	1.93E+02	2.09E+03	1.17E+06	1.52E+06	4.43E+06	8.50E+05	7.40E+06	7.19E+08	3.27E+09	4.74E+10
2	9.19E+03	5.23E+04	1.68E-02	6.93E-03	4.78E+04	1.44E+05	2.58E+04	9.17E+04	3.19E+04	7.67E+04	7.80E+04	1.13E+05
3	2.35E+01	2.17E+01	9.40E+00	3.79E+01	7.62E+00	2.43E+01	2.79E+00	1.15E+01	6.60E+00	2.57E+01	1.12E+01	4.13E+01
4	2.57E+02	2.54E+02	4.65E+02	1.43E+03	1.34E+03	6.11E+03	1.73E+03	6.68E+03	9.25E+02	5.41E+03	2.09E+03	7.99E+03
5	5.71E-01	3.08E-01	1.13E+00	1.68E+00	2.77E+00	3.13E+00	3.20E+00	4.55E+00	2.46E+00	4.24E+00	2.82E+00	4.39E+00
6	1.98E-01	4.11E-01	3.26E-01	5.20E-01	6.00E-01	7.16E-01	4.17E-01	7.28E-01	5.29E-01	6.35E-01	3.63E+00	5.03E+00
7	1.74E-01	1.97E-01	6.37E-01	4.39E-01	6.31E-01	7.28E-01	5.52E-01	7.47E-01	5.71E-01	5.68E-01	2.74E+01	8.86E+01
8	1.12E+03	1.14E+07	4.14E+01	4.03E+02	3.68E+01	2.84E+01	4.68E+00	1.74E+01	5.03E+00	6.26E+02	7.77E+03	5.24E+06
9	2.67E+00	1.25E+01	4.01E+00	1.34E+01	4.17E+00	1.39E+01	3.96E+00	1.34E+01	3.95E+00	1.36E+01	4.16E+00	1.39E+01
10	1.57E+05	2.61E+07	4.97E+02	9.29E+04	5.38E+05	4.89E+06	2.25E+05	3.25E+06	3.53E+05	6.83E+06	1.19E+06	5.60E+07
11	8.25E+00	5.72E+01	1.17E+01	1.43E+02	7.45E+00	2.11E+01	7.63E+00	2.46E+01	7.26E+00	5.09E+01	2.16E+01	2.76E+02
12	1.14E+02	1.71E+03	2.00E+02	8.60E+02	2.39E+02	7.66E+02	2.35E+02	6.27E+02	1.82E+02	7.36E+02	3.08E+02	1.60E+03
13	3.19E+02	4.44E+02	3.16E+02	3.44E+02	3.47E+02	4.15E+02	3.26E+02	3.80E+02	3.31E+02	4.00E+02	4.33E+02	8.35E+02
14	1.93E+02	2.27E+02	2.06E+02	2.76E+02	2.05E+02	2.47E+02	1.97E+02	2.35E+02	2.01E+02	2.65E+02	2.15E+02	3.94E+02
15	1.91E+01	1.18E+03	4.76E+02	1.19E+03	4.42E+02	8.01E+02	3.79E+02	4.90E+02	3.00E+02	9.51E+02	4.74E+02	1.49E+03

above average score among current solvers for the same problem. Taken together, this research proves localization strategy has high potential to be developed further as a new feature for IMGA. It should be challenged to overcome an even more complex problem in future. However, deeper research is needed to analyze overhead and network cost of localization strategy for IMGA. The other alternatives of GA cores also can be implemented to find the most suitable island composition.

ACKNOWLEDGMENTS

This work was supported by Indonesia Endowment Fund for Education (LPDP), a scholarship from Ministry of Finance, Republic of Indonesia. This work was conducted while at Graduate School of Information, Production, and Systems, Waseda University, Japan.

REFERENCES

[1] A. A. Gozali, J. Tirtawangsa, and T. A. Basuki, "Asynchronous Island Model Genetic Algorithm for University Course Timetabling," in *Proceedings of the 10th International Conference on the Practice and Theory of Automated Timetabling*. York: PATAT, 2014, pp. 179–187.

[2] W. Li, E. Ozcan, and R. John, "Multi-objective evolutionary algorithms and hyper-heuristics for wind farm layout optimisation," *Renewable Energy*, vol. 105, pp. 473 – 482, 2017.

[3] S. Mousbah Zeed Mohammed, A. Tajudin Khader, and M. Azmi Al-Betar, "3-SAT Using Island-based Genetic Algorithm," *IEEJ Transactions on Electronics, Information and Systems*, vol. 136, no. 12, pp. 1694–1698, 2016.

[4] J. Li, H. Wang, J. Liu, and L. Jiao, "Solving sat problem with a multi-agent evolutionary algorithm," in *2007 IEEE Congress on Evolutionary Computation*, Sept 2007, pp. 1416–1422.

[5] Q. J. Li and K. Y. Szeto, "Efficiency of adaptive genetic algorithm with mutation matrix in the solution of the knapsack problem of increasing complexity," in *2015 IEEE Congress on Evolutionary Computation (CEC)*, May 2015, pp. 31–38.

[6] M. Kurdi, "An effective new island model genetic algorithm for job shop scheduling problem," *Computers & Operations Research*, vol. 67, pp. 132–142, 2016.

[7] J. Gu, X. Gu, and M. Gu, "A novel parallel quantum genetic algorithm for stochastic job shop scheduling," *Journal of Mathematical Analysis and Applications*, vol. 355, no. 1, pp. 63–81, 2009.

[8] A. A. Gozali and S. Fujimura, "Localization strategy for island model genetic algorithm to preserve population diversity," in *International Conference on Computer and Information Science*, 2017.

[9] M. A. Sotelo Figueroa, A. Hernandez Aguirre, A. Espinal, and J. A. Soria Alcaraz, *Differential Evolution with Self-adaptive Gaussian Perturbation*. Cham: Springer International Publishing, 2017, pp. 617–629.

[10] X.-G. Li, S.-F. Han, and C.-Q. Gong, "Analysis and improvement of fireworks algorithm," *Algorithms 2017*, vol. 10, no. 1, p. 26, feb 2017.

[11] K. Mishra, A. Tripathi, S. Tiwari, and N. Saxena, "Evolution based memetic algorithm and its application in software cost estimation," *Journal of Intelligent & Fuzzy Systems*, vol. 32, no. 3, pp. 2485–2498, Feb 2017.

[12] J. Sakuma and S. Kobayashi, "Extrapolation-directed crossover for real-coded GA: overcoming deceptive phenomena by extrapolative search," in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*. Institute of Electrical and Electronics Engineers (IEEE), 2001.

[13] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective real-parameter numerical optimization," CEC 2017, Tech. Rep., 2017.

[14] Q. Chen, B. Liu, Q. Zhang, J. J. Liang, P. N. Suganthan, and B. Y. Qu, "Problem definitions and evaluation criteria for cec 2015 special session on bound constrained single-objective computationally expensive numerical optimization," CEC 2015, Tech. Rep., 2015.

[15] S. Tsutsui, Y. Fujimoto, and A. Ghosh, "Forking genetic algorithms: Gas with search space division schemes," *Evolutionary Computation*, vol. 5, no. 1, pp. 61–80, March 1997.

[16] Q. Chen, Y. Zhong, and X. Zhang, "A pseudo genetic algorithm," *Neural Computing and Applications*, vol. 19, no. 1, pp. 77–83, 2010.

[17] J. Kennedy, *Particle Swarm Optimization*. Boston, MA: Springer US, 2010, ch. Reference Work Entry, pp. 760–766.

[18] CEC 2015 Official, "Competition on real-parameter single objective computationally expensive optimization (expensive result comparison slide)," CEC 2015, Tech. Rep., 2015.