European Journal of Operational Research 202 (2010) 42–54

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Continuous Optimization

A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design

C.K. Goh [a], K.C. Tan [b,*], D.S. Liu [b], S.C. Chiam [b]

[a] Advanced Technology Centre, Rolls-Royce Singapore Pte Ltd, 50 Nanyang Avenue, Blk N2 B3C-05, Singapore 639798, Singapore
[b] Department of Electrical and Computer Engineering, National University of Singapore, 4, Engineering Drive 3, Singapore 117576, Singapore

ARTICLE INFO

Article history:
Received 4 September 2007
Accepted 4 May 2009
Available online 15 May 2009

Keywords:
Multi-objective optimization
Particle swarm optimization
Competitive–cooperative co-evolution

ABSTRACT

Multi-objective particle swarm optimization (MOPSO) is an optimization technique inspired by bird flocking, which has been steadily gaining attention from the research community because of its high convergence speed. On the other hand, in the face of increasing complexity and dimensionality of today's application coupled with its tendency of premature convergence due to the high convergence speeds, there is a need to improve the efficiency and effectiveness of MOPSO. In this paper a competitive and cooperative co-evolutionary approach is adapted for multi-objective particle swarm optimization algorithm design, which appears to have considerable potential for solving complex optimization problems by explicitly modeling the co-evolution of competing and cooperating species. The competitive and cooperative co-evolution model helps to produce the reasonable problem decompositions by exploiting any correlation, interdependency between components of the problem. The proposed competitive and cooperative co-evolutionary multi-objective particle swarm optimization algorithm (CCPSO) is validated through comparisons with existing state-of-the-art multi-objective algorithms using established benchmarks and metrics. Simulation results demonstrated that CCPSO shows competitive, if not better, performance as compared to the other algorithms.

1. Introduction

Many real world problems involves multiple conflicting objectives and various stochastic search techniques [24–26] for multi-objective (MO) optimization are gaining increasing attention from researchers. Particle swarm optimization (PSO) [23] is a class of stochastic optimization technique that is inspired by the behavior of bird flocks. PSO has been recognized to be suitable for MO problems and it has demonstrated higher convergence speeds as compared to canonical MO evolutionary algorithms (MOEA) [16,17,33]. Although multi-objective PSO (MOPSO) has been shown to be successful in various fields, researchers are facing the increasing challenge of problem complexity in today's applications. The computational cost increases with the size and complexity of the MO problem and the large number of function evaluations involved in the optimization process may be cost prohibitive. The necessity to improve MOPSO's efficacy and efficiency becomes more acute especially in high-dimensional problems.

Several studies [21,31] have shown that the introduction of ecological models and co-evolutionary architectures are effective methods to improve the efficacy of canonical genetic algorithm. Co-evolutionary techniques can overcome the exponential increase in difficulty by segregating the search space into smaller subspaces, and then conducting the overall optimization process over smaller regions. The co-evolutionary paradigm, inspired by the reciprocal evolutionary change driven by the cooperative [20] or competitive interaction [22] between different species, has been extended to MO evolutionary optimization successfully [1,9,10,14,18,27]. In a more recent work, a competitive–cooperative co-evolutionary paradigm which mimics the interplay of competition and cooperation among different species in nature is proposed by Goh and Tan [7]. The underlying idea behind the competitive–cooperative co-evolution framework is to allow the decomposition process of the optimization problem to adapt and emerge rather than being fixed at the start of the evolutionary process. The authors demonstrated the new co-evolutionary model can fulfill the MO optimization goals of attaining a good and diverse solution set with enhanced effectiveness and efficiency.

Despite the progress made in multi-objective evolutionary co-evolutionary optimization, there have not been any genuine attempts to extend it into MOPSO. The closest attempt is a cooperative co-evolutionary PSO for single-objective (SO) optimization proposed by van den Bergh and Engelbrecht [30]. Inspired by the competitive–cooperative co-evolution paradigm that exploits the complementary features of both competitive and cooperative co-evolutionary models, this paper presents a competitive–cooperative co-evolutionary MOPSO (CCPSO). The proposed CCPSO

* Corresponding author.
E-mail address: eletankc@nus.edu.sg (K.C. Tan).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2009.05.005

incorporates various features such as archiving, Pareto-based ranking, and a tournament-based competition strategy.

The reminder of the paper is organized as such: in Section 2, the related background on multi-objective optimization, PSO, and competitive–cooperative co-evolutionary model are reviewed. The proposed CCPSO is described in Section 3. In Section 4, the performance of the proposed algorithm is measured against other leading MOEAs on some established test functions in the field of multi-objective optimization. In addition, a sensitivity analysis of parameters is given. Conclusions are drawn in Section 5.

2. Background information

This section provides the necessary background to appreciate the work presented in this paper. A brief introduction of multi-objective optimization is given in Section 2.1 while the basic PSO algorithm is described in Section 2.2. An overview of the competitive–cooperative co-evolutionary model is presented in Section 2.3.

2.1. Multi-objective optimization

Many real-world applications involve complex optimization problem with various competing specifications and constraints [34–36]. Without any loss of generality, we consider a minimization problem and it tends to find a parameter set $\vec{x}$ for

$$\min \vec{f}(\vec{x}), \quad \vec{x} \in \Re^D, \tag{1}$$

where $\vec{x} = \{x_1, x_2, \ldots, x_D\}$ is a vector with D decision variables and $\vec{f} = \{f_1, f_2, \ldots, f_M\}$ are the M objectives to be minimized.

In contrast to single-objective optimization, the solution to MO optimization problem exists in the form of alternate tradeoffs known as Pareto optimal set. These solutions are also termed non-inferior, admissible or efficient solutions. The corresponding objective vectors of these solutions are termed nondominated and each objective component of any nondominated solution in the Pareto optimal set can only be improved by degrading at least one of its other objective components. A vector $\vec{f}_a$ is said to dominate another vector $\vec{f}_b$, denoted as

$$\begin{aligned}&\vec{f}_a \prec \vec{f}_b, \quad \text{iff } f_{a,i} \leqslant f_{b,i} \quad \forall i = \{1, 2, \ldots, M\} \quad \text{and}\\ &\exists j \in \{1, 2, \ldots, M\} \quad \text{where } f_{a,j} < f_{b,j}\end{aligned} \tag{2}$$

and the Pareto-optimal front is given as

$$\mathbf{PF}^* = \left\{\vec{f}^* \mid \nexists \vec{f} \prec \vec{f}^*, \vec{f} \in \Re^M\right\}. \tag{3}$$

2.2. Particle swarm optimization

The standard particle swarm optimizer maintains a swarm of particles that represent the potential solutions to the problem on hand. Each particle $\vec{p}_i = \{x_{i,1}, x_{i,1}, \ldots, x_{i,D}\}$ embeds the relevant information regarding the D decision variables $\{x_j, j = 1, 2, \ldots, D\}$, and is associated with a fitness that provides an indication of its performance in the objective space $\vec{f} \in \Re^M$. Its equivalence in $\vec{f}$ is denoted by $\vec{f}_i = \{f_{i,1}, f_{i,1}, \ldots, f_{i,M}\}$, where $\{f_k, k = 1, 2, \ldots, M\}$ are the objectives to be minimized.

In essence, the trajectory of each particle is updated according to its own flying experience as well as to that of the best particle in the swarm. The basic PSO algorithm can be described as

$$v_{i,d}^{k+1} = w \cdot v_{i,d}^k + c_1 \cdot r_1^k (p_{i,d}^k - x_{i,d}^k) + c_2 \cdot r_2^k \cdot \left(p_{g,d}^k - x_{i,d}^k\right), \tag{4}$$

$$x_{i,d}^{k+1} = x_{i,d}^k + v_{i,d}^{k+1}, \tag{5}$$

where $v_{i,d}^k$ is the dth dimension velocity of particle i in cycle k; $x_{i,d}^k$ is the dth dimension position of particle i in cycle k; $p_{i,d}^k$ is the dth dimension of personal best (pbest) of particle i in cycle k; $p_{g,d}^k$ is the d dimension of the global-best (gbest) in cycle k; w is the inertia weight; c_1 is the cognitive weight and c_2 is the social weight; r_1 and r_2 are two random values uniformly distributed in the range of $[0,1]$.

2.3. Competitive–cooperative co-evolution

The canonical co-evolutionary paradigm can be broadly classified into two main categories namely, competitive co-evolution and cooperative co-evolution. For the former, the various subpopulations will always fight to gain an advantage over the others. However, for the latter; subpopulations will exchange information within each other during the evolutionary process. Regardless of the different approaches, successful implementation of co-evolution requires the explicit consideration of several design issues [21] such as problem decomposition, parameter interactions and credit assignment, which are inherently problem dependent.

As mentioned in the introduction, the competitive–cooperative model [7] incorporates both elements of cooperation and competition which allows problem decomposition to emerge along the evolutionary process. This model involves two tightly-coupled co-evolutionary processes. As in the case of conventional cooperative co-evolutionary algorithms, individuals from the different species collaborate to solve the problem at hand during the cooperative process. Each species evolves in isolation and there is no restriction on the form of representation or the underlying optimization algorithm. On the other hand, the cooperative species will enter into competition with other species for the right to represent the various components of the problem.

The interaction frequency between the cooperative and competitive process can be determined by the designer according to the problem requirements. For the ensuing discussions, the MO problem is decomposed along the decision variables. Also, each decision variable may be assigned to a number of species populations and a species population may be optimizing more than one decision variable.

2.3.1. Credit assignment

Credit assignment for the competitive–cooperative process is performed at the species and individual level, respectively. In the cooperative process, the different objectives of the MO problem are evaluated by assembling each individual along with the representatives of the other species to form a valid solution. After which, appropriate fitness assignment such as Pareto ranking can be computed. In the competitive process, the fitness of particular species is computed by estimating how well it performs in a particular role relative to its competitors in the cooperation with other species to produce good solutions. For example, the species selected out of D competing species to represent a particular variable is given a higher probability of representing it in the later cycles, while the losing species of the competition is penalized and given a lower probability.

2.3.2. Problem decomposition and component interdependency

The difficulty with problem decomposition is that information regarding the number or role of components is usually not known *a priori*, and many problems exhibit complex interdependencies. The competitive process in the competitive–cooperative model will trigger a potential arms race among the various species to improve their contribution to overall fitness of the ecosystem. The benefits of this competition also include the discovery of interdependencies between the components as the species competition provides an environment in which interdependent components end up within the same species. The reasonable problem decompositions emerge due to co-evolutionary pressure rather than being specified by the user.

2.3.3. Diversity

The competitive–cooperation co-evolutionary model provides a means of exploiting the complementary diversity preservation mechanisms of both competitive and cooperative models. In the cooperative model, the evolution of isolated species tends to generate higher diversity across the different species. This property does not necessarily extend to within each species. Species diversity is also driven by the necessity to deal with the changing competition posed by the other species in the competitive model. Furthermore, the competitive process allows a more diversified search as the optimization of each component is no longer restricted to one species. The competing species provides another round of optimization for each component, which increases the extent of the search while maintaining low computational requirements.

3. Competitive–cooperative co-evolution for MOPSO

This section starts with the description of the cooperative co-evolutionary mechanism and the competitive co-evolutionary mechanism in Sections 3.1 and 3.2, respectively. The selection of gbest to guide the optimization process is presented in Section 3.3 while the archive updating process is described in Section 3.4. The algorithmic flowchart of CCPSO is presented in Section 3.5.

3.1. Cooperative co-evolutionary mechanism

The cooperative co-evolutionary mechanism employed in this paper is illustrated in Fig. 1. Similar to the cooperative co-evolutionary PSO proposed by van den Bergh and Engelbrecht [30], the problem is decomposed in the search space and the decision variables are evolved by different species, or called subswarms in the following discussions. The main difference is that the assignment of decision variables to different subswarms is adapted by the competitive mechanism, which will be described in the next section.

At the start of the optimization process, D subswarms are randomly initialized and the ith variable is assigned to the ith subswarm. In order to evaluate a particle in a subswarm, the particle under evaluation is combined with the representative of every other species to form a complete solution. In this paper, the best particle in the subswarm is defined as the representative of the subswarm. The archive is updated after each particle evaluation. The archive updating process will be described in a later section.

The subswarms are evaluated in an iterative manner. Before proceeding to the evaluation of the next subswarm, the representative of previous subswarm will be updated. This updating process is based on a partial order such that Pareto ranks will be considered first followed by niche count in order to break the tie of ranks. The Pareto rank is given by

$$\mathrm{rank}(i) = 1 + n_i, \tag{6}$$

where n_i is the number of archive members that dominates the particle i. For any two particles, the particle with lower rank is selected. In the case of a tie in rank, the particle with lower niche count is selected. The rationale of selecting a nondominated representative with the lowest niche count is to promote the diversity of the solutions.

3.2. Competitive co-evolutionary mechanism

Ideally, all particles from the competing subswarms should compete with all other particles from the other subswarms in order to determine the extent of its suitability. However, such an exhaustive approach requires extensive computational effort and it is practically infeasible. In this paper, each subswarm will be assigned a probability of representing a particular variable and only two subswarms, the current subswarm and competitor subswarm,

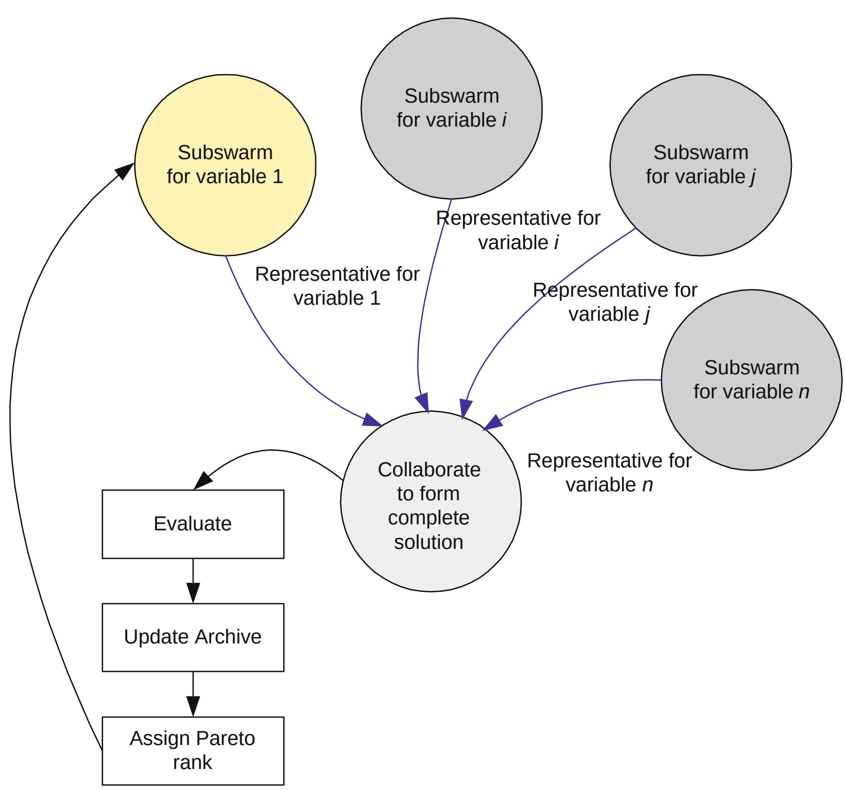

Fig. 1. Illustration of the cooperative co-evolutionary process.

will compete for the right to represent any variable at any one time. This selection probability is initialized as $1/D$ so that all subswarms have equal probability of being selected. This probability will be updated depending on outcome of the competition process.

The competitive co-evolutionary mechanism employed in this paper is illustrated in Fig. 2. As mentioned in the previous section, in the first cycle, the ith variable is assigned to the ith subswarm. In the subsequent cycles, a competing subswarm is selected using roulette wheel selection based on the selection probability to compete against the current species for the right to represent the, say, jth decision variable. During the competition process, the representatives of the current and competitor subswarms will combine with representatives of the other subswarms to form two complete solutions. The subswarm providing the better solution is the winner and will represent the jth decision variable in the next round of cooperation. The update of the probability of subswarm i representing variable j is given by

$$P_{i,j}(k) = P_{i,j}(k-1) \pm \frac{1}{D}\alpha, \quad (7)$$

where α is the learning rate. Therefore, the selection probability of a subswarm will increase as it becomes increasingly adapted to the decision variable. Vice versa, the probability of an unsuitable subswarm will be reduced. Note that it is possible for a subswarm to represent more than one decision variable.

3.3. Selection of global-best

While a single gbest exist in SO optimization, the gbest for MO optimization exist in the form of a set of nondominated solutions which inevitably leads to the issue of global-best selection. This paper adopts the approach presented in [16] for the selection of gbest. In MOPSO, the gbest plays a very important role in guiding the entire swarm towards the global Pareto front, i.e. the selection of appropriate gbest is crucial for the search of a diverse and uniformly distributed solution set.

Each particle in the subswarm will be assigned a nondominated solution from the archive as gbest. Assignment of gbest is performed through independent binary tournament selection of nondominated solutions from the archive for each particle in every cycle. Note that the nondominated solution is a complete solution with all D decision variable. Since each particle will only represent a component of the problem in CCPSO, only the associated decision variables will be represented in the selected gbest. Each particle is likely to be assigned different archived solution as the gbest because of the stochastic nature of the tournament selection. In order to promote diversity as well as to encourage exploration into the least populated areas of the search space, the selection criterion is based on niche count. In the event of a tie, preference will be given to solutions lying at the extreme ends of an arbitrarily selected objective.

3.4. Archiving process

The algorithm applies a fixed-size archive to store nondominated individuals along the evolution. The size of archive can be adjusted according to the desired number of solutions to be distributed on the tradeoffs in the objective space. A complete candidate solution formed by the subswarms will be added to the archive if it is not dominated by any archived solutions. Likewise, any archive members dominated by this candidate solution will be removed. When the predetermined archive size is reached, a recurrent truncation process [12] based on niche count is used to eliminate the most crowded archive member.

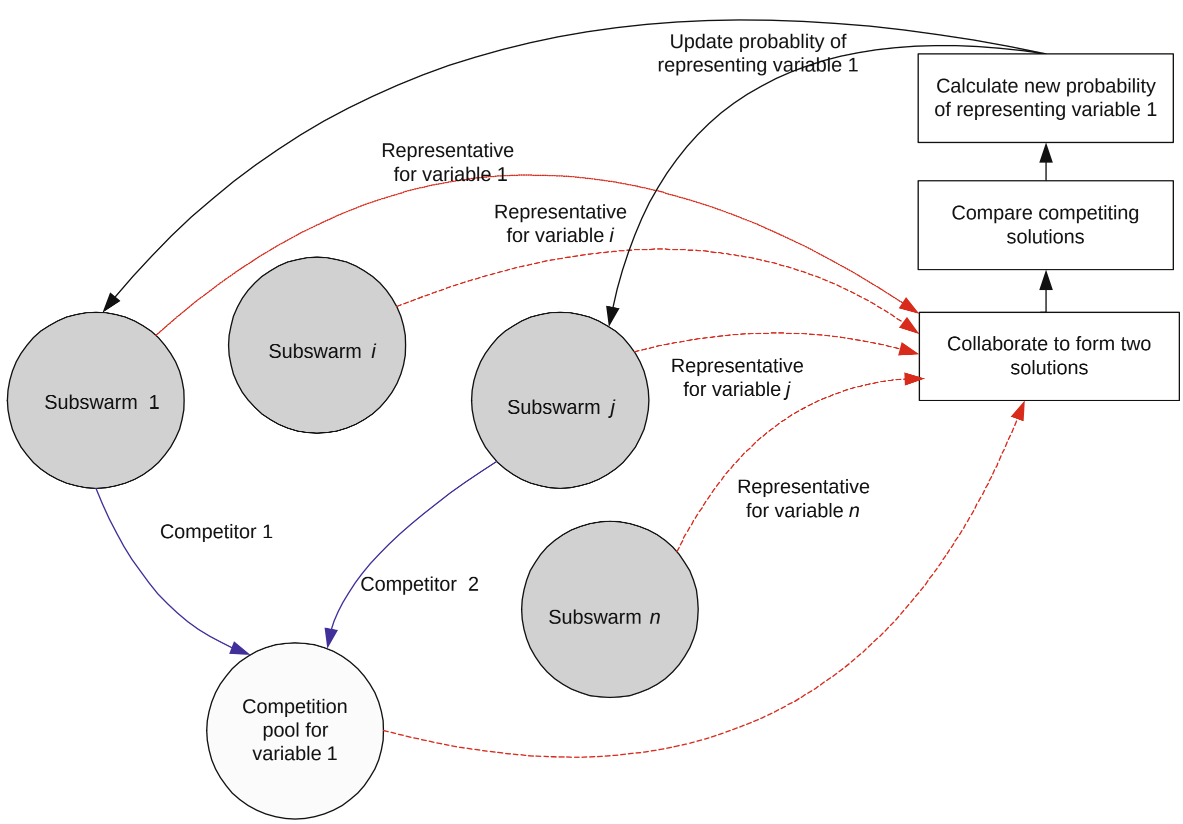

Fig. 2. Illustration of the competitive co-evolutionary process.

3.5. Implementation

The flowchart of the proposed algorithm is shown in Fig. 3. The optimization process starts with the initialization of the different subswarms. After that, the cooperation mechanism described in Section 3.1 is conducted to evaluate the particles in each subswarm. The archive is then updated based on the evaluated solutions. The new set of nondominated solutions in the archive is used as the reference for the calculation of the rank and niche count for each particle. The pbest will be updated if the current solution dominates the previous pbest. In the case in which neither solution dominates each other, the current solution will be given a 50% chance to be the new personal best. The position of the particles are updated and the turbulence operator is applied. As shown in Fig. 3, the competitive process will be activated and the probability of the subswarms in representing various variables are updated.

4. Experimental setup and results

This section starts with a description of various performance metrics and MO test functions in Sections 4.1 and 4.2, respectively. Sensitivity analysis of various parameter settings in the CCPSO is conducted in Section 4.3. Subsequently, a comparative study between CCPSO and various MOEAs that are representatives of the state-of-the-arts will be conducted in Section 4.4.

4.1. MO test functions

Six benchmark problems, FON, KUR, ZDT4, ZDT6, DTLZ2 and DTLZ3, are used to examine and compare the performance of CCPSO and other state-of-the-art MOEAs. These test functions have different problem characteristics [3], such as multi-modality, convexity, discontinuity and non-uniformity, which may challenge the MOEA's ability to converge and maintain population diversity. It has been shown that the performances of MOEAs often do not scale well with respect to the number of objectives [8,11]. Therefore, DTLZ2 and DTLZ3 are formulated here as a five-objective optimization problem. The definition of these test functions is summarized in Table 1.

4.2. Performance metrics

In general, the performance metrics used in the evaluation of MOEA performances must provide an indication of, (1) the distance between the nondominated set produced by the algorithm and the Pareto optimal set of the problem, (2) the distribution of the solutions along the Pareto-optimal front, and (3) the extent covered by the discovered solutions. In this paper, three performance metrics are adopted.

4.2.1. Proximity indicator

The metric of generational distance (GD) gives a good indication of the gap between the $\mathbf{PF}^*$ and the evolved PF. Mathematically, the metric is a function of individual distance given as,

$$\mathrm{GD} = \frac{1}{n_{PF}} \cdot \left(n_{PF} \sum_{i=1}^{n_{PF}} d_i^2 \right)^{\frac{1}{2}} \tag{8}$$

where $n_{PF} = |PF|$, d_i is the Euclidean distance (in objective space) between the ith member of PF and the nearest member of $\mathbf{PF}^*$. A low value of GD is desirable, which reflects a small deviation between the evolved and the true Pareto front.

4.2.2. Diversity indicator

A modified maximum spread (MS′)[6] is applied to measure how well the $\mathbf{PF}^*$ is covered by the evolved PF. Specifically, the modified metric takes into account the proximity to $\mathbf{PF}^*$, e.g., a

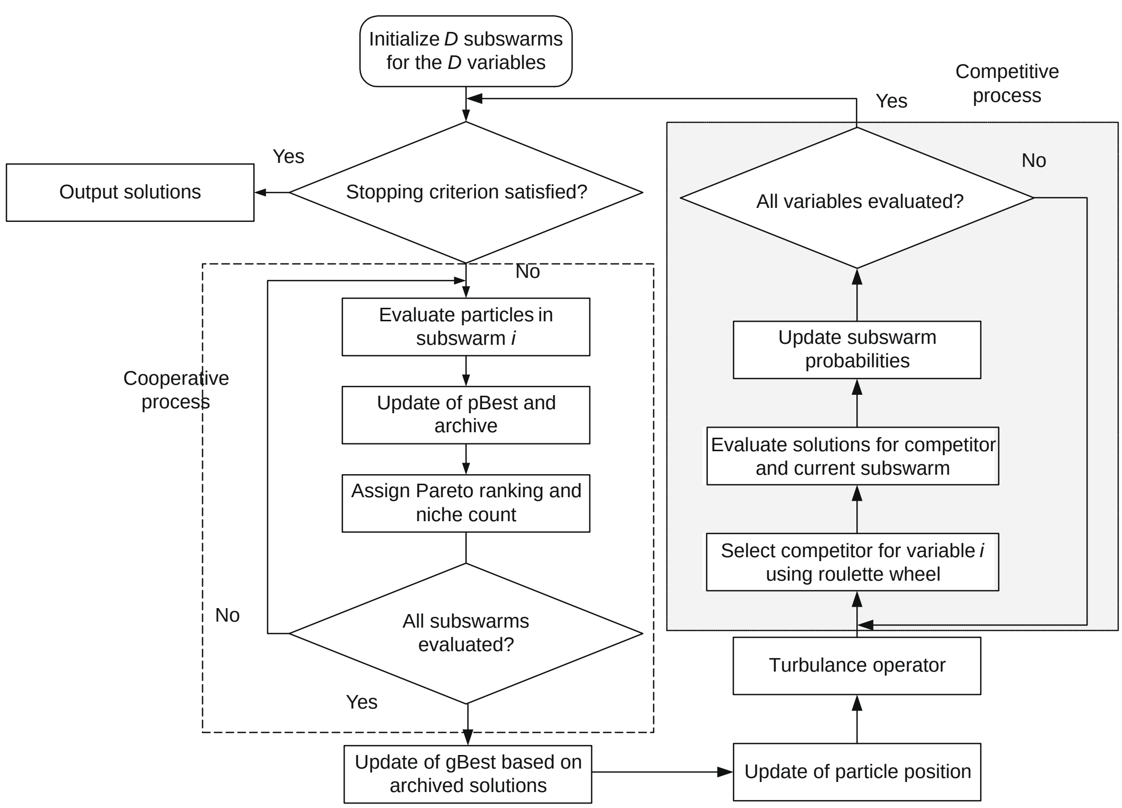

Fig. 3. Algorithmic flow of CCPSO.

Table 1
Definition of static test functions.

	Test	Definition
1	FON	$f_1(x_1,\dots,x_8) = 1 - \exp\left[-\sum_{i=1}^{8}\left(x_i - \frac{1}{\sqrt{8}}\right)^2\right]$, $f_2(x_1,\dots,x_8) = 1 + \exp\left[-\sum_{i=1}^{8}\left(x_i - \frac{1}{\sqrt{8}}\right)^2\right]$, where $-2 \leqslant x_i < 2,\ \forall i = 1,2,\dots,8$
2	KUR	$f_1(x_2,x_3) = \sum_{i=1}^{2}\left[-10\exp\left(-0.2\sqrt{x_i^2 + x_{i+1}^2}\right)\right]$, $f_1(x_2,x_3) = \sum_{i=1}^{3}\left[\lvert x_i\rvert^{0.8} + 5\cdot\sin(x_i^3)\right]$, $x_i \in [-5,5]$
3	ZDT4	$f_1(x_1) = x_1$, $f_2(x_2,\dots,x_m) = g\cdot h$, $g(x_2,\dots,x_m) = 1 + 10(m-1) + \sum_{i=2}^{m}(x_i^2 - 10\cos(4\pi x_i))$, $h(f_1,g) = 1 - \sqrt{\frac{f_1}{g}}$, where $m = 10,\ x_1 \in [0,1],\ -1 \leqslant x_i < 1,\ \forall i = 2,\dots,10$
4	ZDT6	$f_1(x_1) = 1 - \exp(-4x_1)\cdot\sin^6(6\pi x_1)$, $f_2(x_2,\dots,x_m) = g\cdot h$, $g(x_2,\dots,x_m) = 1 + 9\cdot\left(\frac{\sum_{i=2}^{m}x_i}{m-1}\right)^{0.25}$, $h(f_1,g) = 1 - \left(\frac{f_1}{g}\right)^2$, where $m = 10,\ x_i \in [0,1]$
5	DTLZ2	$f_1(\vec{x}) = (1 + g(\vec{x}_M))\cdot\cos(0.5\pi x_1)\cdots\cos(0.5\pi x_{M-1})$, $f_2(\vec{x}) = (1 + g(\vec{x}_M))\cdot\cos(0.5\pi x_1)\cdots\sin(0.5\pi x_{M-1})$, $\vdots$ $f_M(\vec{x}) = (1 + g(\vec{x}_M))\cdot\sin(0.5\pi x_1)$, $g(\vec{x}_M) = \sum_{x_i\in\vec{x}_M}(x_i - 0.5)^2$, where $M = 5,\ \vec{x}_M = \{x_M,\cdots,x_{M+9}\},\ x_i \in [0,1]$
6	DTLZ3	$f_1(\vec{x}) = (1 + g(\vec{x}_M))\cdot\cos(0.5\pi x_1)\cdots\cos(0.5\pi x_{M-1})$, $f_2(\vec{x}) = (1 + g(\vec{x}_M))\cdot\cos(0.5\pi x_1)\cdots\sin(0.5\pi x_{M-1})$, $\vdots$ $f_M(\vec{x}) = (1 + g(\vec{x}_M))\cdot\sin(0.5\pi x_1)$, $g(\vec{x}_M) = 100\{\lvert\vec{x}_M + \sum_{x_i\in\vec{x}_M}(x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\}$, where $M = 5,\ \vec{x}_M = \{x_M,\dots,x_{M+9}\},\ x_i \in [0,1]$

higher value of MS′ reflects that a larger area of the $\mathbf{PF}^*$ is covered by the PF. The metric is given as,

$$MS' = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left[\frac{\min\left[\overline{PF}_i, \overline{PF}_i^*\right] - \max\left[\underline{PF}_i, \underline{PF}_i^*\right]}{\overline{PF}_i^* - \underline{PF}_i^*}\right]^2} \tag{9}$$

where $\overline{PF}_i$ and $\underline{PF}_i$ is the maximum and minimum of the ith objective in PF, respectively; $\overline{PF}_i^*$ and $\underline{PF}_i^*$ is the maximum and minimum of the ith objective in $\mathbf{PF}^*$, respectively.

4.2.3. Distribution indicator

The metric of spacing [19] shows how evenly the nondominated solutions are distributed along the discovered Pareto front. It is given as,

$$S = \frac{1}{\bar{d}'}\cdot\left(\frac{1}{n_{PF}}\cdot\sum_{i=1}^{n_{PF}}(d_i' - \bar{d}')^2\right)^{\frac{1}{2}}, \tag{10}$$

$$\bar{d}' = \frac{1}{n_{PF}}\sum_{i=1}^{n_{PF}} d_i',$$

where $n_{PF} = |PF|$, d_i' is the Euclidean distance (in objective space) between the ith member and its nearest member in PF.

4.3. Sensitivity analysis

In this section, the impact of various parameter settings of CCPSO is examined. A number of simulations are performed with different settings of $w = \{0.2, 0.4, 0.6\}$, $S_{subpop} = \{5, 10, 20, 50\}$, and $S_{arc} = \{100, 150, 200, 250\}$. An additional setup involving an adaptive inertia weight that reduces from 0.9 to 0.4 is also applied and compared against the static settings. For each experiment only one parameter was changed while maintaining all other parameters constant, and the box-plots show results of parameters being varied in increasing order. The parameters are changed one by one in order to tell algorithm designers the individual effects of each parameter. Such information will be a good starting point for designers to fine-tune their program to real world problems.

4.3.1. Inertial weight

Fig. 4 shows the performance of CCPSO over different w settings. The inertia weight, which helps maintain the balance between exploration and exploitation, is varied among 0.2, 0.4, 0.6 and an adaptive weight that reduces from 0.9 to 0.4. Note that a smaller w represents a higher tendency to adopt changes brought about by changes due to pbest and gbest. It can be observed that the inertia has a greater impact on MS as compared to S and GD. Interestingly, a higher w and the adaptive inertia scheme provide a better spread of solutions for both problems. On the other hand, a smaller w tend to provide better convergence, probably due to a higher tendency to adopt new changes. It should also be noted that CCPSO is capable of performing consistently and effectively within a large range of w.

4.3.2. Subswarm size

Fig. 5 shows the performance of CCPSO over different settings $S_{subpop} = \{5, 10, 20, 50\}$. The population size of the subswarm was varied while maintaining the total number of evaluations. From the box-plots, it is apparent that smaller species sizes give rise to better convergence to the true Pareto front. Nonetheless, we note that higher values of MS denoting better diversity are achieved at higher S_{subpop} settings in the case of FON. This is because, by maintaining a fixed number of evaluations, there is an inherent tradeoff between the diversity provided by a larger population size and the number of generations allowed for exploration.

4.3.3. Archive size

Fig. 6 shows the performance of CCPSO over different settings of $S_{arc} = \{100, 150, 200, 250\}$. Apart from the problem of ZDT4 in which the archive size seems to have little effect, CCPSO tends to achieve better results in all aspects with increasing archive sizes. This is because the restriction on the number of archive solutions leads to two phenomena [5] which have a detrimental effect on the search process. The first is the shrinking PF phenomenon which results from the removal of extremal solutions and the subsequent failure to rediscover them. In the second phenomenon, nondominated solutions in the archive are replaced by least crowded individuals. In the subsequent generations, new individuals that would have been dominated by the removed solutions are updated into the archive only to be replaced solutions dominating them. Nonetheless, we note that an archive size of 100 is sufficient to produce good results.

4.4. Simulation results

In this section, the proposed CCPSO is compared with evolutionary multi-objective optimization methods (NSGAII [4], SPEA2 [32], IMOEA [29] and PAES [13]) that are representative of the state-of-the-arts. Two multi-objective particle swarm optimization algorithms (SMOPSO [15] and CMOPSO [2]) have also been included to reflect an accurate measure of the algorithm against other established PSO algorithms implemented rather than simply pitting the CCPSO against other MOEAs, where any inherent advantages or disadvantages that a MOPSO may have over MOEAs may skew

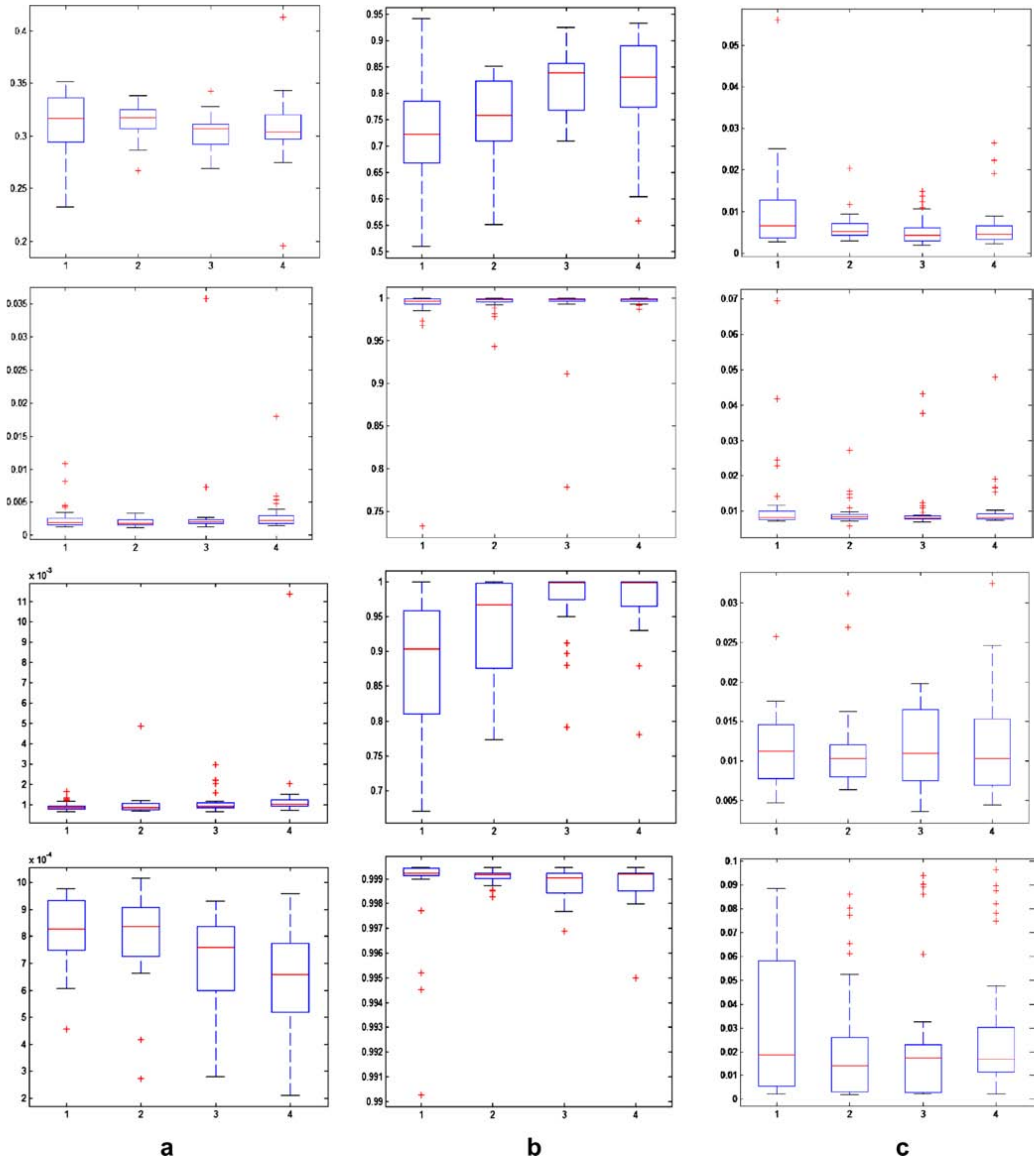

Fig. 4. Algorithm performance in (a) GD, (b) MS, and (c) S for FON (1st row), KUR (2nd row), ZDT4 (3rd row) and ZDT6 (4th row), with various w settings.

the results. The parameter settings and indices of the different algorithms are shown in Tables 2 and 3, respectively.

4.4.1. FON

FON has a nonconvex and nonlinear tradeoff curve that challenges the algorithms ability to find and maintain the entiresi tradeoff curve uniformly. A stopping criterion of 8000 evaluations is used for this problem. The distribution of the different performance metrics is represented by box-plots in Fig. 7.

From the box-plots, it is clear that the PSO paradigm has clear advantage over the various MOEAs used in the study. While SMOPSO, CCPSO and CMOPSO demonstrated competitive performance, the four MOEAs exhibited a varying degree of success. In particular, PAES and IMOEA performed inconsistently in GD and MS. It can be observed that the PSO framework attains the best performance in the metric of MS. Nonetheless, CCPSO and CMOPSO are competitive in the aspects of convergence and distribution as evident in Fig. 7a–b.

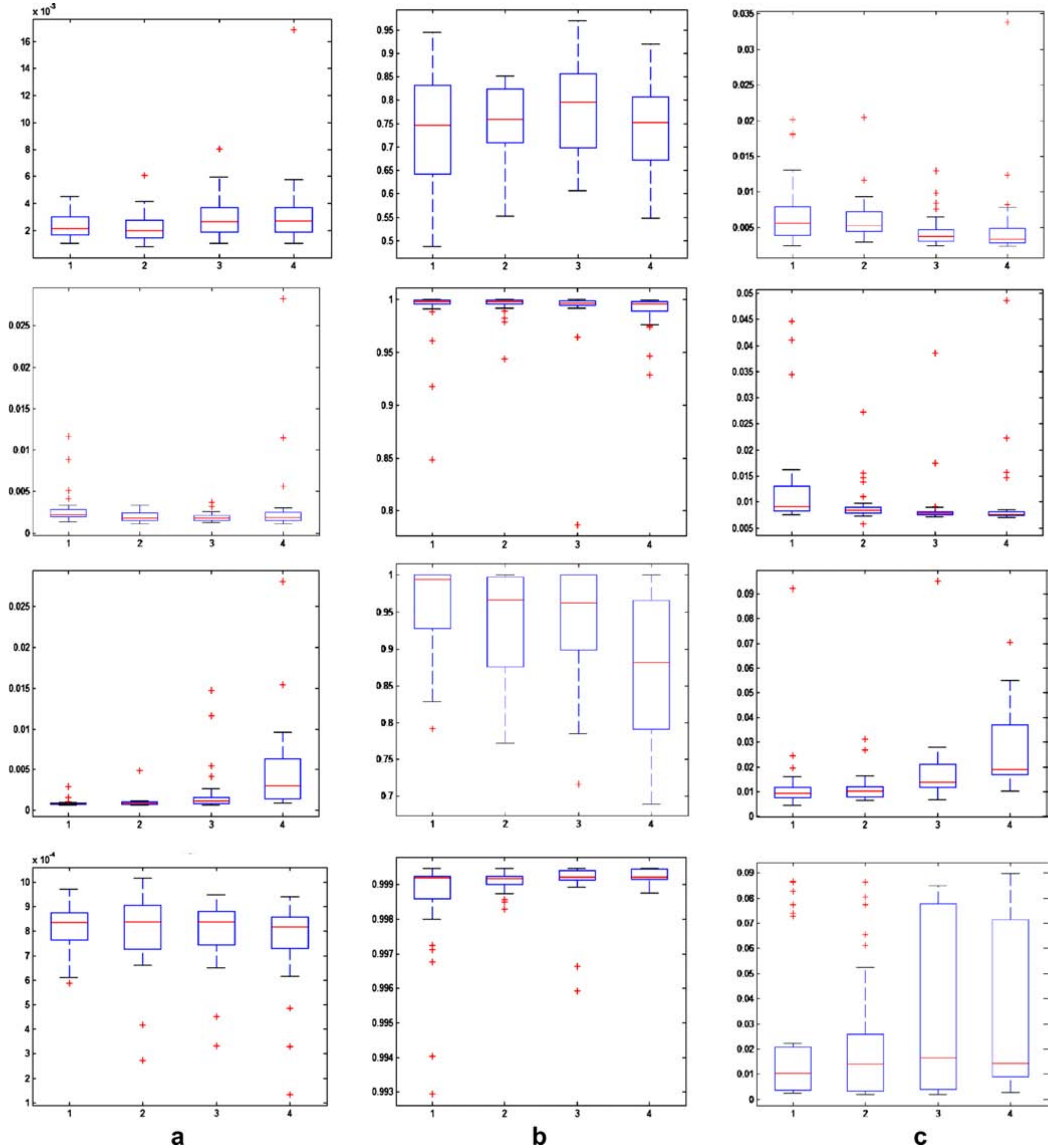

Fig. 5. Algorithm performance in (a) GD, (b) MS, and (c) S for FON (1st row), KUR (2nd row), ZDT4 (3rd row) and ZDT6 (4th row), with various S_{subpop} settings.

4.4.2. KUR

KUR is characterized by a tradeoff that is nonconvex and disconnected, i.e., it contains three distinct disconnected regions on the final tradeoff. The decision variables corresponding to the global tradeoff for KUR are difficult to be discovered, since they are disconnected in the decision variable space as well. A stopping criterion of 4500 evaluations is used for this problem. The distribution of the different performance metrics is represented by box-plots in Fig. 8.

It should be noted that results of PAES in terms of GD is not plotted in the box-plot since it is unable to converge within the allocated number of evaluations. On the other hand, the other algorithms faced no problems finding solutions near the true Pareto front. It is evident from Fig. 8 that CCPSO performs better than the other algorithms in the aspects of GD and S while remaining strongly competitive in terms of MS. Unlike the case of FON where the PSO-based algorithms are significantly faster than the MOEA counterparts, SPEA2 and CCPSO demonstrated competitive convergence rates.

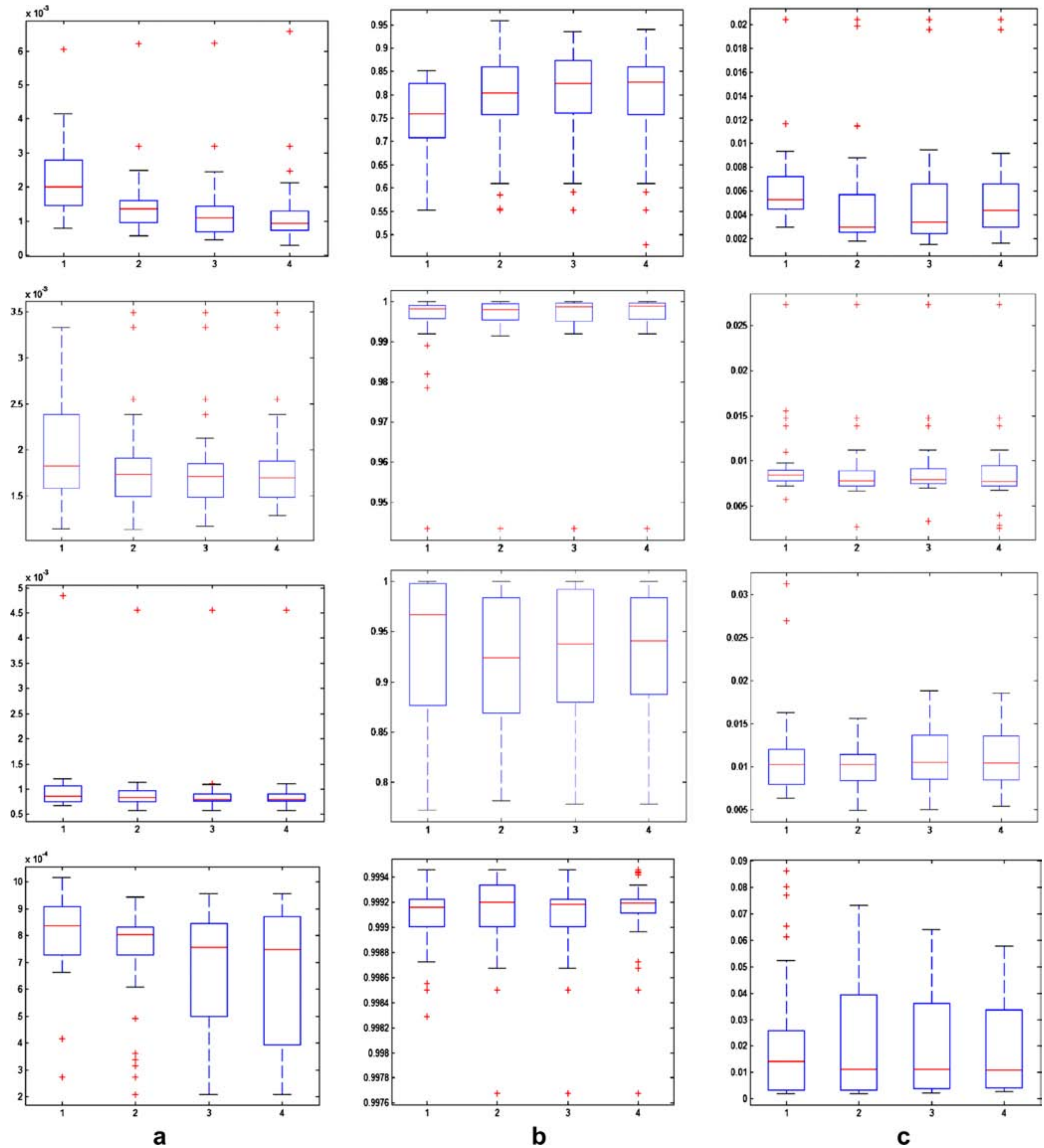

Fig. 6. Algorithm performance in (a) GD, (b) MS, and (c) S for FON (1st row), KUR (2nd row), ZDT4 (3rd row) and ZDT6 (4th row), with various S_{arc} settings.

4.4.3. ZDT4

ZDT4 is characterized by a severe multi-modal landscape and a stopping criterion of 10,000 evaluations is used for this problem. The evolved Pareto fronts from all 30 simulation runs for the algorithms are plotted in Fig. 9. The distribution of the different performance metrics is represented by box-plots in Fig. 10.

ZDT4 proved to be a very difficult problem to solve, mainly due to the high number (219) of local fronts. As observed from Figs. 9d and 10, CCPSO is the only algorithm capable of attaining the true Pareto front. In addition, it is clear that the poor performances of CMOPSO and SMOPSO in the aspects of GD and MS is the consequence of high convergence speed, as evident in Fig. 9 resulting in the algorithms getting trapped in the local optimal fronts.

4.4.4. ZDT6

ZDT6 has a biased search space and non-uniformly distributed solutions along the global tradeoff, which makes it difficult for algorithms to evolve a well-distributed Pareto front. A stopping cri-

Table 2
Parameter setting for different algorithms.

Parameter	Settings
Populations	Population size 100 in NSGAII, SPEA2, SMOPSO, IMOEA, CMOPSO
	Subpopulation size 10 in CCPSO
	Population size 1 in PAES
	Archive (or secondary population) size 100
Chromosome	Binary coding; 30 bits per decision variable in IMOEA, PAES, NSGAII and SPEA2
	Real number representation in CCPSO, CMOPSO, and SMOPSO
Selection	Binary tournament selection
Crossover operator	Uniform crossover in IMOEA, NSGAII and SPEA2
Crossover rate	0.8 in IMOEA, NSGAII and SPEA2
Mutation operator	Bit-flip mutation in IMOEA, PAES, NSGAII and SPEA2
	Turbulence operator in CCPSO, CMOPSO, and SMOPSO
Mutation rate	$\frac{1}{L}$ for ZDT4, ZDT6, DTLZ2 and DTLZ3 where L is the chromosome length
	$\frac{1}{B}$ for FON and KUR where B is the bit size per decision variable
Turbulence strength	0.2
Niche radius	Dynamic sharing [28]

Table 3
Indices of different algorithms.

NSGAII	SPEA2	SMOPSO	CCPSO	IMOEA	CMOPSO	PAES
1	2	3	4	5	6	7

terion of 10,000 evaluations is used for this problem. The distribution of the different performance metrics is represented by box-plots in Fig. 11.

As in the case for FON, the PSO algorithms in general outperformed the rest of the algorithms, managing to cover almost the entire actual Pareto front over the thirty runs. From Fig. 11a and b, it is apparent that NSGAII and SPEA2 are unable to reach the true Pareto front consistently and failed to cover the entire set of optimal solutions throughout the thirty runs. It can also be noted that while PAES outperforms NSGAII and SPEA2 in the aspects of GD, but it performs poorly in attaining a well-distributed Pareto front as shown by the poor MS. It can also be observed that the proposed CCPSO has a better capability to attain a near-optimal, well-distributed, and well-spread Pareto front with better result on GD, MS and S.

4.4.5. DTLZ2

DTLZ2 is used to investigate an algorithms ability to produce adequate pressure for searching toward the large Pareto front in the high-dimensional objective domain. There are five objectives to optimize for the problem of DTLZ2, and a stopping criterion of 20,000 evaluations is used. The distribution of the different performance metrics is represented by box-plots in Fig. 12.

For DTLZ2, the PSO algorithms in general outperformed the rest of the algorithms, managing to get the best GD over the thirty runs. It can be noted from Fig. 12 that while NSGAII and SPEA2 performs well in the aspects of MS and S, it performs poorly in attaining a good GD. From Fig. 12a, it is apparent that the CCPSO scales well for the large number of objectives of DTLZ2, while the MOEAs used

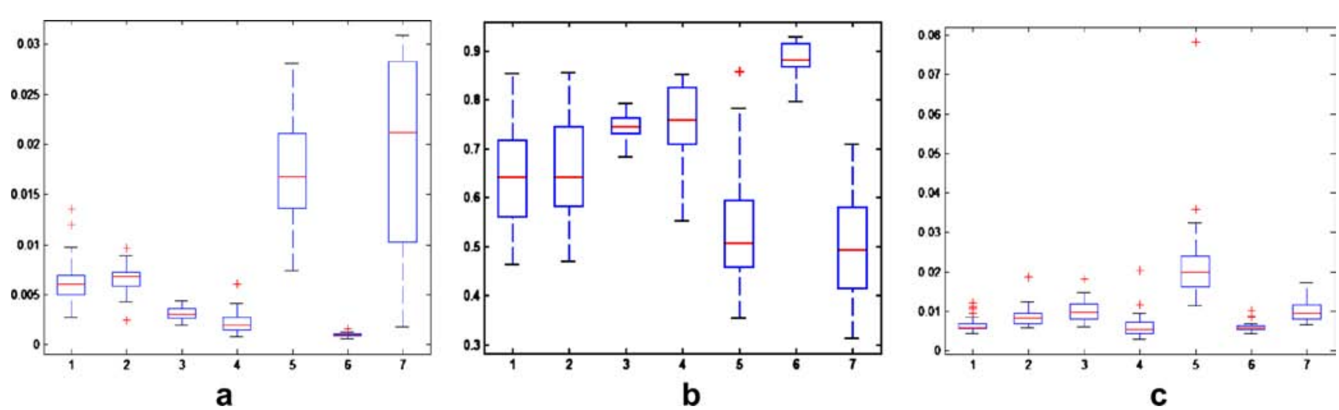

Fig. 7. Algorithm performance in (a) GD, (b) MS, and (c) S for FON.

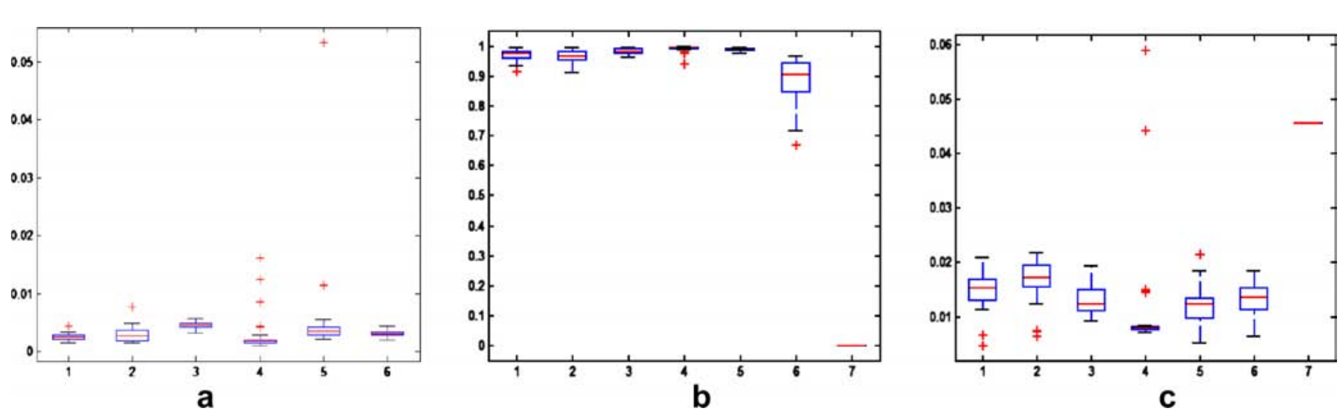

Fig. 8. Algorithm performance in (a) GD, (b) MS, and (c) S for KUR.

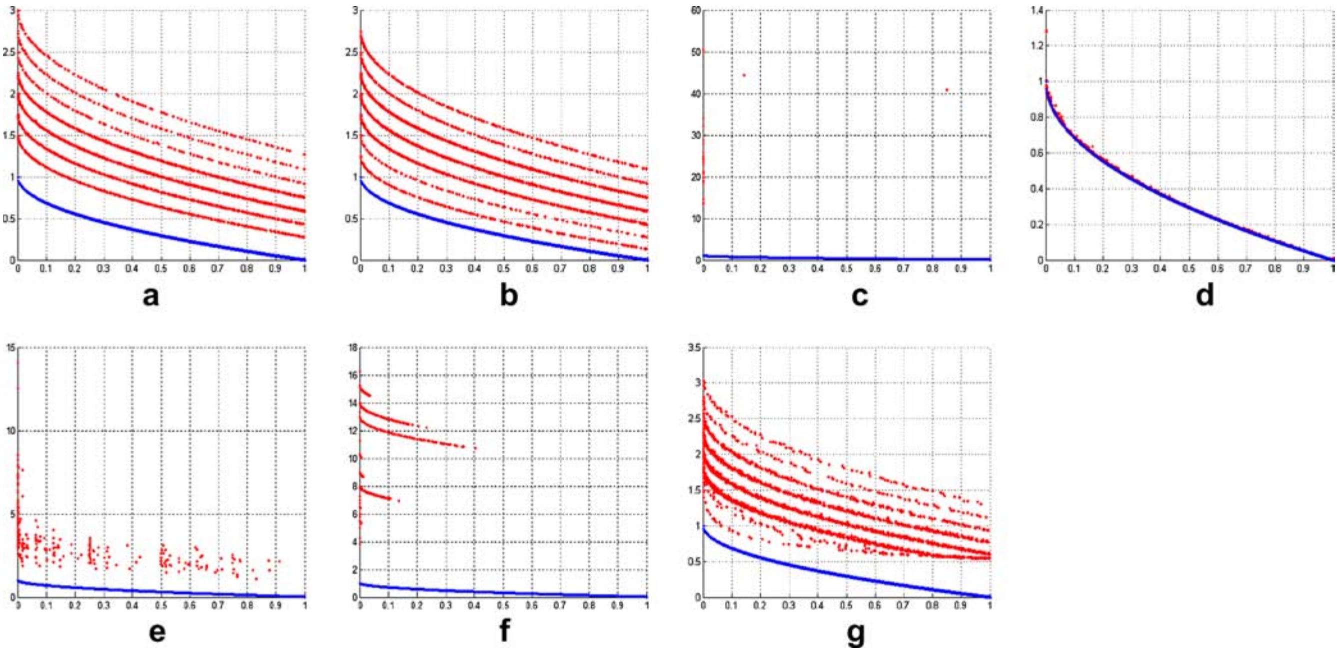

Fig. 9. Pareto fronts generated across 30 runs by (a) NSGAII, (b) SPEA2, (c) SMOPSO, (d) CCPSO, (e) IMOEA, (f) CMOPSO, and (g) PAES for ZDT4.

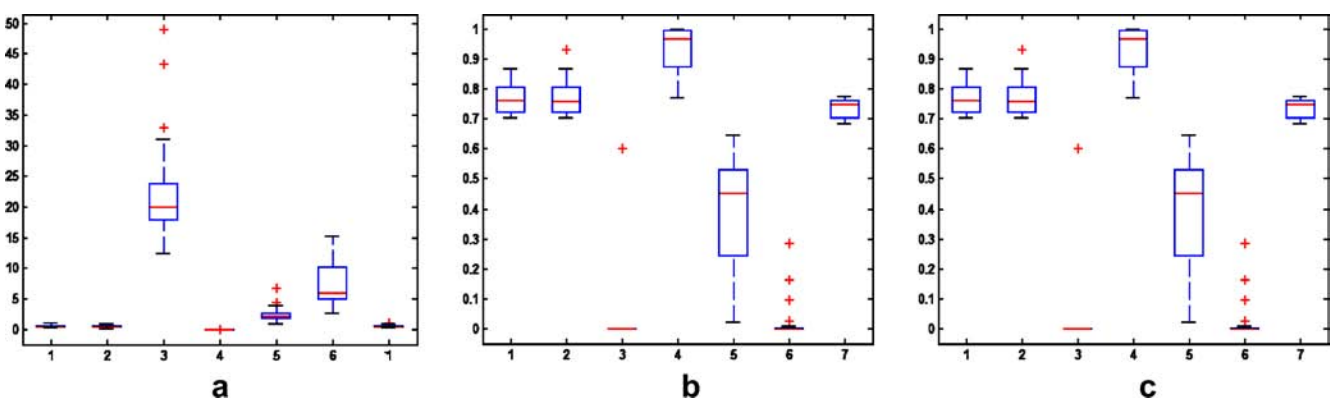

Fig. 10. Algorithm performance in (a) GD, (b) MS, and (c) S for ZDT4.

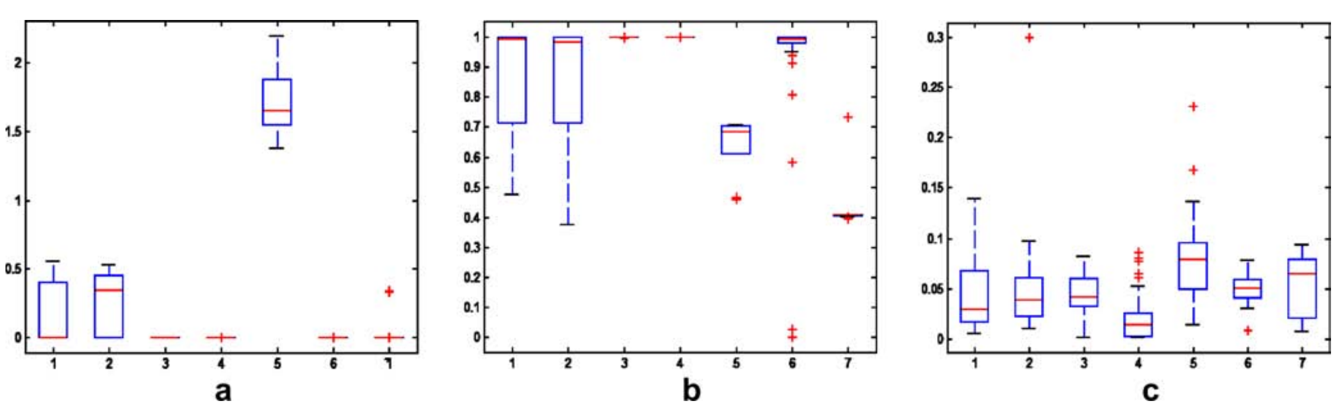

Fig. 11. Algorithm performance in (a) GD, (b) MS, and (c) S for ZDT6.

in this experiment suffer in convergence. CCPSO achieves good convergence rate while maintaining competitive performance on MS and S.

4.4.6. DTLZ3

DTLZ3 has high-dimensional objective space and many local Pareto fronts. All local Pareto-optimal fronts are parallel to the glo-

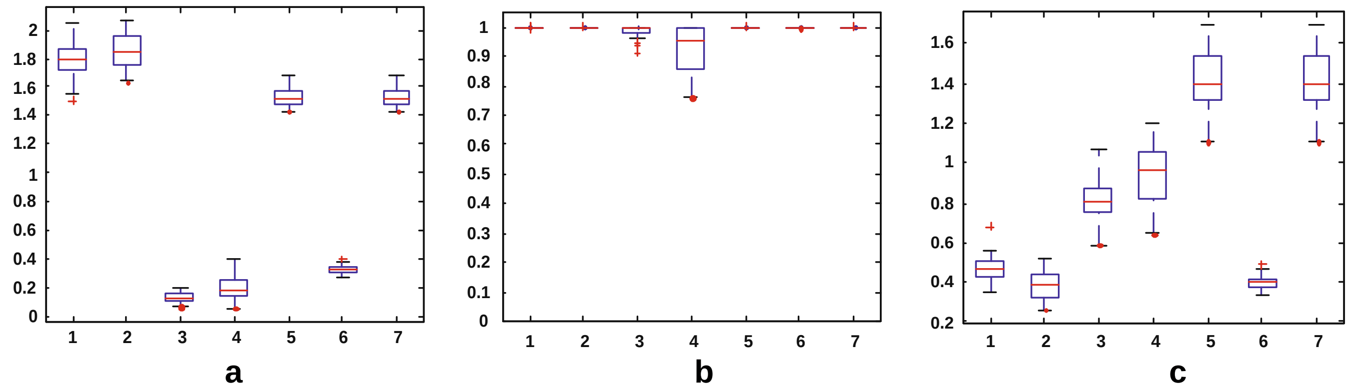

Fig. 12. Algorithm performance in (a) GD, (b) MS, and (c) S for DTLZ2.

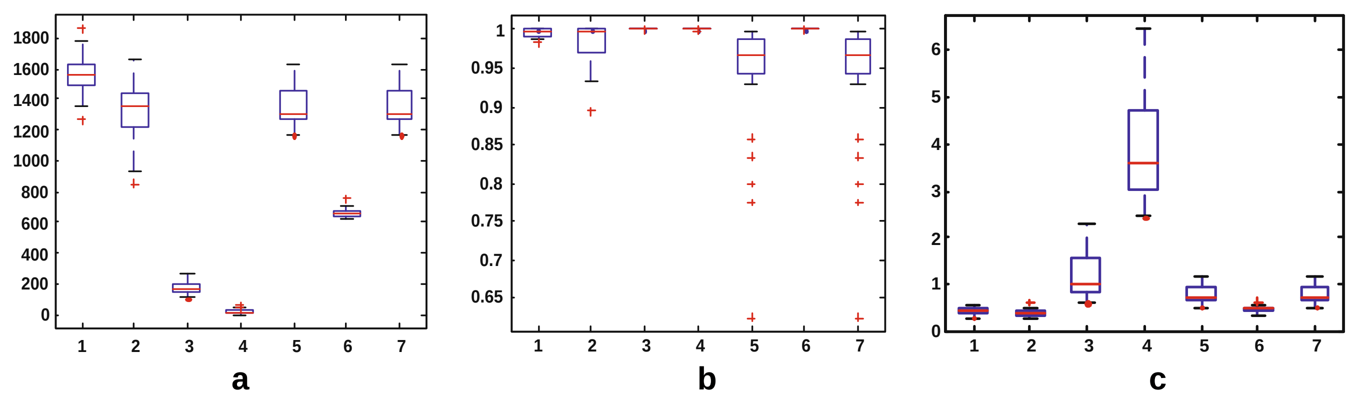

Fig. 13. Algorithm performance in (a) GD, (b) MS, and (c) S for DTLZ3.

bal Pareto-optimal front and an algorithm can get stuck at any of these local Pareto-optimal fronts, before converging to the global Pareto-optimal front. A stopping criterion of 50,000 evaluations is used for this problem. The distribution of the different performance metrics is represented by box-plots in Fig. 13.

As shown in Fig. 13, the PSO algorithms in general outperformed the rest of the algorithms in GD metric. From Fig. 13a, it can be noted that CCPSO is able to escape the local optimal traps as reflected by the good results obtained in GD. It can also be observed that, except CCPSO, other algorithms still have many solutions that are located far away from the true Pareto front. CCPSO is also able to evolve a diverse solution set as evident from Fig. 13b. But the solutions are not evenly distributed along the global Pareto front as shown by the large value on S. In general, the performance of CCPSO is the best among the seven algorithms used here on DTLZ3.

5. Conclusion

In the face of increasing complexity and dimensionality of today's application, there is a need to improve the efficiency and effectiveness of MOPSO. Therefore, in order to improve the performance of MOPSO, the algorithm put forward in this paper attempts to further emulate the conflict and coexistence between cooperation and competition in nature by implementing both aspects into co-evolutionary model. This is accomplished by having species compete amongst themselves for the right to represent more components, and the winners cooperate to solve the whole problem. The proposed algorithm is validated through extensive simulation studies, which include a sensitivity study and a comparative study. In the sensitivity analysis, the effects of different inertia weight, subswarm size and archive size settings on CCPSO performance are examined. We observed that CCPSO achieves better convergence with smaller inertia. On the other hand, the adaptive inertia scheme and higher inertia weights provide a more diverse Pareto front. In the case of subswarm sizing, we observed a tradeoff between diversity provided by a larger population size and convergence provided by the higher number of generations to explore. Increasing the archive size will bring about better performance in CCPSO. In the comparative study, CCPSO is compared against existing state-of-the-art multi-objective algorithms through the use of established benchmarks and metrics. It is observed that the particle swarm based algorithms have faster convergence speeds as compared to algorithms based on the evolutionary framework. However, such high speed of convergence results in poor convergence performance for PSO algorithms in ZDT4, which is characterized by a multi-modal landscape. The introduction of competitive–cooperative co-evolution enables CCPSO to retain the fast convergence speed of PSO while solving for near-optimal and diverse Pareto fronts in all test problems.

References

[1] C.A.C. Coello, M.R. Sierra, A co-evolutionary multi-objective evolutionary algorithm, in: Proceedings of 2003 Congress on Evolutionary Computation, vol. 1, 2003, pp. 482–489.
[2] C.A. Coello, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Transaction on Evolutionary Computation 8 (3) (2004) 256–279.
[3] K. Deb, Multi-objective genetic algorithms: Problem difficulties and construction of test problem, Evolutionary Computation 7 (3) (1999) 205–230.

[4] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.
[5] J.E. Fieldsend, R.M. Everson, S. Singh, Using unconstrained elite archives for multiobjective optimization, IEEE Transactions on Evolutionary Computation 7 (3) (2003) 305–323.
[6] C.K. Goh, K.C. Tan, An investigation on noisy environments in evolutionary multiobjective optimization, IEEE Transactions on Evolutionary Computation 11 (3) (2007) 354–381.
[7] C.K. Goh, K.C. Tan, A competitive–cooperative coevolutionary paradigm for dynamic multi-objective optimization, IEEE Transactions on Evolutionary Computation 13 (1) (2009) 103–127.
[8] E.J. Hughes, Evolutionary many-objective optimisation: Many once or one many? in: Proceedings of 2005 IEEE Congress on Evolutionary Computation, vol. 1, 2005, pp. 222–227.
[9] A. Iorio, X. Li, A cooperative co-evolutionary multi-objective algorithm using non-dominated sorting, in: Proceedings of Genetic and Evolutionary Computation Conference 2004, Lecture Notes in Computer Science, 2004, pp. 537–548.
[10] N. Keerativuttitumrong, N. Chaiyaratana, V. Varavithya, Multi-objective cooperative co-evolutionary genetic algorithm, in: Proceedings of Parallel Problem Solving From Nature VII, Lecture.
[11] V. Khare, X. Yao, K. Deb, Performance scaling of multi-objective evolutionary algorithms, in: Proceedings of the Second International Conference on Evolutionary Multi-criterion Optimization, 2003, pp. 376–390.
[12] E.F. Khor, K.C. Tan, T.H. Lee, C.K. Goh, A study on distribution preservation mechanism in evolutionary multi-objective optimization, Artificial Intelligence Review 23 (1) (2005) 31–56.
[13] J.D. Knowles, D.W. Corne, Approximating the nondominated front using Pareto archived evolutionary strategy, Evolutionary Computation 8 (2) (2000) 149–172.
[14] J. Lohn, W.F. Kraus, G.L. Haith, Comparing a co-evolutionary genetic algorithm for multi-objective optimization, in: Proceedings of 2002 Congress on Evolutionary Computation, 2002, pp. 1157–1162.
[15] S. Mostaghim, J. Teich, Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO), in: Proceedings of 2003 IEEE Swarm Intelligence Symposium, 2003, pp. 26–33.
[16] D.S. Liu, K.C. Tan, C.K. Goh, W.K. Ho, A multiobjective memetic algorithm based on particle swarm optimization, IEEE Transactions on Systems, Man and Cybernetics: Part B (Cybernetics) 37 (1) (2007) 42–50.
[17] D.S. Liu, K.C. Tan, S.Y. Huang, C.K. Goh, W.K. Ho, On solving multiobjective bin packing problems using evolutionary particle swarm optimization, European Journal of Operational Research 190 (2) (2008) 357–382.
[18] K. Maneeratana, K. Boonlong, N. Chaiyaratana, Multi-objective optimisation by co-operative co-evolution, in: Proceedings of Eighth International Conference on Parallel Problem Solving from Nature, 2004, pp. 772–781.
[19] J.R. Scott, Fault tolerant design using single and multi-criteria genetic algorithms, Masters Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1995.
[20] M.A. Potter, K.A. De Jong, Cooperative co-evolution: An architecture for evolving coadapted subcomponents, Evolutionary Computation 8 (1) (2000) 1–29.
[21] M.A. Potter, The design and analysis of a computational model of cooperative co-evolution, Ph.D. Thesis, George Mason University, 1997.
[22] C.D. Rosin, R.K. Belew, New methods for competitive co-evolution, Evolutionary Computation 5 (1) (1997) 1–29.
[23] Y. Shi, R. Eberhart, Parameter selection in particle swarm optimization, in: Proceedings of the Seventh Annual Conference on Evolutionary Programming, 1998, pp. 591–601.
[24] K.C. Tan, C.K. Goh, A.A. Mamun, E.Z. Ei, An evolutionary artificial immune system for multi-objective optimization, European Journal of Operational Research 187 (2) (2008) 371–392.
[25] K.C. Tan, C.Y. Cheong, C.K. Goh, Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation, European Journal of Operational Research 177 (2007) 813–839.
[26] K.C. Tan, Y.H. Chew, L.H. Lee, A hybrid multiobjective evolutionary algorithm for solving truck and trailer vehicle routing problems, European Journal of Operational Research 172 (2006) 855–885.
[27] K.C. Tan, Y.J. Yang, C.K. Goh, A distributed cooperative co-evolutionary algorithm for multi-objective optimization, IEEE Transactions on Evolutionary Computation 10 (5) (2006) 527–549.
[28] K.C. Tan, E.F. Khor, T.H. Lee, R. Sathikannan, An evolutionary algorithm with advanced goal and priority specification for multiobjective optimization, Journal of Artificial Intelligence Research 18 (2003) 183–215.
[29] K.C. Tan, T.H. Lee, E.F. Khor, Evolutionary algorithms with dynamic population and local exploration for multiobjective optimization, IEEE Transactions on Evolutionary Computation 5 (6) (2001) 565–588.
[30] F. van den Bergh, A.P. Engelbrecht, A cooperative approach to particle swarm optimization, IEEE Transaction on Evolutionary Computation 8 (3) (2004) 225–239.
[31] G. Venter, R.T. Haftka, A two species genetic algorithm for designing composite laminates subjected to uncertainty, in: Proceedings of 37th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 1996, pp. 1848–1857.
[32] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength pareto evolutionary algorithm, Technical Report 103, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 2001.
[33] C.T. Tseng, C.J. Liao, A discrete particle swarm optimization for lot-streaming flowshop scheduling problem, European Journal of Operational Research 191 (2) (2008) 360–373.
[34] L.H. Lee, E.P. Chew, S.Y. Teng, Y.K. Chen, Multi-objective simulation-based evolutionary algorithm for an aircraft spare parts allocation problem, European Journal of Operational Research 189 (2) (2008) 476–491.
[35] H. Ishibuchi, K. Narukawa, N. Tsukamoto, Y. Nojima, An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization, European Journal of Operational Research 188 (1) (2008) 57–75.
[36] F. Neumann, Expected runtimes of a simple evolutionary algorithm for the multi-objective minimum spanning tree problem, European Journal of Operational Research 181 (3) (2007) 1620–1629.