

LETTER

An Improved PSO Algorithm for Interval Multi-Objective Optimization Systems

Yong ZHANG^{†a)}, Wanqiu ZHANG[†], Dunwei GONG[†], Yinan GUO[†], *Nonmembers*, and Leida LI[†], *Member*

SUMMARY Considering an uncertain multi-objective optimization system with interval coefficients, this letter proposes an interval multi-objective particle swarm optimization algorithm. In order to improve its performance, a crowding distance measure based on the distance and the overlap degree of intervals, and a method of updating the archive based on the acceptance coefficient of decision-maker, are employed. Finally, results show that our algorithm is capable of generating excellent approximation of the true Pareto front.

key words: interval, multi-objective, particle swarm, crowding distance

1. Introduction

In many real systems, it is often necessary to optimize multiple objectives that are generally conflicting with each other. This kind of problem is called a multi-objective optimization problem (MOP). MOPs still are a challenge to researchers due to the inherent conflicting nature among optimization objectives [1]. In those MOPs, part models usually include various uncertain parameters, such as fuzzy number, random number, interval number. Compared to random and fuzzy parameters, it is easier to get the range (i.e., interval) of an uncertain parameter [2]. Therefore, studying MOPs with interval parameters is of theoretical significance and practical application value.

Generally, an interval multi-objective optimization problem (IMOP) can be described as follows: to find a vector $\vec{x}^* = [x_1^*, x_2^*, \dots, x_n^*]$ satisfying

$$\bar{F}(\vec{x}) = (\bar{f}_1(\vec{x}, \vec{c}), \bar{f}_2(\vec{x}, \vec{c}), \dots, \bar{f}_M(\vec{x}, \vec{c})) \quad (1)$$

where $\vec{x} = (x_1, x_2, \dots, x_n) \in \Omega$ is called the decision variable, the set Ω is called the feasible region. $\vec{c} = (\bar{c}_1, \bar{c}_2, \dots, \bar{c}_L)$ is a vector composed of interval parameters, $\bar{c}_l = [c_l^-, c_l^+]$, $l = 1, 2, \dots, L$; $\bar{f}_j = [f_j^-, f_j^+]$ is the j -th interval objective function, $j = 1, 2, \dots, J$.

To solve interval single-objective optimization problems, researchers have proposed many approaches [3]–[5]. However, there are few articles for solving IMOP. Limbourg et al. introduced an imprecision-propagating multi-objective evolutionary algorithm (MOEA), by extending partial order relation into multiple-objective case [2]. Based on the strength Pareto evolutionary algorithm (SPEA) proposed

in [14], Teich proposed an improved SPEA algorithm using a probabilistic dominance relationship, called estimate strength Pareto evolutionary algorithm [6]. Sun et al. studied the theory of preference polyhedron for interval optimization problems, and presented an interactive multi-objective evolutionary algorithm [7]. Recently, Gong et al. considered the case that the user's preference is interval, and proposed a sorting scheme based on interval to guide the population evolving toward preferred regions [8].

Particle swarm optimization (PSO) is a heuristic search technique that is inspired by the behavior of bird flocks [9]. In previous work, we applied multi-objective PSO algorithm (MOPSO) in interval multi-objective optimization problems [10], [11]. Although simulation results show the effectiveness of the proposed algorithm, this algorithm still has the following disadvantages: (1) Using Sigma interval value to select the global best position easily leads to an excessive number of particles for exploiting the same regions; (2) like most of other PSO algorithms, it requires users to tune such control parameter as inertia weight and acceleration coefficients, in order to achieve desirable solutions of the optimized problem. Focused on the above disadvantages, in this paper we propose an improved multi-objective particle swarm optimizer.

2. The Proposed Algorithm

In this section, we propose an improved multi-objective PSO for interval fitness, by modifying related operators in PSO.

2.1 Selection of the Global Best Position

The global best position of a particle, $Gbest$, is the global best position found so far by neighbors of this particle. Since the neighbourhood of a particle is likely to include many non-dominated optimal solutions, selecting a suitable $Gbest$ for each particle becomes very difficult.

In this paper we use an external archive to save non-dominated optimal solutions found by the swarm so far. Based on the distribution density of elements in the external archive, each particle selects its $Gbest$ from the archive. Since the objective values of particles become interval vector, we propose a novel crowding distance measure to measure the distribution density of elements in the external archive.

First, we consider one dimensional objective space, for

Manuscript received March 2, 2016.

Manuscript revised May 6, 2016.

Manuscript publicized June 1, 2016.

[†]The authors are with School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China.

a) E-mail: yongzh401@126.com

DOI: 10.1587/transinf.2016EDL8052

example, the m -th objective space. In order to calculate the crowding degree of an element in this objective space, all elements in the external archive are ranked firstly based on the midpoints of their corresponding interval objective values, denoted those ranked elements as a_i , $i = 1, 2, \dots, |Ar|$, where $|Ar|$ is the number of elements in the external archive. If $i = 1$ or $i = |Ar|$, the crowding distance of a_i in this objective space is ∞ ; otherwise, the crowding distance of a_i in this objective space is calculated as follows:

$$CD_m(a_i) = D(\bar{f}_m(a_{i-1}), \bar{f}_m(a_i)) + D(\bar{f}_m(a_i), \bar{f}_m(a_{i+1})) \quad (2)$$

$$D(\bar{a}, \bar{b}) = e^{-c(\bar{a}, \bar{b})} \times \sqrt{(a^+ - b^+)^2 + (a^- - b^-)^2} \quad (3)$$

$$c(\bar{a}, \bar{b}) = \frac{2 \max\{0, \min\{a^+ - b^-, b^+ - a^-, w(\bar{a}), w(\bar{b})\}\}}{w(\bar{a}) + w(\bar{b})} \quad (4)$$

Where $w(\bar{a}) = (a^+ - a^-)/2$ reflects the uncertain degree of interval \bar{a} ; $c(\bar{a}, \bar{b})$ represents the overlap degree between interval \bar{a} and interval \bar{b} , $\sqrt{(a^+ - b^+)^2 + (a^- - b^-)^2}$ represents the distance between interval \bar{a} and interval \bar{b} . It is easy to observe that the smaller the distance between \bar{a} and \bar{b} is, the smaller the value of $D(\bar{a}, \bar{b})$ is; accordingly, the smaller the value of $CD_m(a_i)$ is. Similarly, the bigger the overlap degree between \bar{a} and \bar{b} is, the smaller the value of $D(\bar{a}, \bar{b})$ is; accordingly, the smaller the value of $CD_m(a_i)$ is.

Then, by running the above method repeatedly in every objective space, we should get the crowding distances of all the elements in the external archive in the whole objective space. Denoting those crowding distance values as $CD_m(a_j)$, $m = 1, \dots, M$, $j = 1, 2, \dots, |Ar|$, the crowding distance of element a_j in the whole objective spaces can be described as:

$$CD(a_j) = \frac{1}{M} \sum_{m=1}^M CD_m(a_j) \quad (5)$$

Assuming that S_t and Ar_t are the swarm and the external archive at the t -th generation, the selection method of $Gbest$ is described as follows:

Step1: Calculate the crowding distance of every element in Ar_t by the proposed crowding distance measure, and set $i = 1$;

Step2: Determine the global best position of the particle $\vec{x}_i(t)$.

Step2.1: Select two elements from Ar_t at random;

Step2.2: Compare the crowding distance values of the two elements, and select that with higher crowding distance as the global best position of $\vec{x}_i(t)$.

Step3: $i \leftarrow i + 1$, and go to Step 2, until $i > |S_t|$.

2.2 Update of the Personal Best Position

The personal best position, $Pbest$, is the best position achieved by a particle itself so far, which is regarded as the memory of the particle. Assuming that the $Pbest$ of the particle $\vec{x}_i(t)$ is $\vec{p}_i(t)$, new position of the particle at the $(t + 1)$ -th

generation is $\vec{x}_i(t + 1)$. If $\vec{x}_i(t + 1)$ dominates $\vec{p}_i(t)$, then $\vec{x}_i(t + 1)$ becomes new $\vec{p}_i(t + 1)$; if $\vec{p}_i(t)$ and $\vec{x}_i(t + 1)$ do not dominate each other, $\vec{p}_i(t + 1)$ will take $\vec{x}_i(t + 1)$ with the probability 0.5.

2.3 Update of the External Archive

It is worth noting that, in order to prevent losing optimal solutions acquired, all the non-dominated solutions should be saved into the external archive. However, it is unrealistic for many real problems to save all the non-dominated solutions because of high computational complexity. In order to overcome the above disadvantage, a method based on decision-maker tolerate coefficient is introduced to prune the archive in this paper. In this method, only optimal solutions that are dominated with less than μ ($0 \leq \mu \leq 1$) probability have an opportunity to be stored in the archive. Herein, the probability-based dominance relationship proposed in [2] is used to compare two solutions with interval objective values. When the number of saved solutions exceeds the fixed capacity N_a of the archive, only N_a elements with higher crowding distance are saved in the external archive.

Assuming that S_{t+1} is the swarm at the $(t + 1)$ -th generation, and Ar_t is the external archive at the t -th generation, the improved update method of the external archive can be described as follows:

Step1: Initialize the external archive at the $(t + 1)$ -th generation, Ar_{t+1} , as a null set;

Step2: Save both S_{t+1} and Ar_t to Ar_{t+1} simultaneously, namely $Ar_{t+1} \leftarrow S_{t+1} \cup Ar_t$;

Step3: Delete such elements that are dominated by other elements with more than μ probability from Ar_{t+1} ;

Step4: Judge whether the number of elements in Ar_{t+1} is smaller than the fixed capacity N_a or not. If yes, stop the update process, and output result;

Step5: Calculate the crowding distance of each element in Ar_{t+1} based on the proposed crowding distance measure, and delete $|Ar_{t+1}| - N_a$ elements from Ar_{t+1} , which have smaller crowding distance.

2.4 Steps of the Proposed Algorithm

Based on the above work, detailed steps of the proposed algorithm are described as follows:

Step1: Set relative parameters, including the swarm size N_s , the archive size N_a , and the terminal condition of our algorithm.

Step2: Initialize the positions of N_s particles in the search space, and set the personal best position of each particle as itself.

Step3: Calculate the interval objective values of all the particles;

Step4: Compare the dominance probability between elements both in the swarm and the archive, and update the external archive by the method proposed in Sect. 2.3.

Step5: Judge whether the algorithm meet the termination criterion. If yes, stop the algorithm;

Step6: Update the personal best position and the global best position by the method proposed in Sects. 2.1 and 2.2, respectively;

Step7: Update the particle positions. In order to reduce the influence of control coefficients, such as inertia weight and acceleration coefficients, on the performance of the proposed algorithm, a Gaussian sampling based on the global best position $\vec{g}_i = (g_{i,1}, g_{i,2}, \dots, g_{i,n})$ and the personal best position $\vec{p}_i = (p_{i,1}, p_{i,2}, \dots, p_{i,n})$, proposed in [12], is used to generate new particles, as follows:

$$x_{i,j} = \begin{cases} N(rp_{i,j} + (1-r)g_{i,j}, |p_{i,j} - g_{i,j}|), & \text{if } r' < 0.5 \\ g_{i,j}, & \text{otherwise} \end{cases} \quad (6)$$

Where r and r' are two random numbers within $[0, 1]$.

Step8: Run the uniform mutation [12] to improve the global search capability of swarm, and go to Step3.

3. Experiment and Analysis

To construct an imprecise test function, it seems feasible to choose some popular precise test functions. In this paper, an uncertainty factor ε is used to transform five popular precise functions, KUR, ZDT1, ZDT3, ZDT4 [2] and DTLZ1 [15], into interval multi-objective functions. Without loss of generality, this paper considers minimal optimization problems.

$$\varepsilon = \begin{pmatrix} |\sin(10\pi \sum_{i=1}^n x_i)| \\ |\sin(20\pi \sum_{i=1}^n x_i)| \end{pmatrix} \quad (7)$$

Two multi-objective interval evolutionary algorithms, i.e., PD-MOPSO [11] and IP-MOEA [2], are selected for performance comparison. In PD-MOPSO, the inertia weight = 0.6, two learning factor $c_1 = c_2 = 0.2$. In IP-MOEA, the crossover probability = 0.1, the mutation probability = 0.9, and the tournament selection is adopted. In the proposed algorithm, the tolerate coefficient of decision-maker $\mu = 0.2$. For all the three algorithms, the swarm size = 50, the archive size = 20, and the maximal evaluation times = 15000.

3.1 Results and Analysis

This subsection compares the average performance of the three algorithms based on the experimental results of 30 independent runs. Two metrics, the interval set coverage (ISC) and the interval spacing metric (ISP) [11], [13] are employed to evaluate the convergence and the distribution of solutions obtained by an algorithm, respectively. Here, the value of ISC(A, B) being 1 means that each solution of the algorithm B is dominated by at least one solution of the algorithm A. A value of zero for the ISP metric indicates all members of the Pareto front are equidistantly spaced.

Tables 1 to 5 show results of the three algorithms on the test functions in terms of ISC metric. We can see that for all the five test functions, the proposed algorithm has the best convergence among the three algorithms. Taking the

Table 1 ISC values of the three algorithms for KUR (Average/Variance).

ISC(A,B)	Proposed algorithm	IP-MOEA	PD-MOPSO
(Proposed algorithm,*)	–	0.6368/0.0557	0.5995/0.0470
(IP-MOEA,*)	0.0557/0.0097	–	0.2221/0.0366
(PD-MOPSO,*)	0.0594/0.0221	0.3234/0.0423	–

Table 2 ISC values of the three algorithms for ZDT1.

ISC(A,B)	Proposed algorithm	IP-MOEA	PD-MOPSO
(Proposed algorithm,*)	–	0.6028/0.1090	0.2920/0.1021
(IP-MOEA,*)	0.0979/0.0604	–	0.0767/0.0408
(PD-MOPSO,*)	0.2481/0.0833	0.5721/0.1106	–

Table 3 ISC values of the three algorithms for ZDT3.

ISC(A,B)	Proposed algorithm	IP-MOEA	PD-MOPSO
(Proposed algorithm,*)	–	0.6734/0.0906	0.2739/0.0218
(IP-MOEA,*)	0.0486/0.0253	–	0.0654/0.0203
(PD-MOPSO,*)	0.2327/0.0208	0.6099/0.1834	–

Table 4 ISC values of the three algorithms for ZDT4.

ISC(A,B)	Proposed algorithm	IP-MOEA	PD-MOPSO
(Proposed algorithm,*)	–	1/0	1/0
(IP-MOEA,*)	0/0	–	0.3512/0.4660
(PD-MOPSO,*)	0/0	0.4312/0.4698	–

Table 5 ISC values of the three algorithms for DTLZ1.

ISC(A,B)	Proposed algorithm	IP-MOEA	PD-MOPSO
(Proposed algorithm,*)	–	0.5425/0.0787	0.3458/0.0952
(IP-MOEA,*)	0.1050/0.0525	–	0.0685/0.0409
(PD-MOPSO,*)	0.3129/0.1004	0.5013/0.1421	–

Table 6 ISP values of the three algorithms for the three functions.

Function	Proposed algorithm	IP-MOEA	PD-MOPSO
KUR	0.7318/0.0677	0.9221/0.1791	1.1228/0.1553
ZDT1	0.0353/0.0119	0.0542/0.0099	0.0501/0.0142
ZDT3	0.1064/0.0262	0.0614/0.0189	0.1537/0.0508
ZDT4	0.0492/0.0184	0.0772/0.0120	0.0929/0.0149
DTLZ1	0.1131/0.0217	0.1490/0.0193	0.1581/0.0298

function KUR as an example, the proposed algorithm dominates IP-MOEA and PD-MOPSO with more than 0.59 probability, while both IP-MOEA and PD-MOPSO dominate the proposed algorithm with less than 0.06 probability, as their ISC values show.

In terms of the distribution of solutions, Table 6 shows results of the three algorithms in terms of ISP metric. We can see that for the functions KUR, ZDT1, ZDT4 and DTLZ1, the average performance of the proposed algorithm is better than IP-MOEA and PD-MOPSO in terms of ISP. For the function ZDT3, the proposed algorithm has the second best value in terms of ISP.

Furthermore, taking KUR, ZDT3 and ZDT4 as examples, Fig. 1 shows graphical results of these functions produced by the proposed algorithm and other two algorithms. It can be observed from Fig. 1 (a) that for KUR, IP-MOEA

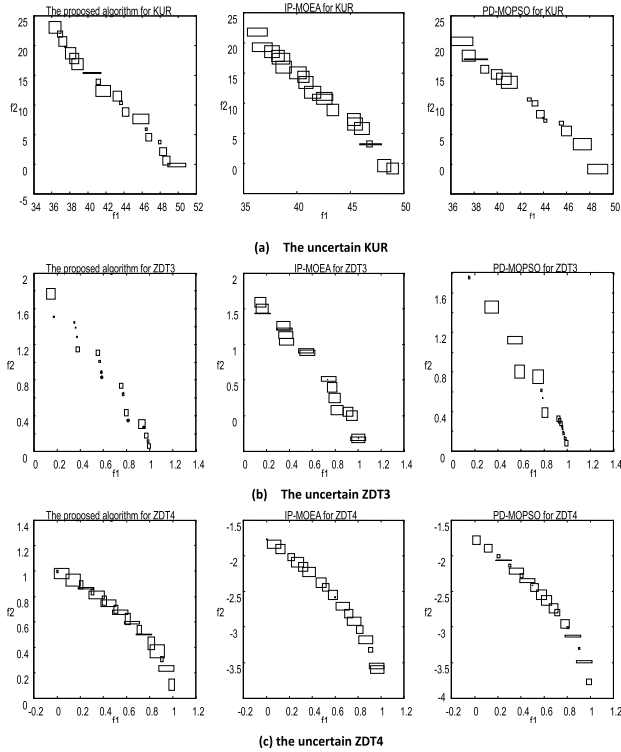


Fig. 1 Pareto fronts produced by the three algorithms on the three uncertain functions.

finds more overlap solutions than PD-MOPSO and the proposed algorithm, and the distribution of solutions of PD-MOPSO is not as good as that of the proposed algorithm. For ZDT3, as Fig. 1 (b) shows, most solutions of PD-MOPSO congregate on the lower left corner, compared with other two algorithms. The main reason is that PD-MOPSO does not consider the distribution of elements in the archive when the particles select their Gbests. Obviously, when overmany particles select the same Gbests, these particles will converge to the same regions.

To be fair in judgment of the performance, we also compare the CPU time of the three algorithms. It can be seen from experimental results that the proposed algorithm and PD-MOPSO consume similar CPU time. However, IP-MOEA consumes more CPU time than PD-MOPSO and the proposed algorithm for all the five functions, as a result of using non-dominated sorting technology.

4. Conclusion

In this paper, a new almost parameter-free MOPSO was proposed to handle interval multi-objective optimization problems. The proposed algorithm does not make use of those control parameters of PSO (i.e., inertia weight, acceleration coefficients, and velocity clamping) to update the particles. This makes it unnecessary for PSO to perform a fine tuning on these control parameters in order to pursue good performance. By comparing with two multi-objective interval evolutionary algorithms, IP-MOEA and PD-MOPSO,

experimental results show that the proposed algorithm can get a high competitive Pareto optimal set on optimizing the selected uncertain test functions.

Acknowledgements

This work was supported by the Research Program of Frontier Discipline of China University of Mining and Technology under Grant 2015XKQY19.

References

- [1] C.W. Ahn and Y. Kim, "Improving proximity and diversity in multiobjective evolutionary algorithms," *IEICE Trans. Inf. & Syst.*, vol.E93-D, no.10, pp.2879–2882, Oct. 2010.
- [2] P. Limbourg and D.E.S. Aponte, "An optimization algorithm for imprecise multi-objective problem functions," *Proc. IEEE Congress on Evolutionary Computation*, pp.459–466, 2005.
- [3] M. Inuiguchi and Y. Kume, "Goal programming problems with interval coefficients and target intervals," *European Journal of Operational Research*, vol.52, no.3, pp.345–360, 1991.
- [4] S. Wang, X. Fan, L. Han, and L. Ge "Improved interval optimization method based on differential evolution for microgrid economic dispatch," *Electric Power Components and Systems*, vol.43, no.16, pp.1882–1890, 2015.
- [5] H. Goto and H. Takahashi, "Efficient representation of the state equation in max-plus linear systems with interval constrained parameters," *IEICE Trans. Fundamentals*, vol.E95-A, no.2, pp.608–612, Feb. 2012.
- [6] J. Teich, "Pareto-front exploration with uncertain objectives," *The First International Conference of Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science*, vol.1993, pp.314–328, Springer, Berlin, Heidelberg, 2001.
- [7] J. Sun, D. Gong, and X. Sun, "Solving interval multi-objective optimization problems using evolutionary algorithms with preference polyhedron," *Proc. 13th Genetic and Evolutionary Computation Conference*, pp.729–736, 2011.
- [8] D. Gong, Y. Liu, and X. Ji, "Evolutionary algorithms with user's preferences for solving hybrid interval multi-objective optimization problems," *Applied Intelligence*, vol.43, no.3, pp.676–694, 2015.
- [9] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proc. IEEE International Conference on Neural Networks*, pp.1942–1948, 1995.
- [10] Y. Zhang, D.-W. Gong, G.-S. Hao, and Y.-Q. Jiang, "Particle swarm optimization for multi-objective systems with interval parameters," *Acta Automatica Sinica*, vol.34, no.8, pp.921–928, 2008.
- [11] Y. Zhang, *Theory of particle swarm optimization for interval multi-objective optimization problems and applications*, Doctoral Thesis, China University of Mining and Technology, Xuzhou, 2009.
- [12] Y. Zhang, D.-W. Gong, Z. Ding, "Barebones multi-objective particle swarm optimization for environmental/economic dispatch," *Information Sciences*, vol.192, pp.213–227, 2012.
- [13] C.A.C. Coello, G.T. Pulido, and M.S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol.8, no.3, pp.256–279, 2004.
- [14] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol.3, no.4, pp.257–271, 1999.
- [15] D. Gong, J. Sun, and X. Ji, "Evolutionary algorithms with preference polyhedron for interval multi-objective optimization problems," *Information Sciences*, vol.233, pp.141–161, 2013.