

# Localized Island Model Genetic Algorithm in Population Diversity Preservation

Alfian Akbar Gozali  
Graduate School of IPS  
Waseda University  
Kitakyushu, Japan  
alfian@fuji.waseda.jp

Shigeru Fujimura  
Graduate School of IPS  
Waseda University  
Kitakyushu, Japan  
fujimura@waseda.jp

**Abstract**—Premature convergence in island model is a consequence of the selection in migration mechanism. It is a process of migrating several individuals (usually the best one) from a source into destination island to keep its diversity. The main reason is the similar characteristic of relocated individual because of the genetic operator configurations are similar. Localized Island Model GA (LIMGGA) tries to implement different island characteristics (localization strategy) to preserve the islands' diversity. By harmonizing standard GA, pseudo GA, and informed GA; LIMGGA could overcome general optimization problem with a great result and acceptable execution time. Moreover, because of its success in maintaining the diversity, LIMGGA could lead the current best-known-so-far solver for this case.

**Keywords**—migration policy, island model, genetic algorithm, localization strategy

## I. INTRODUCTION

Since Fraser introduced Genetic Algorithm (GA) [1] and Bremermann modified it [2] in late 1950, this optimization technique had developing rapidly and adapting widely for many cases. This algorithm has overcome many theoretical problems such as 3-SAT [3], [4], job shop scheduling [5], and many-objective optimization [6]. Furthermore, it has tackled a real-world case like university course timetabling [7], [8] and wind farm layout optimization [9].

Different cases forced researcher to implement suitable kind of GA. Among various types of its implementation, island model (IM) commonly used for solving a lot of cases especially complex problem which need scalability. Island model GA intrinsically preserves its population diversity. It creates an opportunity for its populations (set of solutions) widen their ranges and finally gives an advantage in system efficiency. The problem occurred while a whole islands (populations) tends to converge towards local optimum. Genetic drift forces this problem to happen. This term refers to the set of gene values (alleles) frequency change in a population due to a random sampling of organisms [10]. Because of island model GA uses finite population sizes, it highly possible leads into this problem and loss its diversity as the result.

Premature convergence in island model is a consequence of the selection in migration mechanism. This is a process

of migrating several individuals (usually the best one) from source into destination island in order to keep the system diversity. Yet in implementation, this movement process in a certain time could cause genetic drift. The main reason is the similar characteristic of migrated individual because of the genetic operator configurations are similar.

Naturally, an individual migrates to find a potentially better living environment [11], [12]. It moves from a place with particular characteristics to another place with different one which is potentially better. Ray in [13] stated that living environment (medium) is one of the challenges to inoculate natural evolution into artificial media. This is because the evolutionary process is mainly concerned with adaptation to the living environment.

A shifting paradigm of an island from just a set of solutions to become a living environment with different characteristics (configurations) can be a breakthrough to solve the genetic drift problem. This configuration difference may higher the chance to create a different evolution speed or solution range. It will force each island to generate different characteristic of individuals that has higher probability to jump into another local optimum. Finally, this mechanism will preserve the diversity among islands and delayed the premature convergence.

This work aims the implementation analysis of different island characteristics (localization strategy) to preserve the islands diversity. We call the IM with localization strategy as Localized Island Model Genetic Algorithm (LIMGGA). The main contribution of this work are (1) to introduce a new mechanism of preserving diversity by applying localization strategy and (2) to analyze its effect in handling general optimization problem.

This paper consist of six sections. Section I explains the problem background and the explicit statement of goal and contribution of this research. Section II shows the related works from previous researchers which contains an explanation of island model genetic algorithm. In Section III, we try to introduce LIMGGA deeper from its main idea, the localization strategy, until the migration protocol. Next (Section IV) talks about the implementation of LIMGGA while facing a general optimization problem. Section V discusses the conducted experiment and the analysis of its result. The last but not the least, Section VI concludes all of the works and answers of

Thank you for Indonesia Endowment Fund for Education (LPDP), a scholarship from Ministry of Finance, Republic of Indonesia, for supporting this work.

the problem explained in the first section.

## II. ISLAND MODEL GENETIC ALGORITHM

Island Model Genetic Algorithm (IMGA) is a distributed model of GA which splits its main computational process into several computers (islands) instead of running it in only one machine. This mechanism offers higher scalability and gives better chance to evade the local optimum trap. As mentioned in Section I, proposed technique by Gozali [7], Salmah [4], and Kurdi [5] adapted this kind of distributed GA. Gozali et.al. implemented asynchronous model to overcome university course timetabling with different kind of resources (computer). Salmah et.al. used a synchronous technique to carried out 3SAT, a classical computational problem. Kurdi et.al. proposed different modification part of island model to solve job shop problem. Instead of migrating the best one, he prefers to migrate the worst one in every generations.

This work implemented asynchronous mechanism used in AIMGA [7]. Figure 1 shows architecture level view of AIMGA. The slave Island runs all of the computational processes of the GA and master island controls the distribution order of each best individual's islands.

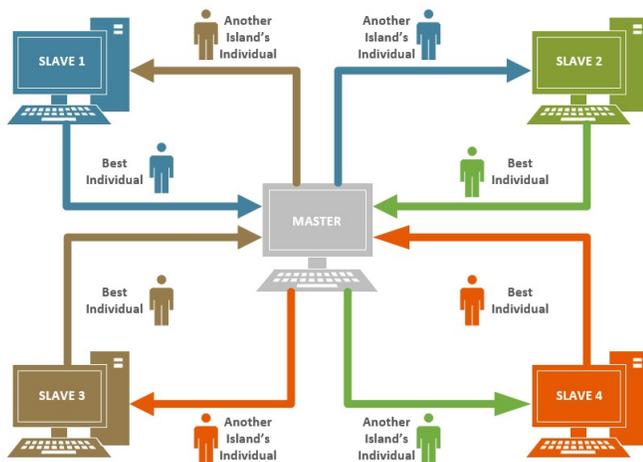


Fig. 1. Architecture level view of AIMGA

We used AIMGA architecture because of its ability to overcome idle computation problem due to different computer specifications. This asynchronous model became the core of LIMGA because in proposed method, we had to implement not only different computer specifications but also various kind of GA cores.

## III. LOCALIZED ISLAND MODEL GENETIC ALGORITHM

LIMGA is a new paradigm-shift approach which sees an island as a single living environment of its population. As the implication, each islands configuration which could be value of its parameters or even its core algorithm might be different. This differences could branch into separate evolution path which could be its speed or chromosome pattern. An island may evolve quicker or more efficient to produce a better individual than other islands.

The main idea of LIMGA is to harmonize each island's differences. Their combination will lead IMGA preserving its diversity and get a better solution as the result. This proposed method has a mechanism called localization strategy. There are two kinds of islands: *master* and *slave*. The master controls migration by setting the IMGA parameter configuration and deciding the slave island which has to migrate its best individual to targeted slave. In the other hand, slave island takes a role in computing or running core genetic algorithm to produce better individual every generation. Commonly there is only one master island to control IMGA, but there are more than one slave islands to do core GA computation.

Before LIMGA, there are several previous researches related with diversity preservation in island model genetic algorithm such as dual population (DPGA) [14]–[16], diversity guided evolutionary programming (DGEP) [17], and ensemble of niching algorithms [18]. The first method, DPGA, split a population into two subpopulations: main and reserve. This approach could make the population more diverse but as long as the GA core is same, the evolution track would remain similar and it would lead into premature convergence. The two last methods, DGEP and ensemble of niching algorithm, more focused in the greedy approach on the selection process by measuring the distance between parent candidates.

Moreover in previous researches [19], [20], LIMGA used three different algorithms as each island's GA core. This research applied single machine model such as Standard GA (SGA) [21] for balance, Pseudo GA (PGA) [22] for speed, and Informed GA (IGA) [7] for performance. Figure 2 shows illustration of LIMGA mechanism used in this work. Master island controls migration by setting the initial parameter configuration and distributes island's individual to the expected island. And the slave islands (SGA, PGA, and IGA) task is processing main genetic algorithm to produce better offspring every generation.

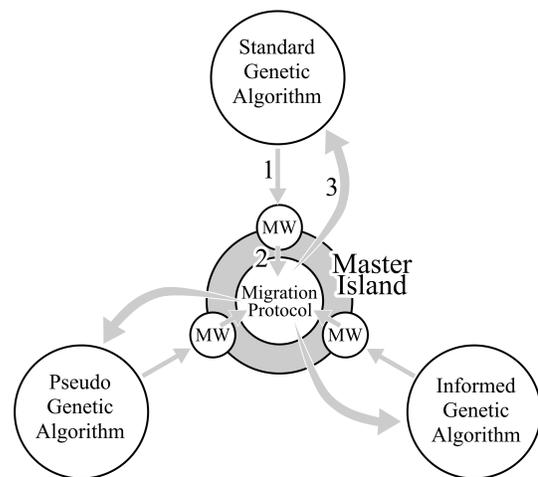


Fig. 2. Interaction between Master and Slave Islands

MW is an abbreviation of *Migration Window*. MW such as a buffer in master to keep the best individual from every island.

They are called as migrants. The three slaves communicates each other under the rule of master island using *migration protocol*. Its pseudocode is explained in Algorithm 1.

---

**Algorithm 1** Migration Protocol Pseudocode
 

---

**Require:** an island  $P$  send the best individual ( $p_i^t$ )  
**if**  $B_s^t \geq \theta$  **then**  
   **if** migrant Windows  $\neq null$  **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 individual movement by master island using predefined migration protocol parameter configuration. Previous research adapted *Bias* value from forking genetic algorithm [23] 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. Equation 1 shows formulation of the bias value. HD is hamming distance between  $p_i^t$  and  $!p_i^t$ .

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

#### IV. LIMGA FOR GENERAL OPTIMIZATION PROBLEM

This work applied LIMGA to solve general optimization problem. The chosen case for this research 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 used similar general optimization problem as had been used in CEC competition 2015 [24]. And we have no prior knowledge regarding these problems. Table I shows the summary of this problems.

All test functions are minimization problems defined as follows:

$$\min f(x), x = [x_1, x_2, \dots, x_D]^T \quad (2)$$

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  $[-100, 100]^D$ .

To implement LIMGA for carrying out this general optimization problems, there are several GA modifications especially in the chromosome structure, IMGA parameters formulation, and genetic operator procedures.

#### A. Fitness Function

This work used fitness function which is similar with the objective formulation in equation 2. The optimization goal for every case is minimization with zero as the most optimum fitness value.

#### B. Chromosome Representation

Each islands used floating type encoding chromosome representation. The number of genes is the maximum input variable which equals the dimension number ( $D$ ). Therefore, the length for this problem set is 10 or 30. This research used encoded chromosome structure according to equation 3.

$$chromosome = [x_1, x_2, x_3, \dots, x_D], D = (10, 30) \quad (3)$$

#### C. Bias Value

The previous bias formulation was formulated to measure the diversity of binary type chromosome representation. Because this work used floating type instead, we modified the bias formulation. This research used a modification as explained in equation 4. Where  $UB$  is an upper bound of the range value which is 100 in this case.

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

#### D. Hamming Distance

Same as the bias formulation, 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  $UB$  is an upper bound of the range value.

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

#### 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 [19], [20]. By considering it, pseudo GA (PGA) and informed GA (IGA) were chosen. For computationally expensive optimization, PGA [22] 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.

1) *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 [20]. According to it, the parameter configuration is  $P_c = 75\%$  and

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

$P_m = 2\%$ . This work used  $\mu = 60$  for the sake of division easiness by three islands.

2) *Pseudo Genetic Algorithm*: PGA has a quite similar process with SGA, but it implements no roulette wheel selection and mutation. PGA uses complementary chromosome [22] 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.

3) *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 [25] with simplification. A gene will

have value and direction whether UP or DOWN. Figure 3 is the illustration of this chromosome modification. Algorithm 2 shows the modified steps of directed mutation (for a chromosome). Where  $\delta$  is a value which shows how far an allele will be mutated and  $DMAX$  is its maximum value ( $0 < \delta < DMAX$ ).

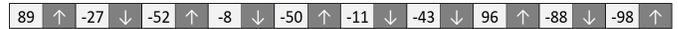


Fig. 3. Chromosome modification for directed mutation

---

#### Algorithm 2 Directed Mutation Pseudocode

---

**Require:** a chromosome  $x$

set all genes direction to *up* {initialization}

**for all** gene  $g$  in chromosome  $x$  **do**

$v_0 \leftarrow \text{fitness}(x)$

$r \leftarrow \text{random}(0,1)$

**if**  $r \leq P_m$  **then**

$\delta \leftarrow \text{random}(0,DMAX)$

**if**  $\text{direction}(x_g) = \textit{up}$  **then**

$x_g \leftarrow x_g + 1$

$v_1 \leftarrow \text{evaluate}(x)$

**if**  $v_0$  is better than  $v_1$  **then**

$\text{direction}(x_g) \leftarrow \textit{down}$

**end if**

**else**

$x_g \leftarrow x_g - 1$

$v_1 \leftarrow \text{evaluate}(x)$

**if**  $v_0$  is better than  $v_1$  **then**

$\text{direction}(x_g) \leftarrow \textit{UP}$

**end if**

**end if**

**end for**

---

## V. EXPERIMENTAL RESULT

The main goals of the conducted experiments are (1) to implement LIMGA, a new mechanism of preserving diversity

in IMGA by applying localization strategy and (2) to analyze its effect in handling general optimization problem. According these goals, we did three main experiments:

- 1) The first experiment objected the LIMGA proof of concept. We implemented general optimization problem for SGA, PGA, IGA, and LIMGA with same total population size. For example, if we used  $\mu = 30$  for single machine SGA, PGA, and IGA; for each LIMGA's slave, we used  $\mu = 10$ . All of the experiments used following LIMGA parameters: migration bias threshold is 0.65 and *DMAX* is 20. This first experiment used *D10* and *D30* unimodal functions (*TF1* and *TF2*) with fitness score and execution time as its evaluation parameters.
- 2) The second experiment aimed diversity preservation analysis of LIMGA. Similar with the first, we run SGA, PGA, IGA, and LIMGA for *TF1* and *TF2* with *D10* and *D30* then comparing the averaged bias value. The evaluation parameter of this experiment is bias value only (degree of diversity).
- 3) The last experiment find the position of LIMGA among the best-known-so-far general optimization problems solver. We execute LIMGA to find the optimum value of *TF1* – *TF15*. Then we compared LIMGA with the winner of CEC 2015, MVMO [26], but only in their scores.

Table II represents result of the first experiment. Table values shows average score of GA method with ten repetitions. In this table we can see that LIMGA dominates all cases than other single models. Among these three, IGA leads TF1 results and PGA gives the best score in TF2. From this statistic, we can conclude that LIMGA could dominate all result because its combination of all single algorithms.

TABLE II  
THE RESULTS COMPARISON OF EXPERIMENT I

Function	SGA	PGA	IGA	LIMGA
TF1 - D10	3.61E+09	6.21E+09	3.36E+08	1.82E+07
TF1 - D30	2.42E+10	5.49E+10	2.33E+09	6.37E+07
TF2 - D10	5.19E+06	3.35E+04	2.65E+06	2.02E+04
TF2 - D30	1.09E+06	1.01E+05	3.10E+05	8.76E+04

To get these results, LIMGA consumes relatively considerable amount of time. In Figure 4, we can observe that for smaller dimension which need less execution effort, LIMGA is more inferior than others. That is because this distributed model deals with communication cost between islands. However, LIMGA could get a faster result for more dimensions. Fewer population size of its slaves could compensate the communication cost. Regarding this result, we can conclude that LIMGA has a high potential replacing single model GA. It could produce better solution with faster execution time along the problem scalability.

In the second experiment, we analyzed bias value trend across generations. Figure 5 shows bias value time-line chart from the first to 1000th generation for every algorithm. To make the explanation become clearer, Figure 6 draws only their polynomial function with order=3.

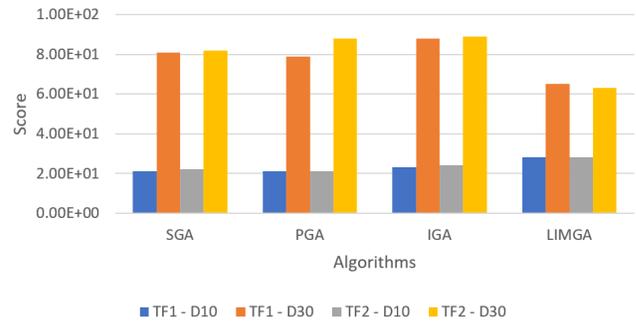


Fig. 4. Experiment 1 Time Comparison

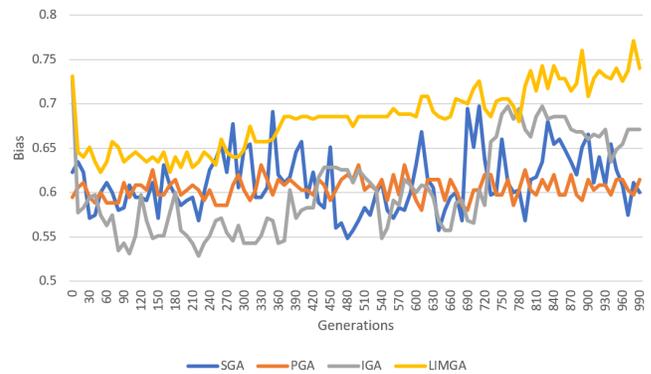


Fig. 5. Experiment 2 Bias Trend-line

From these figures, we could see that PGA has the most stable (monotonic) trend proven by its standard deviation which is the smallest among others. However, PGA's bias value is always below the predefined diversity threshold (0.65). IGA and SGA has unstable trend of bias but IGA tends to has high bias in the end of its generation. The proposed solution, LIMGA, be able to draw increasing trend of bias value tough it seems not too stable but still considerable. This experiment shows LIMGA ability to maintain its diversity in order to avoid local optimum trap.

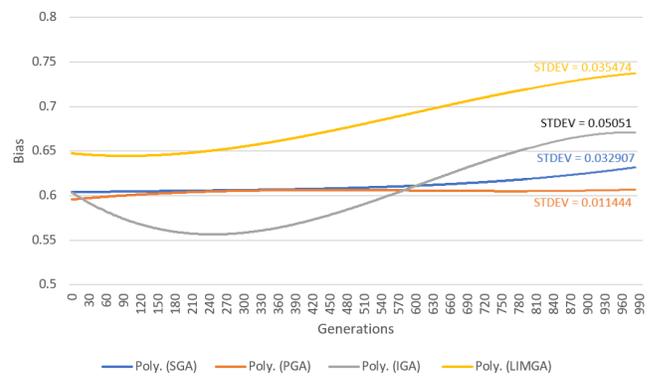


Fig. 6. Experiment 2 Polynomial Bias Trend-line

The last experiment measures how far is the position of

TABLE III  
THE COMPARISON BETWEEN LIMGA AND MVMO

Function	LIMGA		MVMO	
	D10	D30	D10	D30
TF1	5.88E+06	2.36E+07	1.93E+02	2.09E+03
TF2	9.19E+03	5.23E+04	1.68E-02	6.93E-03
TF3	2.35E+01	2.17E+01	9.40E+00	3.79E+01
TF4	2.57E+02	2.54E+02	4.65E+02	1.43E+03
TF5	5.71E-01	3.08E-01	1.13E+00	1.68E+00
TF6	1.98E-01	4.11E-01	3.26E-01	5.20E-01
TF7	1.74E-01	1.97E-01	6.37E-01	4.39E-01
TF8	1.12E+03	1.14E+07	4.14E+01	4.03E+02
TF9	2.67E+00	1.25E+01	4.01E+00	1.34E+01
TF10	1.57E+05	2.61E+07	4.97E+02	9.29E+04
TF11	8.25E+00	5.72E+01	1.17E+01	1.43E+02
TF12	1.14E+02	1.71E+03	2.00E+02	8.60E+02
TF13	3.19E+02	4.44E+02	3.16E+02	3.44E+02
TF14	1.93E+02	2.27E+02	2.06E+02	2.76E+0
TF15	1.91E+01	1.18E+03	4.76E+02	1.19E+03

LIMGA from the winner of CEC 2015, MVMO. Table III points the comparison between LIMGA and MVMO. The result data is extracted from [26]. The comparison evaluation parameter is an only score (minimum value). The gray cell means that LIMGA could gets better score among MVMO. From this table, localization strategy for IMGA can lead the minimum score 16 times (nine in D10 and seven in D30). Overall, the success ratio of localization strategy beating current solver is 0.53. This ratio is higher than MVMO (0.47). This number means localization strategy can bring IMGA into significantly better performance among the best-known-so-far solver with great consistency.

## VI. CONCLUSION

This work goal is the implementation analysis of different island characteristics (localization strategy) to preserve the islands diversity. We applied this mechanism into island model GA to be Localized Island Model GA (LIMGA). This conducted research contribution is mostly to introduce a new mechanism of preserving diversity by applying localization strategy and to analyze its effect in handling general optimization problem. As its problems, this work used 15 functions from CEC 2015 real-parameter single objective computationally expensive optimization competition.

This proposed model used three slave islands: SGA, PGA, and IGA. By modifying the each GA structures and operators to meet the cases, LIMGA could give great result while carrying out the general optimization cases. The first experiment shows that LIMGA dominates the result over those three single models. The execution time is consider as acceptable yet scalable for more complex problems. In the second test, LIMGA able to show its ability to maintain the diversity to get a better solution. The last experiment defines the position of LIMGA against the best-known-so-far solver (MVMO). The proposed model could lead more than half problems for both dimensions with 0.53 success ratio.

Regarding these findings, localization strategy seems promising and having a high potential for island model. The harmony of different GA core variants lead LIMGA towards

the unique evolution tracks. As the result, it could maintain its diversity, has better chance to avoid local optimum, and finally produce a better results. However, deeper investigation of LIMGA implementation for more realistic problems are still needed. In the future, we may analyze the effectiveness of localization strategy to handle job-shop, flow-shop, course timetabling, and other real cases.

## ACKNOWLEDGMENTS

Indonesia Endowment Fund for Education (LPDP), a scholarship from Ministry of Finance, Republic of Indonesia, supports this work. We conduct this research while at Graduate School of Information, Production, and Systems, Waseda University, Japan.

## REFERENCES

- [1] Alex Fraser. Simulation of genetic systems by automatic digital computers. *Australian Journal of Biological Science*, 10(2):492–499, 1957.
- [2] Hans-Joachim Bremermann. The evolution of intelligence : The nervous system as a model of its environment. techreport 1, Dept. of Mathematics, Univ. of Washington, Seattle, 1958. Contract No. 477(17).
- [3] Jinshu Li, Heyong Wang, Jing Liu, and Lincheng Jiao. Solving sat problem with a multiagent evolutionary algorithm. In *IEEE Congress on Evolutionary Computation*, pages 1416–1422, Sept 2007.
- [4] Salmah Mousbah Zeed Mohammed, Ahamad Tajudin Khader, and Mohammed Azmi Al-Betar. 3-SAT Using Island-based Genetic Algorithm. *IEEE Transactions on Electronics, Information and Systems*, 136(12):1694–1698, 2016.
- [5] Mohamed Kurdi. An effective new island model genetic algorithm for job shop scheduling problem. *Computers & Operations Research*, 67:132–142, 2016.
- [6] D. W. Gong, J. Sun, and Z. Miao. A set-based genetic algorithm for interval many-objective optimization problems. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1, 2017.
- [7] Alfian Akbar Gozali, Jimmy Tirtawangsa, and Thomas Anung Basuki. Asynchronous Island Model Genetic Algorithm for University Course Timetabling. In *10th International Conference on the Practice and Theory of Automated Timetabling*, pages 179–187, York, 2014. PATAT.
- [8] Nelishia Pillay Rosanne Els. An evolutionary algorithm hyper-heuristic for producing feasible timetables for the curriculum based university course timetabling problem. In *World Congress on Nature and Biologically Inspired Computing*, Kitakyushu, December 2010.
- [9] Wenwen Li, Ender Ozcan, and Robert John. Multi-objective evolutionary algorithms and hyper-heuristics for wind farm layout optimisation. *Renewable Energy*, 105:473 – 482, 2017.
- [10] A. Rogers and A. Prugel-Bennett. Genetic drift in genetic algorithm selection schemes. *IEEE Transactions on Evolutionary Computation*, 3(4):298–303, Nov 1999.
- [11] Andres J. Garcia, Deepa K. Pindolia, Kenneth K. Lopiano, and Andrew J. Tatem. Modeling internal migration flows in sub-saharan africa using census microdata. *Migration Studies*, 3(1):89, 2014.
- [12] Yannis Vardanis, Jan ke Nilsson, Raymond H.G. Klaassen, Roine Strandberg, and Thomas Alerstam. Consistency in long-distance bird migration: contrasting patterns in time and space for two raptors. *Animal Behaviour*, 113:177 – 187, 2016.
- [13] Thomas S. Ray. An evolutionary approach to synthetic biology: Zen in the art of creating life. *Artificial Life*, 1(1):179–209, October 1993.
- [14] T. Park and K. R. Ryu. A dual-population genetic algorithm for adaptive diversity control. *IEEE Transactions on Evolutionary Computation*, 14(6):865–884, Dec 2010.
- [15] A. J. Umbarkar and M. S. Joshi. Article: Dual population genetic algorithm (ga) versus openmp ga for multimodal function optimization. *International Journal of Computer Applications*, 64(19):29–36, February 2013.
- [16] A.J. Umbarkar, M.S. Joshi, and Wei-Chiang Hong. Multithreaded parallel dual population genetic algorithm (mpdpga) for unconstrained function optimizations on multi-core system. *Applied Mathematics and Computation*, 243(Supplement C):936 – 949, 2014.

- [17] Mohammad Shafiu Alam, Md. Monirul Islam, Xin Yao, and Kazuyuki Murase. Diversity guided evolutionary programming: A novel approach for continuous optimization. *Applied Soft Computing*, 12(6):1693 – 1707, 2012.
- [18] E.L. Yu and P.N. Suganthan. Ensemble of niching algorithms. *Information Sciences*, 180(15):2815 – 2833, 2010.
- [19] Alfian Akbar Gozali and Shigeru Fujimura. Performance analysis of localization strategy for island model genetic algorithm. In *18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2017)*, 2017.
- [20] Alfian Akbar Gozali and Shigeru Fujimura. Localization strategy for island model genetic algorithm to preserve population diversity. In Roger Lee, editor, *Computer and Information Science*, chapter Studies in Computational Intelligence, pages 149–161. Springer International Publishing, Cham, 2018.
- [21] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.
- [22] Qijian Chen, Yuanchang Zhong, and Xiaohong Zhang. A pseudo genetic algorithm. *Neural Computing and Applications*, 19(1):77–83, 2010.
- [23] S. Tsutsui, Y. Fujimoto, and A. Ghosh. Forking genetic algorithms: Gas with search space division schemes. *Evolutionary Computation*, 5(1):61–80, March 1997.
- [24] 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. Technical report, CEC 2015, 2015.
- [25] James Kennedy. Particle swarm optimization. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning*, chapter Reference Work Entry, pages 760–766. Springer US, Boston, MA, 2010.
- [26] CEC 2015 Official. Competition on real-parameter single objective computationally expensive optimization (expensive result comparison slide). Technical report, CEC 2015, 2015.