2017 International Symposium on Nonlinear Theory and Its Applications,
NOLTA2017, Cancun, Mexico, December 4-7, 2017

Multi-Objective Particle Swarm Optimizer Networks with Tree Topology

Kyosuke MIYANO†, Hidehiro NAKANO† and Arata MIYAUCHI†

†Tokyo City University
1–28–1, Tamazutsumi, Setagaya-ku, Tokyo 158–8557, Japan
Email: {g1781528, hnakano}@tcu.ac.jp

Abstract—Multi-Objective Particle Swarm Optimizer (MOPSO) is a kind of metaheuristic algorithms for solving multi-objective optimization problems. In MOPSO, a global best solution set corresponding to the Pareto solution set is stored in an archive memory. Island-model MOPSO (IMOPSO) has a tree topology of sub-swarms; a upper layer sub-swarm search the Pareto solution set in the multi-objective function, while lower layer sub-swarms search the best solutions in each single objective function. IMOPSO can effectively search high-quality Pareto solution set. This paper investigates the performance of some migration strategies. Then, it is shown that a migration strategy between lower layer sub-swarms can provide the good search performance. In the simulation experiments, the results for some benchmark problems are shown.

1. Introduction

In various engineering systems, there are optimization problems in which the design parameters are optimized to obtain desired systems. The problems which have multiple objective functions are called multi-objective optimization problems. The multi-objective optimization problems require that all objective function values are optimized. However, each objective function often has a trade-off relationship with each other. Therefore, the Pareto front approach aims to obtain a solution set called Pareto optimal front solutions in which each solution is not inferior to each other for all objective function values. Particle Swarm Optimizer (PSO) is one of optimization methods [1]. In PSO, particles search solutions in a problem. Each particle has velocity and position information, and has a personal best solution found by the particle in the search process and a global best solution (gbest) among all particles as information shared in the swarm in the search process. PSO can fast solve various optimization problems by using simple operations. Multi-Objective Particle Swarm Optimizer (MOPSO) is the modified PSO to solve multi-objective optimization problems [2]. In the general MOPSO, gbest is selected from the gbest storage named archive at random. MOPSO has a sharing process that gives gbest diversity. However, there is a problem that the calculation amount is increased by the sharing processing. Therefore, Island-model MOPSO (IMOPSO) has been proposed to solve this problem [3]. IMOPSO is a parallelized model in which the particle swarm is divided into an island of MOPSO and plural islands of single-objective PSOs. In IMOPSO, each island searches solution by its own evaluation. Therefore, it is possible to maintain diversity of solutions without sharing processing, and to reduce the amount of calculation. As a problem, searching can sometimes stagnate by the rapid convergence of single-objective PSOs at an early stage. In this research, we add a migration topology between single-objective PSOs. In order to verify the effectiveness of the proposed method, we performs experiments with several representative benchmark functions.

2. Multi-objective optimization problem

In the optimization problem, the objective function is optimized according to defined constraints. The multi-objective optimization problem is formulated as Equation (1).

$$\begin{aligned} \textit{minimize} \quad & f(x) = (f_1(x), ..., f_M(x)) \\ & \textit{subject to} \quad x \in \mathbb{S} \subset \mathbb{R}^D \end{aligned} \tag{1}$$

where $f = (f_1, ..., f_M)$, $f : \mathbb{S} \to \mathbb{R}^M$ are M objective functions to be minimized at the same time, and $f_i : \mathbb{S} \to \mathbb{R}\,(i \in \{1, ..., M\})$ is each single objective function. $x = (x_1, ..., x_D)$ is the solution and $\mathbb{S} = \prod_{j=1}^{D}[a_j, b_j]$ is the solution space in the D dimensions.

When the objective functions are in a tradeoff relationship with each other in the multi-objective optimization problem, there is no solution capable of simultaneously optimizing all the objective functions. In the multi-objective optimization problems, the goal is to find a set of solutions that can not be superior or inferior to each other, called Pareto optimal solutions. The curve formed by the set of Pareto optimal solutions is called the Pareto front.

3. Island-model MOPSO

PSO is a kind of meta-heuristic algorithms that conduct multi-point search, and mimics social behavior of living organisms. In PSO, particles existing in solution space have position information and velocity information. Solutions are searched by moving particles in solution space. Each particle moves around in the search space, taking advantage of each own local best known position (pbest) , and is also guided toward the best known position (gbest) of the

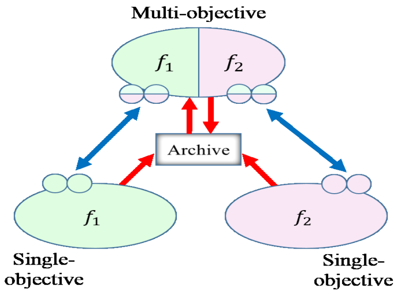

Figure 1: IMOPSO

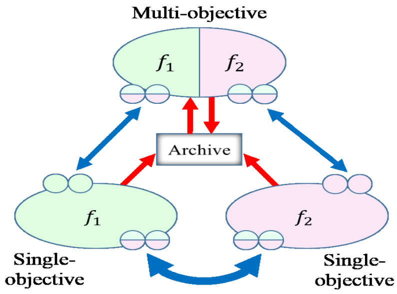

Figure 2: Proposed model

whole swarm. The position and velocity of each particle are updated by the following equation.

$$\begin{aligned} \boldsymbol{v}_{ij}^{k+1} = \boldsymbol{w}\boldsymbol{v}_{ij}{}^{k} + c_1 rand_1(\boldsymbol{pbest}_{ij}^{k} - \boldsymbol{x}_{ij}^{k}) \\ + c_2 rand_2(\boldsymbol{gbest}_{j}^{k} - \boldsymbol{x}_{ij}^{k}) \end{aligned} \tag{2}$$

$$\boldsymbol{x}_{ij}^{k+1} = \boldsymbol{x}_{ij}^{k} + \boldsymbol{v}_{ij}^{k+1} \tag{3}$$

where x is a particle's position, v is a particle's velocity, i is the particle number, j is the ingredient of a variable vector, $rand_1, rand_2$ are uniform random numbers for [0,1], w is an inertia coefficient, and c_1, c_2 are weight coefficients. On the other hand, MOPSO has an archive mechanism that stores multiple gbests. When updating the velocity, one of the stored gbests is assigned randomly to each particle for each iteration.

In the IMOPSO, the island model with a two-layer tree structure is used. Single-objective islands are located at lower hierarchy, and a multi-objective island is located at upper hierarchy. A conceptual diagram of IMOPSO is shown in Fig.1. Each island of the lower hierarchy evaluates each single objective function and it does not evaluate other objective functions. Also, since it is similar to the conventional PSO, each island of the lower hierarchy has independent and unique gbest, and only updates the archive for the island of the upper hierarchy. On the island of the upper hierarchy, all objective functions are evaluated using the Pareto ranking. Also, this island refers to and updates the archive. In IMOPSO, it is possible to search more efficiently by migrating particles between the upper layer island and the lower layer islands in every certain period. With such a network topology, it is possible to efficiently perform global search and local search, and to maintain diversity of solutions.

4. Proposed Algorithm

In this paper, we propose a model with the migration topology between each lower layer island. The archive is common for each island. It is referred only by the upper layer island, and is updated by both upper and lower layer islands. A conceptual diagram of the proposed algorithm is shown in Fig.2. In the proposed algorithm, by the migration between lower layer islands, a particle group optimizing an objective function can obtain particles from the other particle groups optimizing different objective functions. The migrating particles move toward gbest of the new particle group. Then, these particles can search a variety of solutions along Pareto front. By using the proposed network topology, local search and global search can be performed in the lower layer. Therefore, we can expect to effectively find many Pareto optimal solutions.

5. Simulation Experiments

In the experiments we used the ZDT benchmark problems [4] and the DTLZ benchmark problems [5]. In the ZDT benchmark problems, comparison experiments were conducted with IMOPSO (Conventional) and proposed algorithm (Proposed). In the DTLZ benchmark problems, we conducted comparison experiments with IMOPSO, the proposed algorithm, and a three-layer tree structure model (3 layer). A conceptual diagram of the three-layer tree structure model is shown in Fig.3. The three-layer tree structure model is a model in which each two-objective MOPSO is added between the upper layer and the lower layer with respect to the three objective problems. We use Generational Distance (GD) and Inverted Generational Distance (IGD) as performance metrics [6]. They calculate the distances between the Pareto front obtained by each algorithm and PF_{true}; the known true Pareto front.

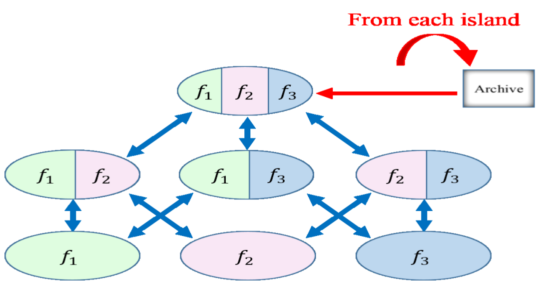

Figure 3: Three-layer tree structure model

Table 1: Experiment parameters

Number of particles	ZDT4	$450(150 \times 3)$
	ZDT6	$600(200 \times 3)$
	DTLZ2	$600(150 \times 4)$
	DTLZ7	
Number of generations		1×10^5
Number of dimensions		10
w		0.9
c_1, c_2		1.0
Archive size		900

Table 2: Experiment result

	Algorithm	GD	IGD
ZDT4	Conventional	0.021351	0.003641
	Proposed	0.000762	0.002033
ZDT6	Conventional	0.007066	0.014386
	Proposed	0.003848	0.001076
DTLZ2	Conventional	0.017757	0.025835
	Proposed	0.013257	0.026907
	3 layers	0.051723	0.040915
DTLZ7	Conventional	2.036947	0.346174
	Proposed	0.288848	0.064085
	3 layers	1.007784	0.237595

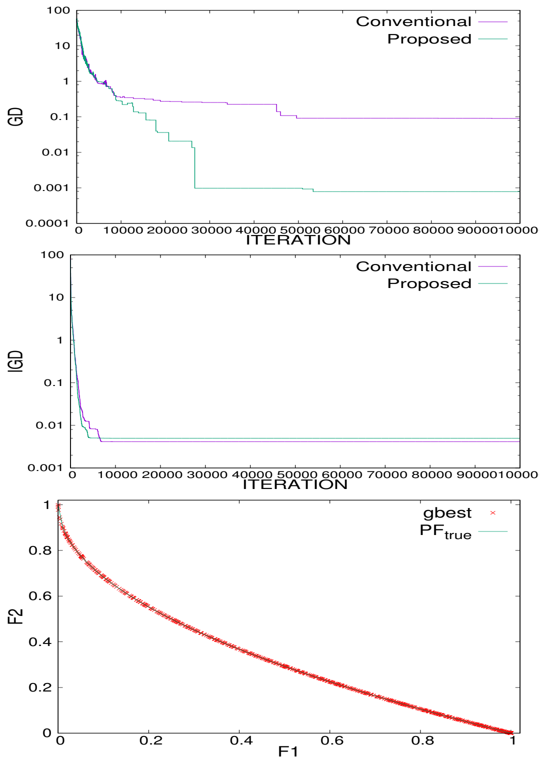

Figure 4: ZDT4

The parameters used in the experiments are shown in Tables 1 and 2. The number of migrating particles between the upper and lower layers was set to 12, and the migration interval was set to every 20 generation. The number of migrating particles between the lower layers in the proposed algorithm was set to 20, and the migration interval was set to every 100 generation. We executed 30 trials in each ZDT benchmark problem and 10 trials in each DTLZ benchmark problem, and the average values were taken as the experimental values. Table 3 shows the experimental results. Figure 4-7 show transition diagrams of *GD* and *IGD* for each generation and the particles of the archive at the final generation in the proposed algorithm.

The proposed algorithm is equivalent to or superior to the conventional algorithm in most test problems. ZDT4 has a multimodal characteristic, and in the conventional method, it falls into local solutions, which form a Pareto front far from PF_{true}. However, we can see that the proposed algorithm was able to form a Pareto front near PF_{true}. ZDT6 has a bias between a decision variable and an objective function, and it was able to approach PF_{true} in both the conventional algorithm and the proposed algorithm. However, it should be noted that the proposed algorithm shows higher IGD than the conventional algorithm. DTLZ7 is a matter of three objectives, and has discontinuous PF_{true}. From the values of *GD* and *IGD*, performance improvement was seen in the proposed algorithm. The three-layer tree structure model exceeded IMOPSO in DTLZ7. However, in another three objective problem (DTLZ2), it was inferior to the other two methods. For this reason, even if simply increasing the number of layers, improvement of performance can not be expected. From these results, it can be seen that with the proposed method, it is possible to promote Pareto solution search by lower layers and to escape from local solutions by adding migration between each lower layer.

6. Conclusion

In this paper, we proposed a model with a migration strategy between lower layer PSOs in two-layer tree structure. As a result of comparative experiments between proposed algorithm and conventional algorithms, the proposed algorithm can find Pareto optimal solutions with higher accuracy than the conventional algorithms. In addition, even when the number of objective functions increased, search performance was improved. In other words, by simply adding a simple particle migration network, it was possible to improve the search performance without applying a large calculation cost. As a future task, the proposed algorithm is applied to more benchmark problems. Especially, it is necessary to verify the performance when the number of objectives and decision variables increases.

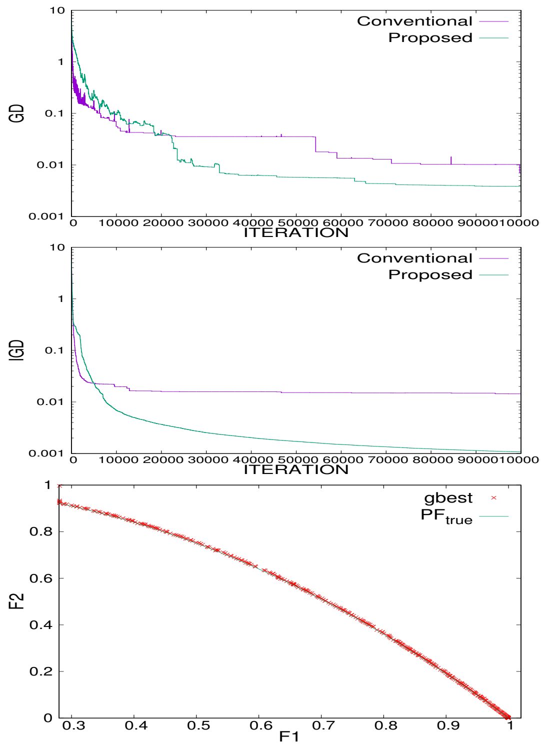

Figure 5: ZDT6

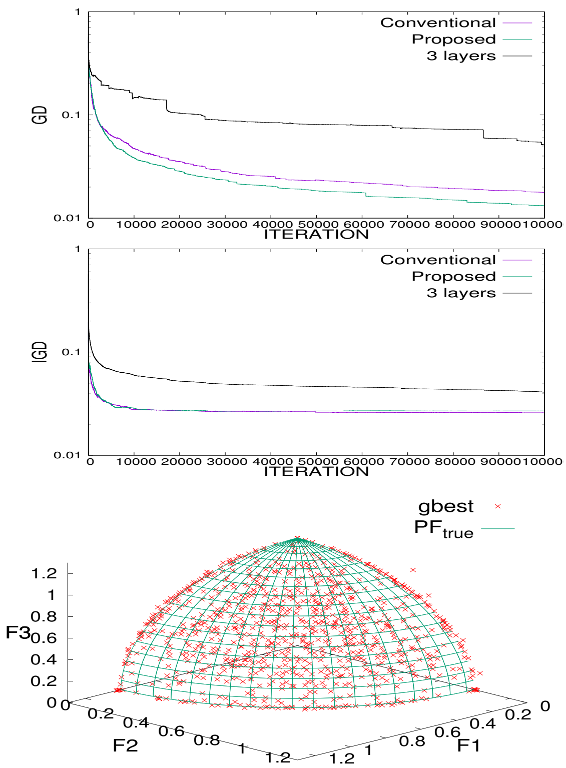

Figure 6: DTLZ2

References

[1] J.Knennedy and R.c. Eberhart, “Particle Swarm Optimization”, Proc.IEEE International Conference on Neural Networks, pp.1942-1948,1995

[2] M.Reyes-Sierra and C.A. Coello Coello, "Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art", Int. J. Computational Intelligence Research, 2, 3, 287-308(2006).

[3] N.Murofushi, H.Nakano and A.Miyauchi, "A Distributed Particle Swarm Optimizer with a Tree Network Topology for Multi-Objective Optimization Problems", Proc.NOLTA, 158-161(2012).

[4] E. Zitzler, K. Deb, L. Thieler, "Scalable Test Problems for Evolutionary Multiobjective Optimization", IEEE Trans. Evol. Computation, 8, 2, 173-195, (2000).

[5] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems", Evolutionary Multiobjective Optimization, Springer, 105-145 (2005).

[6] K. C. Tan, T. H. Lee, Y. J. Yang, and D. S. Liu, "A Co-operative Coevolutionary Algorithm for Multiobjective Optimization", IEEE international Conference on Systems, Man and Cybernetics, pp. 1926-1931, 2004.

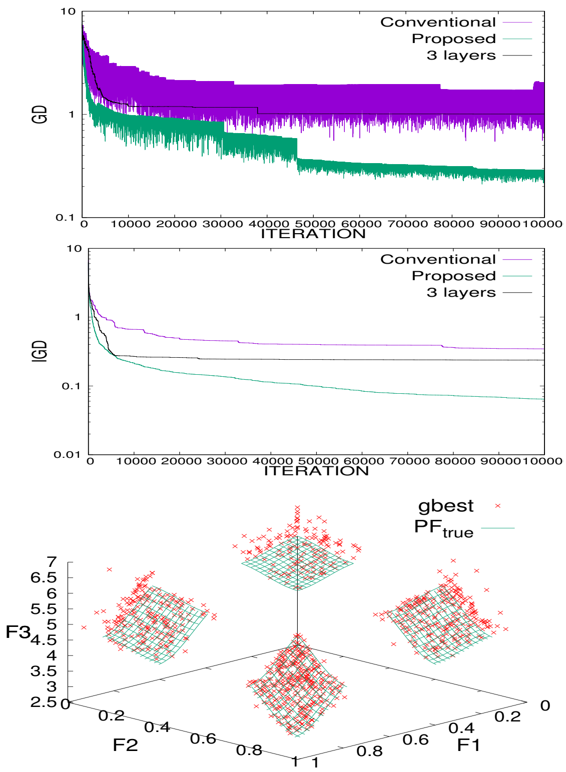

Figure 7: DTLZ7