A Decomposition-based Multi-objective Particle Swarm Optimization Algorithm for Continuous Optimization Problems

Wei Peng
School of Computer
National University of Defense Technology
Changsha, Hunan, 410073, China
wpeng@nudt.edu.cn

Qingfu Zhang
Department of Computing and Electronic Systems
University of Essex
Colchester, Essex, CO4 3SQ, UK
qzhang@essex.ac.uk

Abstract

Particle Swarm Optimization (PSO) is a heuristic optimization technique that uses previous personal best experience and global best experience to search global optimal solutions. This paper studies the application of PSO techniques to multi-objective optimization using decomposition methods. A new decomposition-based multi-objective PSO algorithm is proposed, called MOPSO/D. It integrates PSO into a multi-obejective evolutionary algorithm based on decomposition (MOEA/D). The experimental results demonstrate that MOPSO/D can achieve better performance than a well-known MOEA, NSGA-II with differential evolution (DE), on most of the selected test instances. It shows that MOPSO/D will be a competitive candidate for multi-objective optimization. [1]

1. Introduction

A multi-objective optimization problem (MOP) can be stated as follows:

$$\begin{aligned} &\text{minimize} \quad F(x) = (f_1(x), \ldots, f_m(x)) \\ &\text{subject to} \quad x \in \Omega \end{aligned} \tag{1}$$

where Ω is the decision (variable) space, R^m is the objective space, and $F : \Omega \rightarrow R^m$ consists of m real-valued objective functions. Normally, based on the concept of Pareto dominance, the set of optimal solutions of a MOP in objective space is called the Pareto front (PF), and the corresponding set in decision space is called the Pareto set (PS). Most multi-objective evolutionary algorithms (MOEAs) are to find a set of representative Pareto solutions to approximate the whole PF. During the past decade, many multi-objective Particle Swarm Optimization algorithms (MO-PSOs) have been proposed. A good survey in this area can be found in [1], and the majority of MO-PSOs are based on Pareto dominance.

A Pareto optimal solution to a MOP could be an optimal solution of a scalar optimization problem in which the objective is an aggregation function of all individual objectives. Thus, a MOP can be decomposed into a number of single objective optimization subproblems. This is the basic idea of the aggregating or decomposition approach, and some MOEAs and MO-PSOs have been proposed along this line. The decomposition approach has many attracting advantages, e.g., simplicity, easiness to incorporate local search methods, and so on.

This paper proposes a MO-PSO algorithm using the decomposition approach, named MOPSO/D. It is based on a newly-published algorithm framework, called MOEA/D [2]. MOEA/D attempts to optimize all subproblems simultaneously. A neighborhood relationship is defined based on distances between weight vectors of different subproblems. Each subproblem is optimized by using information only from its neighboring subproblems. With the successful application of MOEA/D in continuous global optimization [2][3], it will be a promising approach by incorporating PSO techniques into the MOEA/D framework. In the proposed MOPSO/D algorithm, the selection of global best and personal best is studied. Then the performance of the proposed algorithm is experimentally studied and compared with NSGA-II [4], a popular Pareto dominance based MOEA.

2. MOPSO/D: A Multi-objective PSO Algorithm based on Decomposition

A general framework of MOEA/D has been proposed in [2]. In MOEA/D, a MOP is decomposed into many single objective subproblems. Each subproblem is defined by one

[1] This work was carried out during the first author's visiting at University of Essex as an academic visitor.

unique weight vector. With the Tchebycheff decomposition method, each subproblem can be stated as:

$$\begin{array}{ll} \text{minimize} & g(x|\lambda, z^*) = max_{1\leq i\leq m}\{\lambda_i\,|f_i(x) - z_i^*|\} \quad (2) \\ \text{subject to} & x \in \Omega \end{array}$$

where $\lambda = (\lambda_1, \ldots, \lambda_m)^T$ is a weight vector satisfying $\lambda_i \geq 0, 1 \leq i \leq m$ and $\sum_{i=1}^{m} \lambda_i = 1$. $z^* = (z_1^*, \ldots, z_m^*)^T$ is a reference point. The objective normalization technique [2] can also be used to improve the algorithm performance.

Each element of a weight vector used in MOEA/D takes the value from $\{\frac{0}{H}, \frac{1}{H}, \ldots, \frac{H}{H}\}$. Therefore, there are $N = C_{H+m-1}^{m-1}$ different weight vectors for a MOP with m objectives [2]. These uniformly-divided weight vectors work well for many MOPs, while dynamically-changing weight vectors can still be used under the MOEA/D framework.

Like MOEA/D, each particle in MOPSO/D will carry one unique weight vector. Therefore, each particle has an unique search direction defined by its weight vector, not like the PSO algorithms for single objective optimization.

In PSO algorithms, each particle's new velocity is calculated by using its current velocity, its current position and its personal best (pbest) and the global best (gbest). The personal best represents the previous best position of a particle. In MOPSO/D, it has the same definition with regard to the aggregating function. In contrast, the selection of global best need be considered carefully since each particle is associated with a different weight vector and a different subproblem. With the notion of neighborhood introduced in MOEA/D, we can define the global best as the best position among those that particles in the neighborhood of a particle have traversed.

At each iteration t, the MOPSO/D algorithm maintains the following data structures:

- a swarm of N particles $x^1, \ldots, x^N \in \Omega$, where each particle has a position vector, a personal best vector and a velocity vector, e.g., p^i, $pbest^i$ and v^i for the i-th particle. N is also the number of subproblems;
- an population of global best vectors for all particles, e.g., $gbest^i$ for the i-th particle or the i-th subproblem;
- a reference point $z = (z_1, \ldots, z_m)^T$;
- an external archive EA, which is used to store non-dominated solutions found during the search process.

Similiar to the MOEA/D framework, the MOPSO/D algorithm can be stated as follows.

Step 1 Initialization

Step 1.1 Set $EA = \emptyset$.

Step 1.2 Compute the Euclidean distances between any two weight vectors. For each $i = 1, \ldots, N$, set $B(i) = \{i_1, \ldots, i_T\}$ where $\lambda^{i_1}, \ldots, \lambda^{i_T}$ are the T closest weight vectors to λ^i. T is called as neighborhood size.

Step 1.3 Generate an initial swarm $x^1, \ldots, x^N$ randomly or by a problem-specific method. The velocity of each particle is set to zero initially.

Step 1.4 Initialize the reference point z by a problem-specific method.

Step 1.5 Calculate the initial personal bests and global bests. Set $pbest^i = gbest^i = p^i$ for each $i = 1, \ldots, N$.

Step 2 Evolve

For $i = 1, \ldots, N$, do

Step 2.1 Fly: Calculate the velocity of the i-th particle using its global best and personal best. Update the position of the i-th particle.

Step 2.2 Improvement: If the distance from the new position to the old position is less than a threshold α, then apply a problem-specific repair/improvement heuristic on p^i.

Step 2.3 Update of z: For each $j = 1, \ldots, m$, if $z_j < f_j(p^i)$, then set $z_j = f_j(p^i)$.

Step 2.4 Update of personal best: If $g(p^i|\lambda^i) \leq g(pbest^i|\lambda^i)$, then set $pbest^i = p^i$.

Step 2.5 Update of global bests: For each index $j \in B(i)$, if $g(p^i|\lambda^j) \leq g(gbest^j|\lambda^j)$, then set $gbest^j = p^i$.

Step 2.6 Update of EA: Remove from EA all the vectors dominated by $F(p^i)$. Add $F(p^i)$ to EA if no vectors in EA dominate it.

Step 3 Stopping If stopping criteria is satisfied, then stop and output EA. Otherwise go to **Step 2**.

The velocity and the new position of the i-th particle is calculated just like the traditional PSO algorithms:

$$v_j^i = w_0 v_j^i + c_1 r_1 (pbest_j^i - p_j^i) + c_2 r_2 (gbest_j^i - p_j^i) \quad (3)$$

$$p_j^i = p_j^i + v_j^i \quad (4)$$

where w_0 is inertia factor, c_1 and c_2 are two coefficients. r_1 and r_2 are two random numbers in the range of $[0, 1]$.

Research works have shown that turbulance on new positions of particles will benefit the algorithm on converging to global optima. Therefore, we apply the polynomia mutation [3] on positions after they are calculated.

When the new position of a particle is very close to its old position, it is very possible that it has fallen into or is close to a local optimum. We use a simple heuristic operation to improve the solution's quality or help it to step out

Table 1. Parameter Setting of Experiments

N: Population size	100
External archive size	500
Max function evaluation number	5e+5
Number of runs for each instance	20
T: Neighborhood size of MOPSO/D	30
Parameter CR of DE	0.1
Parameter SR (or F) of DE	0.5
Parameter p_m of polynomia mutation	0.05
Distribution index η of polynomia mutation	20
w_0: inertia factor of MOPSO/D	0.4
c_1, c_2: coefficients of MOPSO/D	2.0
α: threshold in MOPSO/D	1e-6
K: parameter in MOPSO/D	1000

the local optimum. We apply the polynomia mutation on the particle's position for K times and update the related data structures using the procedure from Step 2.3 to Step 2.6.

3 Experiments

We compare the MOPSO/D to the popular NSGA-II algorithm [4] with the differential evolution (DE) operator. DE is a heuristic which is simple but very fast and robust in numerical optimization. Table 1 gives the parameter setting of MOPSO/D and NSGA-II in the experiments.

The inverted generational distance (IGD) [6] is used to compare the performance of the two algorithms.

Nine bi-objective test instances are selected. They are F1, F2, F3, F4, F5 from [3] and S_ZDT1, S_ZDT2, S_ZDT4, S_ZDT6 from [5]. The test instances F1-F5 have the same convex PF shape, but their PS shapes are different nonlinear curves in the decision space. Figure 1 and Figure 2 show the evolutions of IGD values and the Pareto fronts with the lowest IGD values for instances F1, F2, F3, S_ZDT2, S_ZDT4, and S_ZDT6, respectively.

The figures show that the proposed MOPSO/D algorithm has better performance than NSGA-II on all test instances except F2. Besides, MOPSO/D achieves good results using only 5000 function evaluation numbers on these instances.

Table 2 gives the minimal and average IGD values obtained by the studied algorithms in 20 runs. MOPSO/D achieves the best performance results in all test instances except F2. The above experimental results verify that the MOPSO/D algorithm is a competitive candidate for some global optimization problems.

4 Conclusion

In this paper, a new multi-objective evolutionary algorithm named MOPSO/D is proposed. It integrates the PSO technique with the MOEA/D algorithm framework so that it can benefit from both. The experimental results verify the superior of MOPSO/D to NSGA-II on some test instances. The future work includes the improving of MOPSO/D and comparing of MOPSO/D to other MOEAs on other test problems. Besides, new mechanisms should be developed with regard to different characteristics of different optimization problems.

Table 2. Minimal and Average IGD metrics

Test Instance	MOPSO/D Minimal	NSGA-II Minimal	MOPSO/D Average	NSGA-II Average
F1	**0.001432**	0.038370	**0.002039**	0.046129
F2	0.037471	0.029064	0.066536	0.049645
F3	**0.013406**	0.035365	**0.023867**	0.042998
F4	**0.009811**	0.035045	**0.019007**	0.045003
F5	**0.013846**	0.034713	**0.023374**	0.039418
S_ZDT1	**0.001980**	0.009633	**0.003097**	0.011782
S_ZDT2	**0.001641**	0.014171	**0.001901**	0.016306
S_ZDT4	**0.026457**	0.646905	**0.141315**	1.499477
S_ZDT6	**0.062729**	1.484551	**0.088629**	1.572417

References

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

[2] Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Trans. on Evol. Comp., vol.11, pp.712-731, 2007.

[3] H. Li and Q. Zhang. Comparison Between NSGA-II and MOEA/D on a Set of Multiobjective Optimization Problems with Complicated Pareto Sets. IEEE Trans. on Evol. Comp., accepted, 2008.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Trans. on Evol. Comp., vol.6, 2002, pp.182-197.

[5] V. L. Huang, A. K. Qin, K. Deb, et al. Problem definitions for performance assessment on multi-objective optimization algorithms. Nanyang Technological University, Singapore, 639798, Tech. Report, Jan. 2007.

[6] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans. on Evol. Comp., 7(2):117-132, 2003.

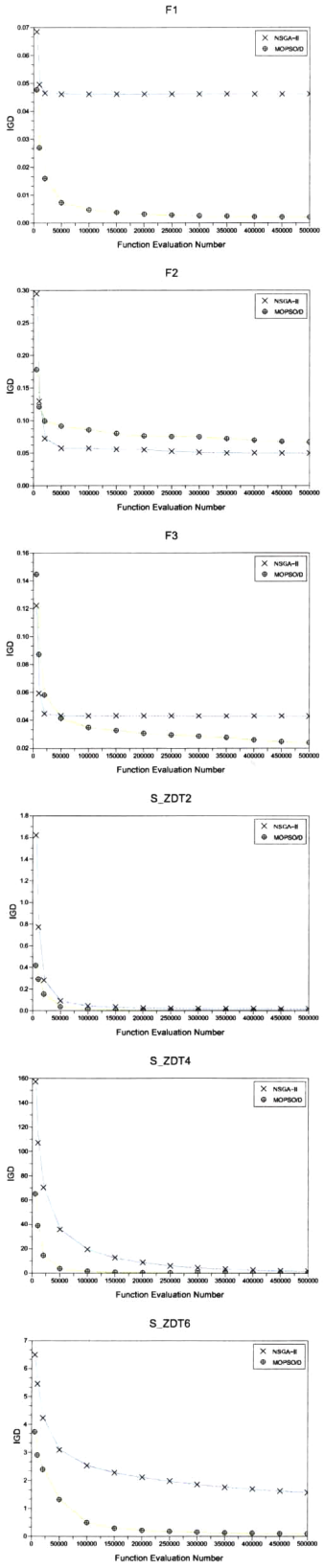

Figure 1. IGD-metric vs. function evaluation number

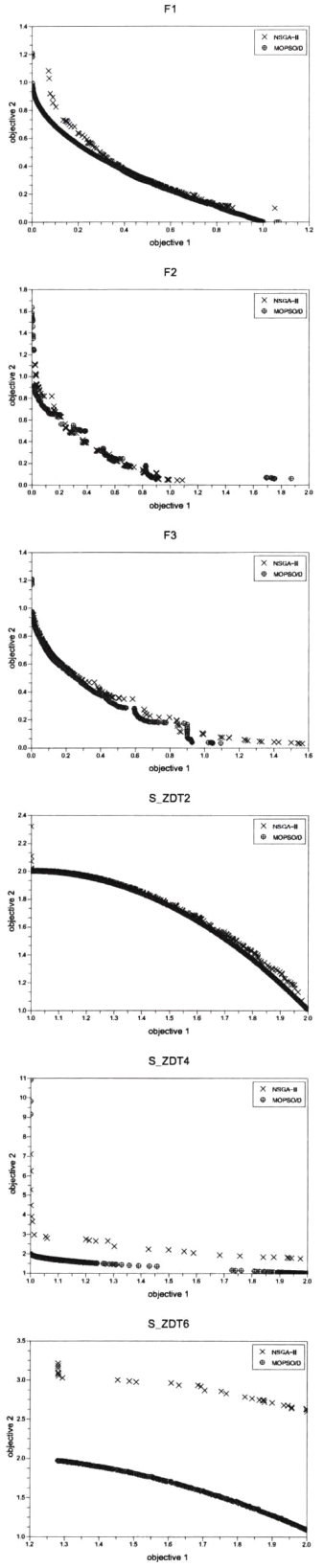

Figure 2. Best Pareto fronts found by all algorithms