IEICE Proceeding Series

A Distributed Particle Swarm Optimizer with a Tree Network Topology for Multi-Objective Optimization Problems

Noriyuki Murofushi, Hidehiro Nakano, Arata Miyauchi

Vol. 1 pp. 158-161
Publication Date: 2014/03/17
Online ISSN: 2188-5079

2012 International Symposium on Nonlinear Theory and its Applications
NOLTA2012, Palma, Majorca, Spain, October 22-26, 2012

A Distributed Particle Swarm Optimizer with a Tree Network Topology for Multi-Objective Optimization Problems

Noriyuki Murofushi†, Hidehiro Nakano† and Arata Miyauchi†

†Tokyo City University
1-28-1, Tamazutsumi, Setagaya-ku, Tokyo, 158-8557 Japan
Email: {murofushi, nakano, miyauchi}@ic.cs.tcu.ac.jp

Abstract—This paper presents a distributed particle swarm optimizer for multi-objective optimization problems. In the proposed method, multiple subswarms construct a two-layered tree topology. The subswarms in the lower layer search local solutions for a part of the objective functions, and the subswarm in the higher layer searches Pareto front solutions for all objective functions. Some particles migrate between these layers with a constant interval. The proposed algorithm is simple and requires low computation cost. Some simulation results are presented.

1. Introduction

In various engineering systems, there are optimization problems in which the design parameters are optimized to obtain feasible 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, generally, trade-off characteristics between each objective function exist. Therefore, it is needed to obtain a variety of feasible solutions as candidates to actual design. Many methods to solve the multi-objective problems have been proposed, e.g., multi-objective genetic algorithms and multi-objective particle swarm optimizers [1]- [4]. 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. For the evaluation of each individual, the Pareto ranking method is often used [2]. On the other hand, in order to keep the diversity of each solution, the sharing function method is known as an effective method [3]. In the method, the degree of congestion of each individual is computed. Then, solutions with diversity can be obtained. However, this method requires high computation cost.

In this paper, we propose a simple algorithm for multi-objective optimization problems based on Particle Swarm Optimizers (PSOs, [5]). In PSO, particles search solutions in a target 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 among all particles as information shared in the swarm in the search process. PSO can fast solve various optimization problems by using simple operations. The proposed algorithm introduces a two-layered subswarm structure [4]. The low layer consists of multiple subswarms corresponding to the number of objective functions. Each subswarm finds a solution to each objective function. The high layer is a single subswarm which finds Pareto optimal front solutions. These subswarms use archive and grid schemes to store Pareto front solutions and to share them as a global best solution set in each subswarm. Between the subswarms of the high and low layers, some particles migrate periodically. The proposed algorithm is simple and requires low computation cost. For the benchmark problems, the numerical simulations are performed and the effectiveness of the proposed algorithm can be verified.

2. Conventional Algorithm

Particle Swarm Optimization (PSO) is a kind of metaheuristic algorithms emulating actions in swarms such as birds and fishes. Each particle moves around in the search space, taking advantage of the particle's local best known position (pbest) , and is also guided toward the best known positions (gbest) of the whole swarm. Then, velocity and position vectors of each particle are updated. The update equations are as follows.

$$v_{ij}^{k+1} = wv_{ij}^k + c_1 rand_1(pbest_{ij}^k - x_{ij}^k) + c_2 rand_2(gbest_j^k - x_{ij}^k) \quad (1)$$

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \quad (2)$$

where x is a particle's position, v is a particle's velocity, and i is particle number, j is the ingredient of a variable vector, $rand_1$ and $rand_2$ are uniform random numbers for [0,1], w is an inertia coefficient, and c_1 and c_2 are weight coefficients. In the general Multi-objective PSO(MOPSO), $gbest^k$ is selected from the gbest storage at random [1]. Since plural objective functions exist in multi-objective optimization problems, plural optimum solutions (Pareto optimal solutions) can exist, where each Pareto optimal solution is not inferior to each other for all objective function values. In multi-objective optimization, the fitness of particles is given by Pareto ranking. A rank of a particle corresponds to the number of particles by which it is dominated. The rank of the Pareto optimal solution is 1. Furthermore, in order to find more various Pareto optimal solu-

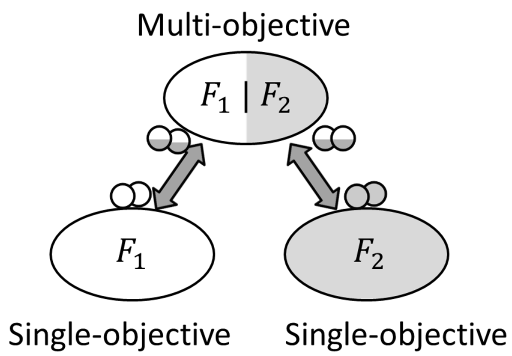

Figure 1: The island model with a two-layered tree structure

tions (Pareto optimal front), a congestion degree is adopted to evaluate solutions. Then, the fitness of the place where many particles gather becomes low. The most common congestion degree calculation method is the sharing function method. Fitness F_i' corrected by a sharing function from the Fitness F_i is given by the following equation.

$$F_i{}' = \frac{F_i}{\sum_{j=1}^{n} sh(d(i,j))} \tag{3}$$

where i and j are particle numbers, n is the number of particles, and sh is a sharing function which determines how dense the solution is. The following equation for the sharing function is used.

$$sh(d(i,j)) = \begin{cases} 1 - \frac{d(i,j)}{\sigma_{share}} & if \;\; d(i,j) < \sigma_{share} \\ 0 & otherwise \end{cases} \tag{4}$$

where $d(i,j)$ is the Euclidean distance in objective space between the i-th and j-th particles. Niche radius σ_{share} is a sharing parameter. Evaluation is lowered when the distance between particles is smaller than the niche radius σ_{share}. However, this method requires high computation cost.

3. Proposed Algorithm

To reduce calculation cost, in the proposed method, the island model with a two-layered tree structure is used. As shown in Fig. 1, single-objective islands are located at low hierarchy, and a multi-objective island is located at high hierarchy. Each island of the low hierarchy evaluates each single objective function and it does not evaluate other objective functions. On the island of the high hierarchy, all objective functions are evaluated using the Pareto ranking. Some particles migrate between these layers with a constant interval. Such a hierarchy structure can perform both global search and local search better than the network structure in which the same multi-objective islands are evenly connected.

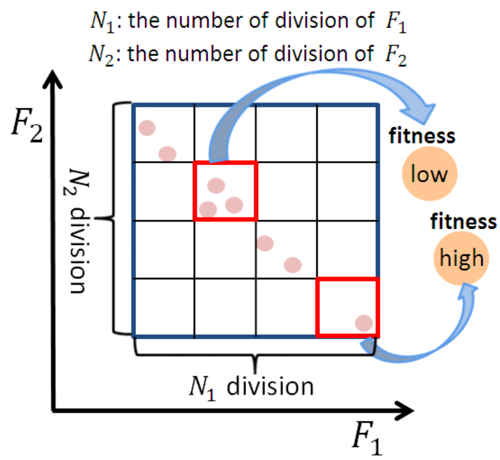

Figure 2: Grid and archive schemes

In addition, in order to obtain a solutions with diversity, the following schemes are introduced. Solution space is divided as grids. The maximum and the minimum of each objective function value are calculated from particles. The region for the i th objective function F_i is divided by N_i. The conception diagram is shown in Fig. 2.

If all the Pareto optimum solutions found by search are stored as a gbest set, large memory resource is required. So, it is necessary to decide the capacity of gbest storage. A particle newly selected as a candidate gbest is compare with the particles already stored as the gbest set, and the particle with the Pareto rank 1 is stored. If the acceptable number which can be stored is exceeded, a solution in a grid where particles are dense is eliminated based on the evaluation by using the density degree. Then a variety of solutions can be obtained without using the sharing function.

4. Simulation Experiments

To compare the proposed method with the general MOPSO using the sharing function method, the benchmark selected from the ZDT set [6] are used. All the problems have two objective functions.

To measure the performances of each method quantitatively, the following two metrics are used.

- Generational Distance (GD) [7]

$$GD = \frac{1}{n}\sqrt{\sum_{i=1}^{n} d_i^2} \tag{5}$$

where n is the number of solutions included in the set of solutions, d_i is the Euclidean distance in objective space between the i th particle and the nearest member in the set of

Table 1: Common parameters in the simulation experiments.

	proposed method	conventional method
w	0.9	
c_1 , c_2	1.0	
# of islands	3	1
Migration interval	20	-
# of migration particles	1	-
# of dimensions	10	

Pareto optimal solutions. This metric shows the mean distance between the found Pareto front and the actual Pareto optimal front.

- Spacing (S) [7]

$$S = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i - \overline{d})^2} \;/\; \overline{d} \tag{6}$$

where $\overline{d} = \frac{1}{n}\sum_{i=1}^{n} d_i$. d_i is the Euclidean distance in objective space between the i th particle and its nearest member in the particles. This metric measures the uniformity in the found Pareto front.

The performances about the proposed method and general MOPSO are compared through simulation experiments. Experimental parameters are shown in Tables 1 and 2. All the test programs are performed 10 times, the best results are shown in Table 3. The Pareto front solutions in the proposed method at the last generation are shown in Figs. 3-7.

The results of the proposed method are almost the same or better than those of the conventional method in most benchmarks. Especially, much better performances can be verified in ZDT6. ZDT6 has the large deviations of distribution between decision variables and objective function values. In ZDT6, it is difficult for the conventional method to evenly search solution space. However, the proposed method with a two-layered tree structure can search wider solution space than the conventional method. In ZDT4, the value of GD in the proposed method is lower than that in the conventional method. However, as shown in Fig.6, some particles in the proposed method can reach the Pareto optimum front, although other particles cannot. On the other hand, we have confirmed that all particles in the conventional method cannot reach the Pareto optimum front. This result means that the proposed method is not inferior to the conventional method in this problem.

The proposed algorithm is simple and requires low computation cost. The effectiveness of the proposed method can be verified.

Table 2: Case parameters in the simulation experiments.

	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
# of particles	102 (3×34)	102 (3×34)	300 (3×100)	501 (3×167)	102 (3×34)
# of iterations	1×10^4	2×10^3	1×10^4	1×10^5	1×10^4
# of divisions	N_1=15, N_2=15	N_1=10, N_2=10	N_1=10, N_2=18	N_1=17, N_2=17	N_1=17, N_2=10
σ_{share}	0.003	0.12	0.002	0.1	0.2

Table 3: Comparison of performances on test problems

problem	method	GD	S
ZDT1	conventional method	0.002561	1.305792
	proposed method	0.000198	0.566345
ZDT2	conventional method	0.009106	1.620482
	proposed method	0.005435	1.243906
ZDT3	conventional method	0.014512	2.048453
	proposed method	0.014549	1.413845
ZDT4	conventional method	0.077915	5.912123
	proposed method	0.309881	2.610747
ZDT6	conventional method	0.053571	5.081610
	proposed method	0.002070	0.716897

5. CONCLUSIONS

This paper has proposed the island-type MOPSO with two-layered tree structure. This method also introduces grid and archive schemes as the congestion degree calculation. The proposed method has been evaluated on five test problems currently used in the literature. As a result, the better dispersed Pareto optimum solutions are found without using the sharing function method. That is, the proposed method can search optimal solutions without requiring high calculation cost.

In the future, we plan to test the algorithm proposed in

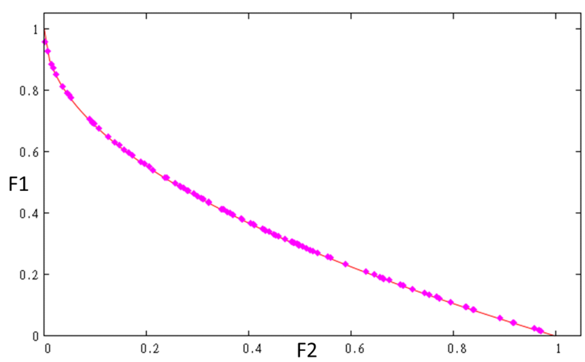

Figure 3: ZDT1

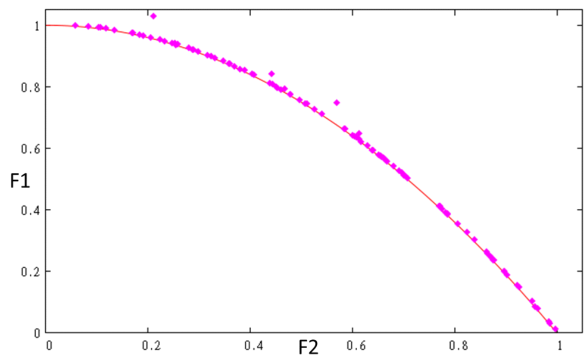

Figure 4: ZDT2

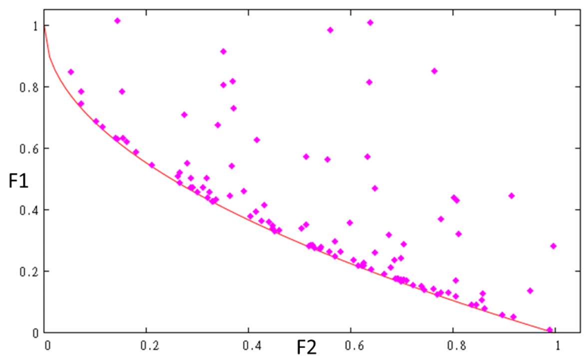

Figure 6: ZDT4

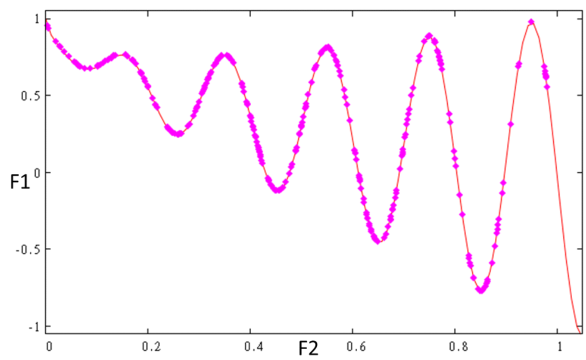

Figure 5: ZDT3

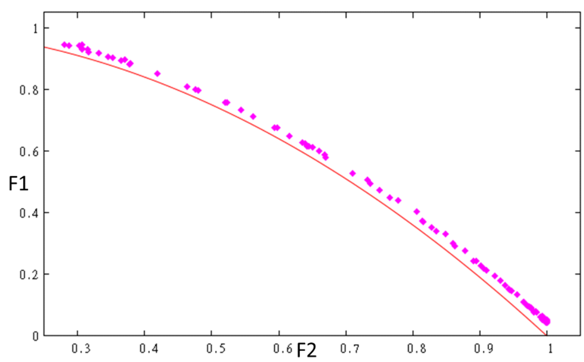

Figure 7: ZDT6

this work on a wider set of benchmark problems. In applying to a problem especially with many objective functions, it will become an important point how single-objective and multi-objective islands are built.

References

[1] 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*, vol. 2, no. 3, pp. 287-308, 2006.

[2] C.M.Fonseca and P.J.Fleming, "Genetic algorithms for multiobjective optimization," *Proc. of the Fifth International Conference on Genetic Algorithms*, pp.416–423, 1993.

[3] D.E.Goldberg and J.Richardson, "Genetic algorithms with sharing for multi-modal function optimization," *Proc. of the Second International Conference on Genetic Algorithms*, pp.41–49, 1987.

[4] K.Shibata, H.Nakano, and A.Miyauchi, "A learning method for dynamic Bayesian network structures using a multi-objective particle swarm optimizer," *Artificial Life and Robotics*, vol.16, No.3, pp.329–332, 2011.

[5] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," *Proc. of IEEE Int. Conf. on Neural Networks*, pp. 1942-1948, 1995.

[6] K.Deb, L.Thiele, M.Laumanns, and E.Zitzler, "Scalable Test Problems for Evolutionary Multi-Objective Optimization,“ *Evolutionary multi-objective Optimization:Theoretical Advances and Applications, A.Abraham, R.jain, and R.Goldberg, Eds.*, pp.105–145, 2005.

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