Analysis of Evolutionary Multi-Tasking as an Island Model

Ryuichi Hashimoto
Department of Computer Science and Intelligent Systems,
Graduate School of Engineering,
Osaka Prefecture University,
Osaka, 599-8531, Japan
ryuichi.hashimoto@ci.cs.osakafu-u.ac.jp

Hisao Ishibuchi
Shenzhen Key Laboratory of Computational Intelligence
Department of Computer Science and Engineering,
Southern University of Science and Technology,
Shenzhen, 518005, China
hisao@sustc.edu.cn

Naoki Masuyama
Department of Computer Science and Intelligent Systems,
Graduate School of Engineering,
Osaka Prefecture University,
Osaka, 599-8531, Japan
masuyama@cs.osakafu-u.ac.jp

Yusuke Nojima
Department of Computer Science and Intelligent Systems,
Graduate School of Engineering,
Osaka Prefecture University,
Osaka, 599-8531, Japan
nojima@cs.osakafu-u.ac.jp

ABSTRACT

Recently, an idea of evolutionary multi-tasking has been proposed and applied to various types of optimization problems. The basic idea of evolutionary multi-tasking is to simultaneously solve multiple optimization problems (i.e., tasks) in a cooperative manner by a single run of an evolutionary algorithm. For this purpose, each individual in a population has its own task. This means that a population of individuals can be viewed as being divided into multiple sub-populations. The number of sub-populations is the same as the number of tasks to be solved. In this paper, first we explain that a multi-factorial evolutionary algorithm (MFEA), which is a representative algorithm of evolutionary multi-tasking, can be viewed as a special island model. MFEA has the following two features: (i) Crossover is performed not only within an island but also between islands, and (ii) no migration is performed between islands. Information of individuals in one island is transferred to another island through inter-island crossover. Next, we propose a simple implementation of evolutionary multi-tasking in the framework of the standard island model. Then, we compare our island model with MFEA through computational experiments. Promising results are obtained by our implementation of evolutionary multi-tasking.

CCS CONCEPTS

• **Mathematics of computing → Optimization algorithms**

KEYWORDS

Evolutionary computation, multi-tasking, island model

ACM Reference format:

R. Hashimoto, H. Ishibuchi, N. Masuyama, and Y. Nojima. 2018. Analysis of Evolutionary Multi-Tasking as an Island Model. In *Companion of ACM GECCO conference, Kyoto, Japan, July 2018 (GECCO'18)*, 4 pages.
DOI: 10.1145/3205651.3208228

1 INTRODUCTION

Recently, an idea of evolutionary multi-tasking has been proposed to solve multiple optimization problems (i.e. tasks) by a single run of an evolutionary algorithm [2, 3, 7, 11]. This idea applied to various types of optimization problems such as bi-level [8], multi-objective [6], and many-objective problems [10].

In evolutionary multi-tasking, each individual in a population has its own task. The task assignment is based on its relative strength for each task. For example, let us assume that the ranking of an individual for three tasks is the 10th for Task 1, 20th for Task 2, and 30th for Task 3. Then, it may be a good idea to assign this individual to Task 1. A population can be viewed as being divided into multiple sub-populations by the task assignment. The number of the sub-populations is the same as the number of the tasks to be solved. Individuals in each sub-population are evolved to optimize the corresponding task by crossover and mutation. Information in one sub-population is transferred to others by crossover between individuals from different sub-populations. In this manner, multiple tasks are solved in a cooperative manner in evolutionary multi-tasking.

Whereas evolutionary multi-tasking is a hot topic in the field of evolutionary computation, its search behavior has not been clearly analyzed and explained yet. As a result, it is still not very clear why good results can be obtained by evolutionary multi-tasking. One reason for the difficulty of analyzing evolutionary multi-tasking seems to be its complicated implementation as a

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07…$15.00
https://doi.org/10.1145/3205651.3208228

single-population model. For example, MFEA (multi-factorial Evolutionary Algorithm) [2, 5], which is a representative of evolutionary multi-tasking algorithms, calculates the following four indexes [5]: the factorial cost of each individual for each task (which is the fitness value if each task is an unconstrained optimization problem), the factorial rank of each individual for each task (which is the ranking in the population for each task), the skill factor of each individual (the task with the highest rank), and the scalar fitness of each individual (the inverse of the highest rank). Using these four indexes, a task and a fitness value are assigned to each individual. Whereas a population can be viewed as being divided into sub-populations by the task assignment (i.e., by the skill factor of each individual), MFEA is implemented as a single-population model. This leads to somewhat complicated procedures for generating an offspring and assigning a task to the generated offspring. For example, to generate an offspring, the skill factors of the two parents are checked. If they have the same skill factors, two offspring are generated by crossover and mutation. In this case, the same skill factor as the parents is assigned to the generated offspring. If the two parents have different skill factors, a special crossover probability is used. With this probability, two offspring are generated by crossover and mutation. The skill factor of each offspring is calculated for task and fitness assignment. Otherwise (i.e., when crossover is not used), an offspring is generated by mutation from each parent. In this case, the generated offspring has the same skill factor as its own parent. Due to the crossover between individuals with different skill factors, the number of the generated offspring with each skill factor (i.e., offspring for each task) is different. Through a generation update mechanism, the same number of individuals with each skill factor are selected from the parent and offspring populations.

In this paper, we propose a simple implementation of multi-tasking in the framework of island models [1, 4], in order to examine whether competitive performance can be obtained from a simple island model in comparison with a complicated single-population evolutionary multi-tasking model (i.e., MFEA). The difference between the standard island model and the proposed multi-tasking island model is as follows: Each island has its own fitness function in the proposed model. Except for this difference, our proposed island model is the same as the standard island model based on the crossover within each island and migration between islands.

This paper is organized as follows. In Section 2, we explain MFEA (which is a representative algorithm of evolutionary multi-tasking) as a special island model. In Section 3, we propose a simple implementation of multi-tasking as an island model. In Section 4, we examine the performance of the proposed multi-tasking island model. We use multi-tasking test problems in [2]. Finally, we conclude this paper in Section 5.

2 EVOLUTIONARY MULTI-TASKING

In this section, we explain MFEA as an island model for highlighting its features whereas it was implemented as a single-population model. For simplicity of explanations, we assume that we have two tasks (i.e. Task 1 and Task 2), which are simultaneously solved. In MFEA, a task is assigned to each individual depending on its relative strength for each task (i.e., its factorial rank for each task) in an initial population. Thus an initial population can be viewed as being divided into multiple islands (i.e., sub-populations). The number of islands is the same as the number of tasks to be solved. Under our assumption of two tasks, the population is divided into two islands.

Let us denote the population size by μ. From the current population with μ individuals, μ offspring are generated. The overall generation update mechanism of MFEA is the $(\mu + \mu)$ES style. An offspring population is generated from the current population. First, $\mu/2$ pairs of parents are randomly selected from the current population. If two parents are selected from the same island, two offspring are generated by crossover and mutation. The generated offspring are assigned to the same island as their parents. If two parents are selected from different islands, two offspring are generated by crossover and mutation with a special crossover probability p. In this case (i.e., when two offspring are generated by crossover and mutation), each offspring is assigned to an island based on its relative strength for each task. When crossover is not applied (which happens with the probability of $1 - p$), an offspring is generated from each parent by mutation. Then each offspring is assigned to the same island as its parent. In this manner, μ offspring are generated and distributed to the two islands. In each island, the best $\mu/2$ individuals are selected based on the fitness function values for the corresponding task. MFEA iterates the above operations until the termination condition is satisfied.

As an island model, MFEA has the following two features.

I. Crossover between individuals from different islands is performed in every generation. Each offspring generated by inter-island crossover is assigned to one island based on its relative strength for each task.

II. No migration is performed. Information in each island is transferred by inter-island crossover (not by migration).

These two features are clearly different from the standard island model where crossover is always performed within each island and migration is periodically performed. Moreover, MFEA also has the following feature:

III. Different crossover probabilities are used depending on the combination of two parents.

This is also clearly different from standard evolutionary algorithms where the same crossover probability is used for all pairs of parents.

3 PROPOSED ISLAND MODEL

In this section, we propose a simple island model for evolutionary multi-tasking. For simplicity of explanations, we assume that we have two tasks as in Section 2. In this model, $\mu/2$ initial individuals are randomly generated for each island..

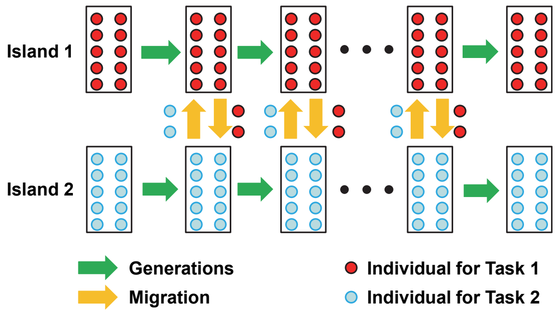


Figure 1 : A proposed island model.

In each island, $\mu/2$ offspring are generated by crossover and mutation from the current individuals. Each individual is always selected as a parent only once in each generation. The best $\mu/2$ individuals are separately selected from the parent and offspring populations in each island.

The following migration procedure is periodically performed: A pre-specified number of individuals are randomly selected from each island. At the same time, the same number of the bottom individuals are removed from each island. Then, each of the randomly selected individuals is copied and inserted to the other island. To avoiding the increase of computational costs, migrated individuals are not evaluated. The worst fitness value is assigned to the migrated individuals in each island. Thus those copies are removed through the next generation update. However, they are used as parents before the generation update since each individual is selected as a parent once. When the optimal solutions of two tasks are totally different, the fitness values of the migrated copies are likely to be bad. This is another reason for assigning the worst fitness value to all the migrated copies without recalculating their fitness values.

The proposed island model is illustrated in Fig. 1. This model has two parameters: migration interval T (i.e., migration is performed every T generations) and migration size N (i.e., copies of N individuals are migrated). Effects of these parameters on the performance of this model are examined in detail in the next section. In one extreme setting where the migration interval is the same as the total number of generations, the island model is the same as the independent run for each task in each island.

4 EXPERIMENTAL RESULTS

We apply the proposed simple island model to test problems which are used to evaluate MFEA in [2]. Each test problem is a pair of two well-known function minimization problems. These test problems are classified into nine categories based on the degree of the intersection of the global optima of the tasks and the levels of the similarity between them. We examine various settings of the parameters T and N in the proposed model and compare experimental results with those by MFEA. After computational experiments, we examine whether one model (i.e., the proposed island model and MFEA) is significantly better or worse than the other. The differences between the results are assessed by Welch's t-test or Brunner-Munzel test at the 0.05 significance level. Welch's t-test is utilized if the results follow a normal distribution, while Brunner-Munzel test is used if the results do not follow the normal distribution. If the difference of the results between MFEA and the proposed island model under a particular setting is statistically significant, the setting is highlighted by red or blue in the corresponding figures (See Fig. 2-5). Red represents that the proposed island model obtains better results than MFEA. Blue represents that MFEA obtains better results than the proposed island model.

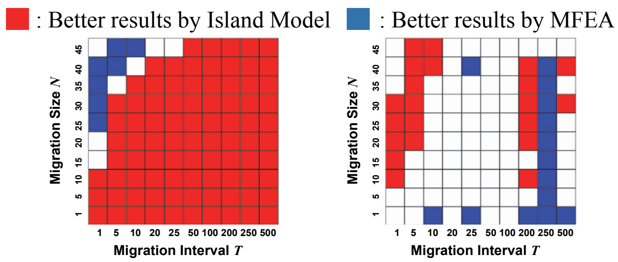
As in [2], our computational experiments are performed under the following parameter specifications:

Population size: 100 (50 for each sub-population),
Termination condition: 1000 generations,
Crossover: SBX (distribution index: 2),
Crossover probability: 1.0,
Special crossover probability in MFEA: 0.3,
Mutation: PM (distribution index: 5),
Mutation probability: 1/ (string length),
Migration interval: T = 1, 5, 10, 20, 25, 50, 100, 200, 250, 500,
Migration size: N = 1, 5, 10, 15, 20, 25, 30, 35, 40, 45.

Experimental results on CI+HS, CI+MS, CI+LS and PI+HS are shown in Figs. 2-5. Due to the page limitation, we do not show the experimental results on the rest of the test problems in [2]. Except for Ackley function on the CI+LS, similar or better results were obtained by the proposed simple island model than MFEA with a lot of settings especially small migration intervals. Similar or better results are also obtained by the proposed simple island models for many tasks in the other five test problems in [2]. In MFEA, some offspring are generated by mutation from parents because of the small probability (0.3) of the special crossover. In contrast, all the offspring in the proposed simple island model are generated by crossover and mutation. Thus, the proposed simple island model may be able to search the decision space more efficiently than MFEA. This may be a possible reason why the proposed simple island model can obtain better results than MFEA with many parameter specifications especially small migration intervals.

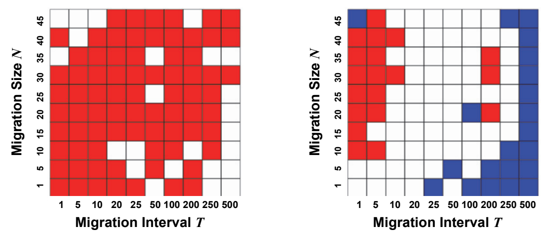
However, the proposed island model shows worse results than MFEA when the migration interval is very small and the migration size is very large. Under those extreme parameter specifications, many individuals are added from one island to the other (which means that many current individuals are removed). This clearly disturbs the evolution in each island towards the optimal solution of each task.

Fig. 2 (b) and Fig. 3 (b) show that better results are obtained from small migration intervals than large migration intervals in the proposed model. Similar results are observed in some other tasks such as Ackley and Rosenbrock functions on PI+MS. This may be because frequent migration with an appropriate migration size between two islands helps the search in each island to escape from local minima through maintaining a large diversity of the sub-population. That is, frequent migration may prevent each island from converging to a local minimum. This may be one of the main advantages of multi-tasking over single-tasking.



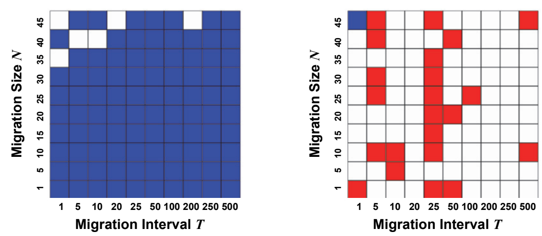
(a) Griewank function (b) Rastrigin function

Figure 2 : Experimental results on the CI+HS Problem.



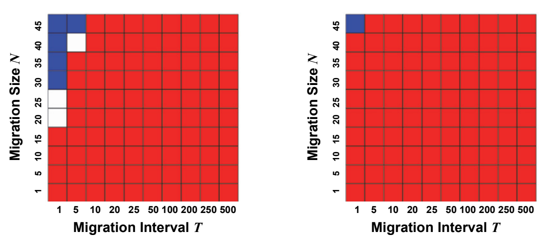
(a) Ackley function (b) Rastrigin function

Figure 3 : Experimental results on the CI+MS Problem.



(a) Ackley function (b) Schwefel function

Figure 4 : Experimental results on the CI+LS Problem.



(a) Rastrigin function (b) Sphere function

Figure 5 : Experimental results on the PI+HS Problem.

In Fig. 4 (a), Task 1 (i.e., Ackley function) in CI+LS has the following two features: i) It has the globally optimal solution near the edge of the search space (whereas Task 1 in CI+MS in Fig. 3 has the globally optimal solution at the center of the search space), and ii) similar fitness values are assigned to solutions except for a neighboring region of the globally optimal solution. In this situation, it is difficult for both algorithms to find the globally optimal solution of Task 1 in Fig. 4 (a). Better performance of MFEA in Fig. 4 (a) may be explained as follows. Whereas all offspring in the proposed island model are generated by crossover and mutation, some offspring in MFEA are generated by mutation. This property of MFEA may beneficial in efficiently finding better local solutions. This may be also beneficial in searching for the globally optimal solution near the boundary of the search space. To further examine this issue, we apply two single-tasking evolutionary algorithms to Task 1. One has only mutation. The other has both crossover and mutation. Better results are obtained from the mutation-only variant. We need to further examine the results in Fig. 4 (a).

5 CONCLUSIONS

In this paper, we first explained MFEA as a special island model where crossover is performed between islands and migration is not performed. Next, we proposed a simple implementation of evolutionary multi-tasking in the framework of the standard island model where crossover is performed within each island and migration is periodically performed. Then, we compared the proposed island model with MFEA. Experimental results showed that similar or better results were obtained by the proposed island model than MFEA over a wide range of parameter specifications especially small migration intervals. That is, it was confirmed that a positive effect of evolutionary multi-tasking can be obtained by periodical migration in the proposed model as well as crossover between individuals for different tasks in MFEA. This observation suggests that evolutionary multi-tasking is a general idea which can be implemented in various manners. This may means that the reported results of evolutionary multi-tasking in the literature [3, 6, 8, 11] may be improved by other implementations.

REFERENCES

[1] E. Cantú-Paz. 1998. A survey of parallel genetic algorithms, *Calculateurs Paralleles* 10, 2 (1998), 141-171.

[2] B. Da, Y. S. Ong, L. Feng, A.K. Qin, A. Gupta, Z. Zhu, C. K. Ting, K. Tang, X. Yao. 2017. Evolutionary multitasking for single-objective continuous optimization: benchmark problems, performance metric, and baseline results. arXiv preprint arXiv:1706.03470 (2017)

[3] J. Ding, C. Yang, Y. Jin, T. Chai. 2017. Generalized multi-tasking for evolutionary optimization of expensive problems, *IEEE Trans. on Evolutionary Computation* (Early Access Paper: Online Available from IEEE Xplore)

[4] M. Gorges-Schleuter. 1990. Explicit parallelism of genetic algorithms through population structures. In *Proceedings of International Conference of Parallel Problem Solving from Nature I* (PPSN 1990), Springer, Dortmund, Germany, 150-159.

[5] A. Gupta, Y. S. Ong, and L. Feng. 2016. Multifactorial evolution: Towards evolutionary multitasking. *IEEE Trans. on Evolutionary Computation* 20, 3(2016), 343-357.

[6] A. Gupta, Y. S. Ong, L. Feng and K. C. Tan. 2017. Multiobjective multifactorial optimization in evolutionary multitasking. *IEEE Trans. on Cybernetics* 47, 7(2017), 1652-1665.

[7] A. Gupta, B. Da, Y. Yuan and Y. S. Ong. 2017. On the emerging notion of evolutionary multitasking: A Computational analog of cognitive multitasking. *Recent Advances in Evolutionary Multi-objective Optimization*. New York, USA: Springer, 2017, 139-157.

[8] A. Gupta, Y. S. Ong, L. Feng. 2015. Evolutionary multitasking in bi-level optimization. *Complex & Intelligent Systems* 1, 1-4 (2015), 83-95.

[9] Y. S. Ong, A. Gupta. 2016. Evolutionary multitasking: A computer science view of cognitive multitasking. *Cognitive Computation* 8, 2 (2016), 521-544.

[10] L. Pan, C. He, C. He, Y. Tian, H. Wang, X. Zhang, and Y. Jin. 2018. A Classification Based Surrogate-Assisted Evolutionary Algorithm for Expensive Many-Objective Optimization. *IEEE Trans. on Evolutionary Computation* (Early Access Paper: Online Available from IEEE Xplore)

[11] Y. W, Wen and C. K. Ting. 2017. Parting ways and reallocating resources in evolutionary multitasking. In *Proceeding of 2017 Congress on Evolutionary Computation* (CEC 2017), IEEE, San Sebastián, Spain, 1295-1302.