



A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D

Muhammad Asif Jan ^{a,*}, Rashida Adeeb Khanum ^b

^a Department of Mathematics, Kohat University of Science & Technology (KUST), Kohat 26000, Khyber Pakhtunkhwa, Pakistan

^b Department of Mathematical Sciences, University of Essex, Wivenhoe Park, Colchester, Essex CO4 3SQ, United Kingdom

ARTICLE INFO

Article history:

Received 24 October 2011

Received in revised form 20 June 2012

Accepted 23 July 2012

Available online 21 August 2012

Keywords:

Constrained multiobjective optimization

Constraint-domination principle

Decomposition

MOEA/D

Stochastic ranking

ABSTRACT

Penalty functions are frequently employed for handling constraints in *constrained optimization problems* (COPs). In penalty function methods, penalty coefficients balance objective and penalty functions. However, finding appropriate penalty coefficients to strike the right balance is often very hard. They are problems dependent. *Stochastic ranking* (SR) and *constraint-domination principle* (CDP) are two promising penalty functions based constraint handling techniques that avoid penalty coefficients. In this paper, the extended/modified versions of SR and CDP are implemented for the first time in the *multiobjective evolutionary algorithm based on decomposition* (MOEA/D) framework. This led to two new algorithms, CMOEA/D-DE-SR and CMOEA/D-DE-CDP. The performance of these new algorithms is tested on CTP-series and CF-series test instances in terms of the HV-metric, IGD-metric, and SC-metric. The experimental results are compared with NSGA-II, IDEA, and the three best performers of CEC 2009 MOEA competition, which showed better and competitive performance of the proposed algorithms on most test instances of the two test suits. The sensitivity of the performance of proposed algorithms to parameters is also investigated. The experimental results reveal that CDP works better than SR in the MOEA/D framework.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

This paper considers the following *constrained multiobjective optimization problem* (CMOP):

$$\begin{aligned} \text{Minimize } & F(x) = (f_1(x), f_2(x), \dots, f_m(x))^T; \\ \text{Subject to } & g_j(x) \geq 0, j = 1, \dots, p; \\ & l_k \leq x_k \leq u_k, k = 1, \dots, n; \end{aligned} \quad (1)$$

where $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ is an n dimensional vector of decision variables, F is the objective vector function that consists of m real-valued objective functions, and $g_j(x) \geq 0$ are inequality constraints. The objective and constraint functions, f_i 's and g_j 's, could be linear or non linear real-valued functions. l_k and u_k are the lower and upper bounds (called bound constraints) of x_k , $k = 1, \dots, n$, respectively, which define the search region $\mathcal{S} = \{x = (x_1, \dots, x_n)^T | l_k \leq x_k \leq u_k, k = 1, \dots, n\}$.

A solution which satisfies all constraints in (1) is called a feasible solution. More specifically, if for a solution $x = (x_1, \dots, x_n)^T$, $g_j(x) \geq 0$, $j = 1, \dots, p$ and $l_k \leq x_k \leq u_k$, $k = 1, \dots, n$, then it is called a feasible solution. The set of all feasible solutions is called the feasible region. Mathematically, we can write:

$$\mathcal{F} = \{x \in \mathcal{S} \subset \mathbb{R}^n | g_j(x) \geq 0, j = 1, \dots, p\}. \quad (2)$$

However, If a solution is not feasible, we call it infeasible. The set of all infeasible solutions is called the infeasible region.

The feasible *attainable objective set* (AOS) can be defined as $\{F(x) | x \in \mathcal{F}\}$.

More often, the objectives in (1) contradict one another. Thus, a single solution in the feasible search region could not minimize all the objectives simultaneously. Instead, a set of optimal compromising/tradeoff solutions that satisfy all constraints (i.e., feasible solutions) is desired. The best tradeoffs among the objectives can be defined in terms of Pareto-optimality [1–3].

A solution x is said to Pareto-dominate or simply dominate another solution y , mathematically denoted as $x \leq y$, if $f_i(x) \leq f_i(y)$,

* Corresponding author. Tel.: +92 3119012700; fax: +92 922554556.

E-mail addresses: janmathpk@gmail.com, majan@kust.edu.pk (M.A. Jan), rakhan@essex.ac.uk (R.A. Khanum).

$\forall i = 1, \dots, m$ and $f_i(x) < f_j(y)$ for at least one $j \in \{1, \dots, m\}$.¹ This definition of domination is sometimes referred to as a weak dominance relation. A solution $x^* \in \mathcal{F}$ is Pareto-optimal to (1) if there is no solution $x \in \mathcal{F}$ such that $F(x) \leq F(x^*)$. $F(x^*)$ is then called a Pareto-optimal (objective) vector. In other words, any enhancement in a Pareto-optimal solution in one objective must lead to degradation to at least one other objective. The set of all Pareto-optimal solutions is called the *Pareto set* (PS) in the decision space and *Pareto front* (PF) in the objective space [1].

In *multiobjective evolutionary algorithms* (MOEAs), better fitness values are mostly assigned to those individuals that are near the PF and in a less crowded region in the objective space. The two fundamental schemes for fitness assignment in MOEAs are Pareto-based ranking and decomposition [4].

In Pareto-based ranking, individuals are compared based on Pareto dominance. As a result, a scalar value (rank) is assigned to each individual in the population. Then, traditional selection operators can be used. The representative Pareto-based MOEAs include MOGA [5], NPGA [6], PAES [7], SPEA-II [8], and NSGA-II [9].

In decomposition based fitness assignment schemes, all individuals in the population are compared and ranked with respect to a weighted scalar optimization function. This function can be acquired by linearly or nonlinearly aggregating objectives of a CMOP. Contrary to Pareto-based ranking, here the fitness value of each individual is independent of other individuals in the population.

Different weighted scalar functions can be obtained by using different weight vectors, each one represents a particular search direction toward the PF. Therefore, in order to approximate the entire PF, one has to alter the weight vectors of a weighted scalar function during the search. The exemplary decomposition-based MOEAs are IMMOGLS [10], UGA [11], cMOGA [12], MOGLS [13], MOSPS [14], and MOEA/D [3].

The most popular approach to deal with constraints is to use penalty functions. In penalty function methods, appropriate penalty coefficients are required to balance objective and penalty functions. However, it is often very hard to find such appropriate coefficients [15]. They are problems dependent.

SR [15] and CDP [9] are two promising penalty function based constraint handling techniques that try to balance objective and penalty functions explicitly and directly, and thus avoid penalty coefficients. Since they do not employ any penalty coefficient in the adopted penalty functions, a priori knowledge of the constrained problem at hand is also not needed. SR is inspired from the need of balancing objective and penalty functions directly and explicitly in constrained optimization [15]. In SR, a small percentage of infeasible solutions is compared based on objective function values, and the rests are compared based on constraint violation. While in CDP, they are compared based on constraint violation only. Moreover, CDP favors feasible solutions over infeasible solutions.

In this paper, we first modify and then implement these two techniques in the framework of MOEA/D-DE [16], an improved version of MOEA/D.

The rest of this paper is organized as follows. Section 2 presents the two commonly used weight-based decomposition approaches. Section 3 briefly introduces MOEA/D and adapts the algorithmic framework of MOEA/D-DE for CMOPs. Section 4 illustrates constraint handling techniques SR and CDP. Section 5 discusses the experimental settings. Section 6 introduces the metrics that will be used for the performance assessment of the algorithms proposed in this paper. Section 7 shows and discusses the experimental

results on CTP-series [2,17] and CF-series [18] test instances. Section 8 compares our experimental results with NSGA-II [9], IDEA [19], and the three best performers [20–22] of CEC 2009 MOEA competition. Section 9 comments on the sensitivity of the performance of the suggested algorithms to the parameters involved. Finally, Section 10 concludes this paper with a summary of the work carried out.

2. Weight-based decomposition approaches for multiobjective optimization

A lot of decomposition techniques have been developed in mathematical programming [1,23]. The two commonly used weight-based decomposition techniques are weighted sum approach [24,25] and weighted Tchebycheff approach [1]. The recent ones include normal-boundary intersection method [26], normalized normal constraint method [27], and *penalty-based boundary intersection* (PBI) method [3,4]. In the following, we present the two commonly used weight-based decomposition approaches and their Pareto optimality conditions.

2.1. Weighted sum approach

The weighed sum approach considers a convex combination of the objective functions. Its aggregation function is of the following form [1]:

$$\text{Minimize } g^{ws}(x|\lambda) = \sum_{i=1}^m \lambda_i f_i(x); \quad (3)$$

Subject to $x \in \mathcal{F}$;

where $\lambda = (\lambda_1, \dots, \lambda_m)^T$ is a weight vector, i.e., $\lambda_i \geq 0 \forall i = 1, \dots, m$ and $\sum_{i=1}^m \lambda_i = 1$.

Theorem 1 ([1]). *The solution of weighting scalar optimization problem (3) is Pareto-optimal to the CMOP (1) if the weighting coefficients are positive.*

Theorem 2 ([1]). *If $x^* \in \mathcal{F}$ is a unique solution of the weighting scalar optimization problem (3), then x^* is Pareto-optimal to the CMOP (1).*

Theorem 3 ([1]). *Let the CMOP (1) be convex. If $x^* \in \mathcal{F}$ is Pareto-optimal, then there exists a weight vector $\lambda = (\lambda_1, \dots, \lambda_m)^T$ with $\lambda_i \geq 0 \forall i = 1, \dots, m$ and $\sum_{i=1}^m \lambda_i = 1$ such that x^* is the optimal solution of the weighting scalar optimization problem (3).*

Theorems 1 and 2 demonstrate the conditions for the Pareto optimality of solutions of the weighted sum problem given in (3). Theorem 3 claims that if the given CMOP is convex, then any Pareto-optimal solution of it can be obtained by the weighted sum approach. Here, different Pareto-optimal solutions can be found by changing the weighting coefficients in the weighted sum problem (3).

2.2. Weighted Tchebycheff approach

The superiority of weighted Tchebycheff approach over the weighted sum approach is that it is less sensitive to the shape of PF. Moreover, it can be used to find the Pareto-optimal solutions in both convex and nonconvex PFs. It is defined as follows [1]:

$$\text{Minimize } g^{te}(x|\lambda, z^*) = \max_{1 \leq i \leq m} \{\lambda_i |f_i(x) - z_i^*|\}; \quad (4)$$

Subject to $x \in \mathcal{F} \subset \mathcal{R}^n$;

where $z^* = (z_1^*, \dots, z_m^*)^T$ is the reference point, i.e., $z_i^* = \min\{f_i(x) | x \in \mathcal{F}\} \forall i = 1, \dots, m$.

Theorem 4 ([1]). *The optimal solution of the weighted Tchebycheff problem (4) is weakly Pareto-optimal, if all weighting coefficients are positive.*

¹ This definition of domination is for minimization. All the inequalities should be reversed if the goal is to maximize the objectives in (1). “dominate” means “be better than”.

Theorem 5 ([1]). If the weighted Tchebycheff problem (4) has a unique solution, then it is Pareto-optimal.

Theorem 6 ([23]). Let x^* be weakly Pareto-optimal, then there exists a weight vector $\lambda = (\lambda_1, \dots, \lambda_m)^T$ with $\lambda_i \geq 0 \forall i = 1, \dots, m$ and $\sum_{i=1}^m \lambda_i = 1$ such that x^* is the optimal solution of the weighted Tchebycheff problem (4).

According to Theorem 6, for each Pareto-optimal solution x^* of the CMOP (1), there exists a weight vector λ such that x^* is the optimal solution of the weighted Tchebycheff problem (4) even if the CMOP (1) is not convex.

3. Multiobjective evolutionary algorithm based on decomposition

Zhang and Li [3] proposed a simple yet efficient MOEA, called *multiobjective evolutionary algorithm based on decomposition* (MOEA/D). MOEA/D tackles the problem of approximation of the PF by explicitly decomposing an MOP into a number of scalar objective optimization subproblems (In this paper, we use the Tchebycheff aggregation function for this purpose.). These subproblems are then optimized concurrently and collaboratively by evolving population of solutions using an EA. The neighborhood relations among these subproblems are defined based on the Euclidean distances between their aggregation coefficient vectors. Optimization of a subproblem uses the information, mainly from its neighboring subproblems.

MOEA/D has a number of distinctive versions (e.g., MOEA/D-DE [16], MOEA/D-DRA [18]). MOEA/D-DE [16] is one of the effective versions of MOEA/D. It distinguishes from its predecessor (MOEA/D) in the following aspects [16]:

- It picks out three parent solutions from the whole population with a low probability in order to ameliorate the exploration ability of the search.
- It limits the maximal number of solutions replaced by a better child solution.
- It employs DE [28] operator for generating new child solutions.

We modified the algorithmic framework of MOEA/D-DE to solve CMOPs (e.g., CTP-series [2,17] and CF-series [18] test instances). The pseudo-code of the modified framework, denoted by CMOEA/D-DE, is given in Algorithm 1:

Algorithm 1. Pseudo-code of CMOEA/D-DE.

```

1: Generate weight vectors  $\lambda^i = (\lambda_1^i, \dots, \lambda_m^i)^T$ ,  $i = 1, \dots, N$  according to the criteria described in Section 5.2;
2: Compute the Euclidean distances between any two weight vectors;
3: Find the  $T$  closest weight vectors to each weight vector, set  $B(i) = \{\lambda^{i_1}, \dots, \lambda^{i_T}\}$  where  $\lambda^{i_1}, \dots, \lambda^{i_T}$  are the  $T$  closest weight vectors to the weight vector  $\lambda^i$ ;
4: Sample uniformly randomly an initial population  $\{x^i, i = 1, \dots, N\}$  from the search region;
5: Evaluate each  $x^i$ ,  $F(x^i) = (f_1(x^i), \dots, f_m(x^i))$ ,  $V(x^i) = |\sum_{j=1}^p \min(g_j(x^i), 0)|$ ,  $i = 1, \dots, N$ .  $V(x^i)$  is the degree of constraint violation of individual  $x^i$ . If  $V(x^i) = 0$ , individual  $x^i$  is feasible; otherwise, it is infeasible;
6: Initialize  $z = (z_1, \dots, z_m)^T$  by setting  $z_i = \min\{f_i(x^1), f_i(x^2), \dots, f_i(x^N)\}$ ;
7: Set  $gen = 1$ ;
8: while Stopping criterion is not satisfied do
9:   for  $i = 1$  to  $N$  do
10:    Select mating pool/update range and generate an offspring  $y$  according to Algorithm 2;
11:    Evaluate  $y$ ,  $F(y) = (f_1(y), \dots, f_m(y))$ ,  $V(y) = |\sum_{j=1}^p \min(g_j(y), 0)|$ ;
12:    Update  $z$ , for each  $j = 1, \dots, m$ , if  $z_j > f_j(y)$ , then set  $z_j = f_j(y)$ ;
13:    Update neighboring solutions according to Algorithm 3;
14:   end for
15: end while
16: Output:  $\{x^1, \dots, x^N\}$ ,  $\{f(x^1), \dots, f(x^N)\}$ , and  $\{V(x^1), \dots, V(x^N)\}$ .

```

Algorithm 2. Pseudo-code of mating/update range selection and reproduction.

```

1: Uniformly randomly generate a number  $rand$  from  $(0, 1)$ ;
2: if  $rand < \delta$  then
3:    $P = B(i)$ ;
4: else
5:    $P = \{1, \dots, N\}$ ;
6: end if
7: Set  $r_1 = i$  and randomly select two indexes  $r_2$  and  $r_3$  from  $P$ , and then generate an offspring solution  $y'$  from  $x^{r_1}$ ,  $x^{r_2}$ , and  $x^{r_3}$  by a DE operator;
8: Perform a mutation operator on  $y'$  with probability  $p_m$  to produce a new solution  $y$ ;
9: Repair  $y$ , if an element of  $y$  is out of the boundary by randomly selecting its value inside the boundary;

```

Algorithm 3. Pseudo-code of update scheme.

```

1: Update the set  $P$  of neighboring solutions of  $y$  as follows:
2: Set  $c = 0$  and then do the following:
3: if  $c = n_r$  or  $P$  is empty then
4:   return;
5: else
6:   Randomly pick an index  $j$  from  $P$ ;
7:   Compute the Tchebycheff aggregation function values of  $y$  and  $x^j$ ;
8:   if  $g^{te}(y|\lambda^j, z) \leq g^{te}(x^j|\lambda^j, z)$  then
9:      $x^j = y$ ,  $f(x^j) = f(y)$ ,  $V(x^j) = V(y)$ , and  $c = c + 1$ ;
10:   end if
11:   Remove  $j$  from  $P$  and go to step 3;
12: end if

```

In the DE operator used in step 7 of Algorithm 2, each y'_k in $y' = (y'_1, \dots, y'_n)^T$ is generated as follows [28]:

$$y'_k = \begin{cases} x_k^{r_1} + F \times (x_k^{r_2} - x_k^{r_3}), & \text{with probability } CR; \\ x_k^{r_1}, & \text{with probability } 1 - CR, \end{cases}$$

where CR and F are two control parameters.

The mutation operator in step 8 of Algorithm 2 generate $y = (y_1, \dots, y_n)^T$ from y' in the following way [2]:

$$y_k = \begin{cases} y'_k + \sigma_k \times (u_k - l_k), & \text{with probability } p_m; \\ y'_k, & \text{with probability } 1 - p_m, \end{cases}$$

with

$$\sigma_k = \begin{cases} (2 \times rand)^{\frac{1}{\eta+1}} - 1, & \text{if } rand < 0.5; \\ 1 - (2 - 2 \times rand)^{\frac{1}{\eta+1}}, & \text{otherwise,} \end{cases}$$

where $rand$ is a uniformly random number from $[0, 1]$. The distribution index η and the mutation rate p_m are two control parameters.

4. Stochastic Ranking and constraint-domination principle

In this section, we introduce the two selected penalty function based constraint handling techniques: SR and CDP.

4.1. Stochastic Ranking

Runarsson and Yao [15] proposed “Stochastic Ranking (SR)” for balancing objective and penalty functions directly and explicitly in the employed penalty function. In this approach, a probability parameter p_f decides the use of the objective function only for comparisons in ranking in the infeasible region of the search space. That is, if a given pair of adjacent individuals are both feasible, then $p_f = 1$, and the one with better objective function value is ranked higher than the other. Else, namely, when one individual is feasible and the other is infeasible or both are infeasible. It is p_f . In the latter

case, if a uniformly generated random number r between 0 and 1 is less than p_f , then the one with better objective value is ranked higher than the other; otherwise, the one with a smaller degree of constraint violation is ranked higher than the other.

In our case, if a contesting parent and offspring are both feasible or $r < p_f$, and if the offspring's Tchebycheff aggregation function value is smaller than that of the parent's Tchebycheff aggregation function value. It replaces the parent. Otherwise, if the offspring's degree of constraint violation is less than that of the parent, it substitutes the parent. We take up the modified SR constraint handling strategy in the update scheme of CMOEA/D-DE that results in a new algorithm, termed as CMOEA/D-DE-SR. The pseudo-code of the update scheme of CMOEA/D-DE-SR is given in [Algorithm 4](#):

Algorithm 4. Update scheme of CMOEA/D-DE-SR. rand is random number generator between 0 and 1.

```

1:   Update neighboring solutions  $x^j, j \in P$  of an offspring  $y$  as follows:
2:   Set  $c = 0$  and then do the following:
3:   if  $c = n_r$  or  $P$  is empty then
4:     break;
5:   else
6:     Randomly pick an index  $j$  from  $P$ ;
7:      $r = \text{rand}(0, 1)$ ;
8:     if  $(V(y) = V(x^j) = 0) \text{ or } (r < p_f)$  then
9:       if  $g^{te}(y|\lambda^j, z) \leq g^{te}(x^j|\lambda^j, z)$  then
10:         $x^j = y, f(x^j) = f(y), V(x^j) = V(y)$ , and  $c = c + 1$ ;
11:      end if
12:    else
13:      if  $V(y) < V(x^j)$  then
14:         $x^j = y, f(x^j) = f(y), V(x^j) = V(y)$ , and  $c = c + 1$ ;
15:      end if
16:    end if
17:    Remove  $j$  from  $P$  and go to step 3;
18:  end if

```

4.2. Constraint-domination principle

In [29], Deb proposed the following penalty-parameterless function, where infeasible solutions are compared based only on their constraint violation [29]:

$$f_p(x) = \begin{cases} f(x), & \text{if } g_j(x) \geq 0, \forall j = 1, \dots, q; \\ f_{\max} + \sum_{i=1}^m \langle g_i(x) \rangle, & \text{otherwise,} \end{cases} \quad (5)$$

where $\langle \cdot \rangle$ denotes the absolute value of the operand, if the operand is negative, and returns a value zero, otherwise. f_{\max} is the objective function value of the worst feasible solution in the current population. If no feasible solution exists in the current population, f_{\max} is set to zero. Thus, the infeasible solution's fitness depends on the amount of constraint violations and on the population of solutions at hand. While, the feasible solution's fitness is always fixed and is equal to its objective function value only.

In NSGA-II [9], Deb et al., extended the above constraint handling technique to constrained multiobjective optimization. They simply revised the definition of domination between two solutions to work with their Pareto ranking based MOEA, NSGA-II and suggested the so-called constrained NSGA-II. The revised definition of domination is given as follows. A solution x is said to constraint-dominate another solution y , if any of the following criterion is satisfied [9]:

1. x is feasible and y is infeasible.
2. Both x and y are infeasible and x has a smaller degree of constraint violation than y .
3. Both x and y are feasible and x dominates y .

We implement the CDP with condition 3 examined with the Tchebycheff aggregation function values instead in the update scheme of CMOEA/D-DE. That is, if a contesting parent and offspring

are both feasible, and if the offspring Tchebycheff aggregation function value is smaller than that of the parent value, then the offspring replaces the parent. This results in a new algorithm, termed as CMOEA/D-DE-CDP. The pseudo-code of the update scheme of CMOEA/D-DE-CDP is given in [Algorithm 5](#):

Algorithm 5. Update scheme of CMOEA/D-DE-CDP.

```

1:   Update neighboring solutions  $x^j, j \in P$  of an offspring  $y$  as follows:
2:   Set  $c = 0$  and then do the following:
3:   if  $c = n_r$  or  $P$  is empty then
4:     break;
5:   else
6:     Randomly pick an index  $j$  from  $P$ ;
7:     if  $(V(y) = 0 \& V(x^j) > 0)$  then
8:        $x^j = y, f(x^j) = f(y), V(x^j) = V(y)$ , and  $c = c + 1$ ;
9:     else if  $(V(y) > 0 \& V(x^j) > 0)$  then
10:      if  $V(y) < V(x^j)$  then
11:         $x^j = y, f(x^j) = f(y), V(x^j) = V(y)$ , and  $c = c + 1$ ;
12:      end if
13:    else if  $(V(y) = V(x^j) = 0)$  then
14:      if  $g^{te}(y|\lambda^j, z) \leq g^{te}(x^j|\lambda^j, z)$  then
15:         $x^j = y, f(x^j) = f(y), V(x^j) = V(y)$ , and  $c = c + 1$ ;
16:      end if
17:    end if
18:    Remove  $j$  from  $P$  and go to step 3;
19:  end if

```

The constraint handling techniques SR and CDP can be differentiated by the way infeasible solutions are compared. Moreover, the two techniques are equivalent when $p_f = 0$.

5. Experimental settings

Throughout this paper, unless otherwise said, we will keep the following parameters' settings and weight vectors' selection criteria.

5.1. Parameters settings

The parameters' settings are same as in [19] for CTP-series and in [18] for CF-series test instances. They are given as follows:

- Population size N :
 - $N = 200$ for 2-objective CTP-series test instances, CTP1–CTP8;
 - $N = 600$ for 2-objective CF-series test instances, CF1–CF7;
 - $N = 1000$ for 3-objective CF-series test instances, CF8–CF10;
- Dimensionality of variables: The number of variables is set to 2 in CTP-series test instances, while it is set to 10 in CF-series test instances.
- Neighborhood size, $T = 0.1N$;
- Update number, $n_r = 0.01N$;
- Update probability, $\delta = 0.9$;
- Parameter values for DE and mutation operators: $CR = 1.0$ and $F = 0.5$, $\eta = 20$ and $p_m = 1/n$, where n is the number of dimension;
- Maximal number of generations: The maximal number of generations is set to 200 for the CTP-series test instances, CTP1–CTP8, 500 for the 2-objective CF-series test instances, CF1–CF7, and 300 for the 3-objective CF-series test instances, CF8–CF10.
- Stopping criterion: The algorithm stops after 40,000 function evaluations for test instances CTP1–CTP8 and after 300,000 function evaluations for test instances CF1–CF10.
- Number of runs: Each algorithm is run 30 times independently for each test instance.

5.2. Weight vectors selection

A set of N weight vectors, W , is originated by using the following criteria [18]:

1. Form a set W_1 comprising of uniformly randomly generated 5000 weight vectors.
2. Initialize set W as the set holding all the weight vectors $(1, 0, \dots, 0, 0), (0, 1, \dots, 0, 0), \dots, (0, 0, \dots, 0, 1)$.
3. Find the weight vector in set W_1 with the largest distance to set W , add it to set W and delete it from set W_1 .
4. If the size of set W is equal to the size of the population N , then stop and return set W . Otherwise, go to step 3.

6. Performance metrics

Contrary to *single objective optimization* (SOO), where one measures the quality of a single solution, one checks the quality of a set of nondominated solutions in *multiobjective optimization* (MOO). Generally, two aspects of the final nondominated solutions are worked out for this purpose: convergence and diversity. Convergence depicts closeness of the final nondominated solutions to the true PF, whereas diversity targets on the distribution of the final solutions along the true PF.

In [2], a number of performance metrics has been cited, including hypervolume, generational distance, and set coverage, etc. However, none of the cited performance metrics can credibly evaluate both the convergence and the diversity of nondominated solutions [30,31]. Thus, one has to exercise several metrics in the performance evaluation of MOEAs. In this paper, we will use the *hypervolume* metric (HV-metric) [2] statistics for comparing results on CTP-series test instances, CTP1–CTP8, and the *inverted generational distance* metric (IGD-metric) [3,32] statistics for comparing results on CF-series test instances, CF1–CF10. Moreover, we will also employ the *set coverage* metric (SC-metric) [3] to compare the nondominated solutions obtained by different algorithms. The reason for choosing the first two metrics is that the algorithms in comparison have also used these performance metrics. They are detailed as follows.

6.1. Performance metric hypervolume

Hypervolume metric (HV-metric) computes the volume in the function space covered by the elements of a set P for problems with minimizing objectives [33,34]. Mathematically, a hypercube v_i is assembled for each solution $i \in P$ with a reference point R and the solution i as the diagonal corners of the hypercube. Then, a union of all these hypercubes is taken and its HV is calculated [2]:

$$HV = \text{volume} \left(\bigcup_{i=1}^{|P|} v_i \right). \quad (6)$$

In this paper, the reference point R used for test instances CTP6 and CTP8 is $(2, 20)$, while for the rest of the test instances, it is $(2, 2)$.

6.2. Performance metric inverted generational distance

The *inverted generational distance* metric (IGD-metric) is defined as follows [3,32].

Suppose P^* and P be two sets of uniformly distributed points and approximate points along/to the PF, respectively. The mean distance from P^* to P is given as follows:

$$D(P, P^*) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|}, \quad (7)$$

where $d(v, P)$ is the minimum Euclidean distance between v and the points in P . If P^* is large enough to represent the PF very well, $D(P, P^*)$ could measure both the diversity and convergence of P in a sense [3].

6.2.1. Solutions' selection for IGD-metric calculation

For calculating the IGD-metric values, we select 100 feasible nondominated solutions in the case of 2-objective and 150 in the

Table 1

The HV-metric statistics based on 30 independent runs of CMOEA/D-DE-SR (indicated by SR) and CMOEA/D-DE-CDP (indicated by CDP). The results in boldface indicate the better results. ‘-’ means that no feasible solution is found.

Test instance	Best (highest)		Mean		St. dev.	
	SR	CDP	SR	CDP	SR	CDP
CTP1	2.7633	2.7647	2.7624	2.7646	0.0007	0.0001
CTP2	3.0528	3.0595	3.0508	3.0594	0.0014	0.0001
CTP3	3.0008	3.0282	2.9881	3.0238	0.0059	0.0024
CTP4	2.8266	2.9761	2.7129	2.9147	0.0457	0.0342
CTP5	3.0064	3.0382	2.9816	3.0302	0.0167	0.0062
CTP6	–	–	–	–	–	–
CTP7	–	3.6162	3.6125	3.6128	3.6124	0.0011
CTP8	–	–	–	–	–	–

case of 3-objective test instances from each final population. The final solution set P is selected from the output set $O = \{F(x^1), \dots, F(x^N)\}$ of the algorithm as follows [18]:

- For the instances with two objectives, the set P of 100 final solutions consists of the best solutions in set O for the subproblems with weights $(0, 1), (1/99, 98/99), \dots, (98/99, 1/99), (1, 0)$.
- For the instances with three objectives, the following criteria are followed:
 1. Randomly select an element e from O and set $O_1 = O \setminus \{e\}$ and $P = \{e\}$.
 2. Find the element in O_1 with the largest distance to P , delete it from O_1 and add it to P .
 3. If the size of P becomes 150, stop; otherwise, go to (2).

6.3. Performance metric set coverage

The *set coverage* metric (SC-metric) is defined as follows [3].

Suppose A_1 and A_2 be two approximations to the PF of a CMOP. $SC(A_1, A_2)$ gives the percentage of the solutions in A_2 that are dominated by at least one solution in A_1 , i.e.,

$$SC(A_1, A_2) = \frac{|\{\mathbf{u} \in A_2 | \exists \mathbf{v} \in A_1 : \mathbf{v} \text{ dominates } \mathbf{u}\}|}{|A_2|} \quad (8)$$

It may be noted that it is not necessary that $SC(A_1, A_2) = 1 - SC(A_2, A_1)$. Moreover, if $SC(A_1, A_2) = 1$, then all solutions in A_2 are dominated by some solutions in A_1 , and if $SC(A_1, A_2) = 0$, then no solution in A_2 is dominated by a solution in A_1 .

7. Experimental results and discussion

In this section, we present the experimental results obtained from CMOEA/D-DE-SR and CMOEA/D-DE-CDP on CTP-series and CF-series test instances.

7.1. Experimental results on CTP-series test instances

Table 1 shows the HV-metric statistics based on 30 independent runs of CMOEA/D-DE-SR with $p_f = 0.05$ (our extensive experiments with different p_f values reveal that CMOEA/D-DE-SR can obtain comparative results with $p_f = 0.05$, see Section 9.2) and CMOEA/D-DE-CDP for the CTP-series test instances. These statistics are based on feasible solutions found in the final populations of each algorithmic run and include the best (i.e., highest), mean, and standard deviation values of the HV-metric.

From the experimental results in **Table 1**, it is clear that all the statistics of CMOEA/D-DE-SR with $p_f = 0.05$ are worse than CMOEA/D-DE-CDP for test instances CTP1–CTP5, CTP7 except the best and mean values of CTP7. However, looking at the plots of the nondominated front with the best HV-metric value and all the 30

Table 2

The average set coverage between CMOEA/D-DE-SR (indicated by SR) and CMOEA/D-DE-CDP (indicated by CDP) on CTP-series test instances. The results in boldface indicate the better results. ‘–’ means that no feasible solution is found.

Test instance	C(CDP, SR)	C(SR, CDP)
CTP1	0.25	0.01
CTP2	0.81	0.01
CTP3	0.99	0.02
CTP4	0.98	0.02
CTP5	0.88	0.02
CTP6	–	–
CTP7	0.23	0.02
CTP8	–	–

final nondominated fronts for CTP7 in Fig. 2, it can be observed that CMOEA/D-DE-CDP consistently found better nondominated solutions. Thus, the better best and mean HV-metric values obtained by CMOEA/D-DE-SR are not desirable as compared to the ones acquired with CMOEA/D-DE-CDP.

For test instances CTP6 and CTP8, both algorithms fail to find any feasible solution in all 30 algorithmic runs. They get trapped in the infeasible region and finally converge to more than one infeasible solutions in case of CMOEA/D-DE-SR and to a single infeasible solution in case of CMOEA/D-DE-CDP (see Fig. 2). These two test instances are hard problems, since the constraints in them split the search space into feasible and infeasible bands of varying widths. The feasible bands are marked with F in Fig. 2. In test instance CTP6, the unconstrained PF is infeasible and the PF lies on a part of the constraint boundary. The bands in it are also parallel to unconstrained PF (see Fig. 2). In test instance CTP8, the constraints generate disconnected islands of feasible objective space (see Fig. 2). The reason for the failure of both algorithms is the presence of such hard constraints in these two test instances. They cause highly infeasible initial populations. For example, we observed in a run of CMOEA/D-DE-SR on test instance CTP8 that the minimum degree of normalized constraint violation of an infeasible solution in the initial population is 0.4286, while at the end of the algorithmic run, it is 0.3737. As a result, CMOEA/D-DE-SR comparing infeasible solutions based on aggregation function values with a small p_f value and rests based on constraint violation and CMOEA/D-DE-CDP comparing infeasible solutions based on constraint violation only get confined in the infeasible region due to the replacement and update schemes in these algorithms. The infeasible solutions eventually converge to multiple or to a single infeasible solution even with a small update number, n_r .

Table 2 presents the average of the SC-metric values of the final nondominated fronts obtained by CMOEA/D-DE-SR with $p_f = 0.05$ and CMOEA/D-DE-CDP for the CTP-series test instances. The results of this table disclose that, in terms of the SC-metric, the nondominated solutions found by CMOEA/D-DE-CDP are much better than CMOEA/D-DE-SR for test instances CTP1–CTP5, CTP7. For example, considering test instance CTP3, 99% of the nondominated solutions of CMOEA/D-DE-SR are dominated by those of CMOEA/D-DE-CDP, while only 2% vice versa.

Figs. 1 and 2 plot, in the function space, the nondominated solutions with the best (i.e., highest) HV-metric value found by CMOEA/D-DE-SR with $p_f = 0.05$ and CMOEA/D-DE-CDP in 30 independent runs for CTP-series test instances. To show the divergence of the nondominated solutions found by both algorithms, all the 30 final nondominated fronts found are also plotted in these figures.

These figures clearly show that CMOEA/D-DE-CDP consistently approximated well the PFs of six test instances CTP1–CTP5, and CTP7. CMOEA/D-DE-SR with $p_f = 0.05$, on the other hand, found good approximations of the PFs for three test instances CTP1, CTP2, and CTP7. Although it distributed well the final solutions along the

Table 3

The IGD-metric statistics based on 30 independent runs of CMOEA/D-DE-SR (indicated by SR) and CMOEA/D-DE-CDP (indicated by CDP). The results in boldface indicate the better results.

Test instance	Best (lowest)		Mean		St. dev.	
	SR	CDP	SR	CDP	SR	CDP
CF1	0.0005	0.0002	0.0015	0.0006	0.0016	0.0002
CF2	0.0024	0.0026	0.0033	0.0040	0.0016	0.0018
CF3	0.0767	0.0632	0.1437	0.1382	0.0448	0.0441
CF4	0.0054	0.0051	0.0103	0.0092	0.0046	0.0030
CF5	0.0343	0.0353	0.1388	0.1843	0.1109	0.1278
CF6	0.0091	0.0074	0.0292	0.0261	0.0199	0.0166
CF7	0.0494	0.0381	0.1894	0.1718	0.0917	0.0957
CF8	0.0397	0.0404	0.0435	0.0441	0.0024	0.0028
CF9	0.0447	0.0449	0.0489	0.0497	0.0029	0.0030
CF10	0.1169	0.1123	0.2911	0.2563	0.1651	0.1453

PFs (or along different parts of the PFs), but failed in converging well to them for other three test instances CTP3–CTP5. As to the convergence to the PFs, uniformness of the nondominated solutions and small variance in the objective space, CMOEA/D-DE-CDP obviously outperforms CMOEA/D-DE-SR on these six test instances.

Fig. 2 shows that both CMOEA/D-DE-SR and CMOEA/D-DE-CDP failed in approximating the PFs of test instances CTP6 and CTP8. The reason for the failure of both algorithms has been discussed above.

Figs. 3 and 4 present the evolution of the average HV-metric values versus function evaluations and the average generation feasibility versus generations in CMOEA/D-DE-SR and CMOEA/D-DE-CDP for six CTP-series test instances, CTP1–CTP5 and CTP7. The generation feasibility means the ratio of the number of feasible solutions in a generation to the population size.

From Fig. 3, it is very clear that CMOEA/D-DE-CDP can find higher HV-metric values than those obtained from CMOEA/D-DE-SR on test instances CTP1–CTP5. The HV-metric values of both algorithms are same after the initial difference in few early generations on test instance CTP7.

Fig. 4 shows that CMOEA/D-DE-CDP converges to the feasible region quickly, and the feasibility ratio becomes 1 mostly by generation 10 for the six test instances, CTP1–CTP5 and CTP7. It remains close to 1 in CMOEA/D-DE-SR except test instance CTP1. Both CDP and the replacement and update scheme help CMOEA/D-DE-CDP to converge quickly to the feasible region. However, in CMOEA/D-DE-SR, a small percentage of comparisons in the infeasible region based on aggregation function values facilitates retaining some infeasible solutions in the population.

Moreover, the overall PF of CTP1 is made of some portion of the unconstrained PF and some parts of the constraints boundaries, while those of CTP2–CTP5 and CTP7 consists of disconnected discrete/continuous solutions/regions. Thus, in order to approach them much diversity is required in the population right from the beginning of the algorithmic runs. CMOEA/D-DE serves this purpose, since it defines separate scalar objective subproblems for each solution, which are then optimized simultaneously and collaboratively. Therefore, the quick convergence to the feasible regions and, as a result, more exploration of them by CMOEA/D-DE, could be one of the reasons for better performance of CMOEA/D-DE-CDP than CMOEA/D-DE-SR on test instances CTP1–CTP5 and CTP7. Nonetheless, keeping some good infeasible solutions during evolution process could also provide comparative results for some test instances (see Table 1 for the HV-metric statistics and Table 3 for the IGD-metric statistics of CMOEA/D-DE-SR).

Furthermore, the graphs for test instances CTP6 and CTP8 are not presented, since both algorithms fail to find any feasible solutions for these two instances.

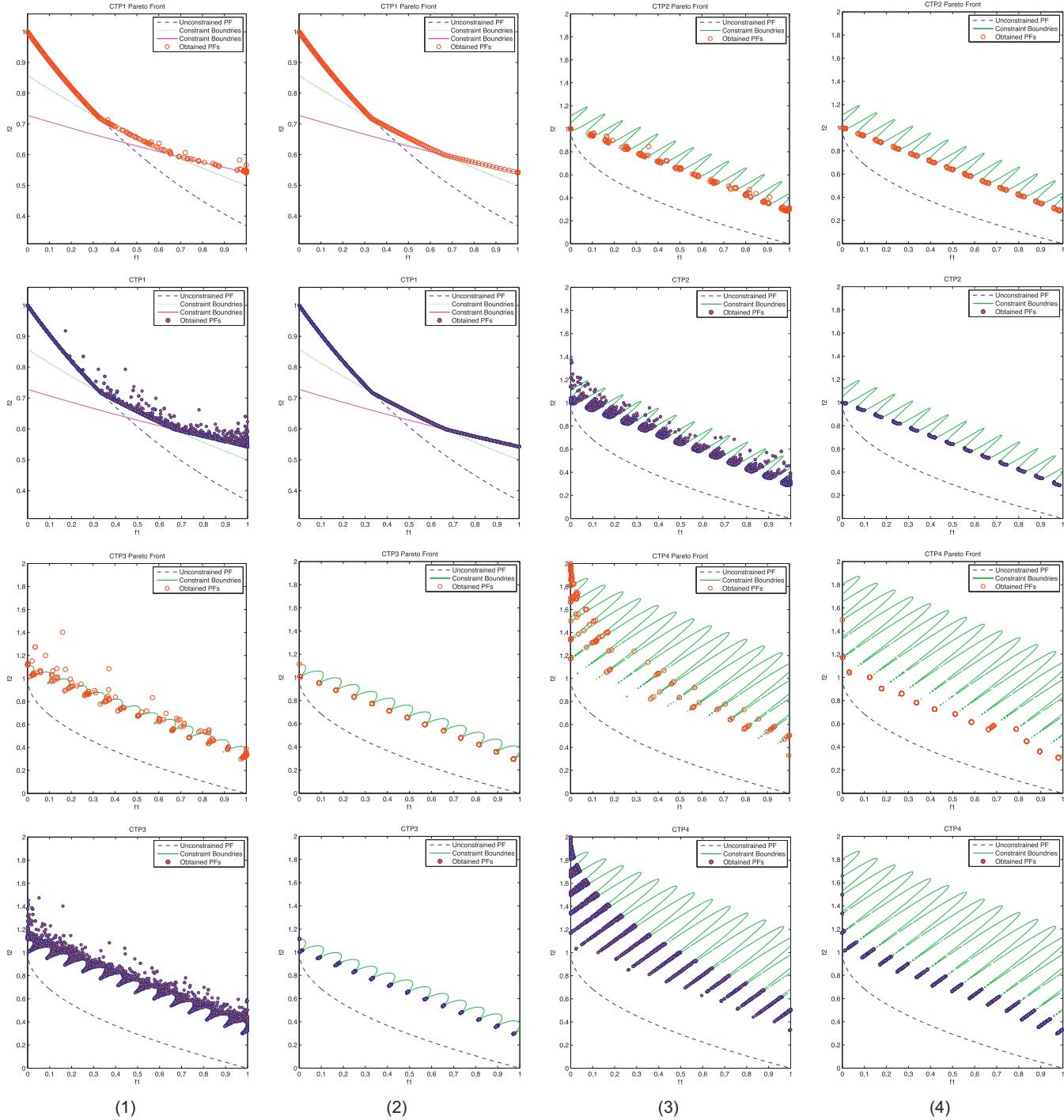


Fig. 1. Plots of the nondominated front with the best HV-metric value and all the 30 final nondominated fronts found by CMOEA/D-DE-SR (columns 1 and 3) and CMOEA/D-DE-CDP (columns 2 and 4) for CTP1–CTP4.

7.2. Experimental results on CF-series test instances

Table 3 presents the best (i.e., lowest), mean, and standard deviation of the IGD-metric values for CF-series test instances found by CMOEA/D-DE-SR with $p_f=0.05$ and CMOEA/D-DE-CDP. These statistics are based on 30 independent runs. As it can be seen from this table that CMOEA/D-DE-SR found better results for two 2-objective test instances CF2 and CF5 and two 3-objective test instances CF8 and CF9. CMOEA/D-DE-CDP, on the other hand, found better statistics for five 2-objective test instances CF1, CF3, CF4, CF6

and CF7 except the standard deviation value for test instance CF7 and one 3-objective test instance CF10.

It can also be observed from Table 3 that both algorithms found small values for the mean of IGD-metric on test instances CF1, CF2, CF4, CF6, CF8, CF9. These results empirically demonstrate that the final nondominated solutions found by both algorithms for these test instances approximate the PF very well in a sense. Moreover, because of the small differences in the IGD-metric statistics for these test instances, the obtained results by both algorithms may be considered as comparable.

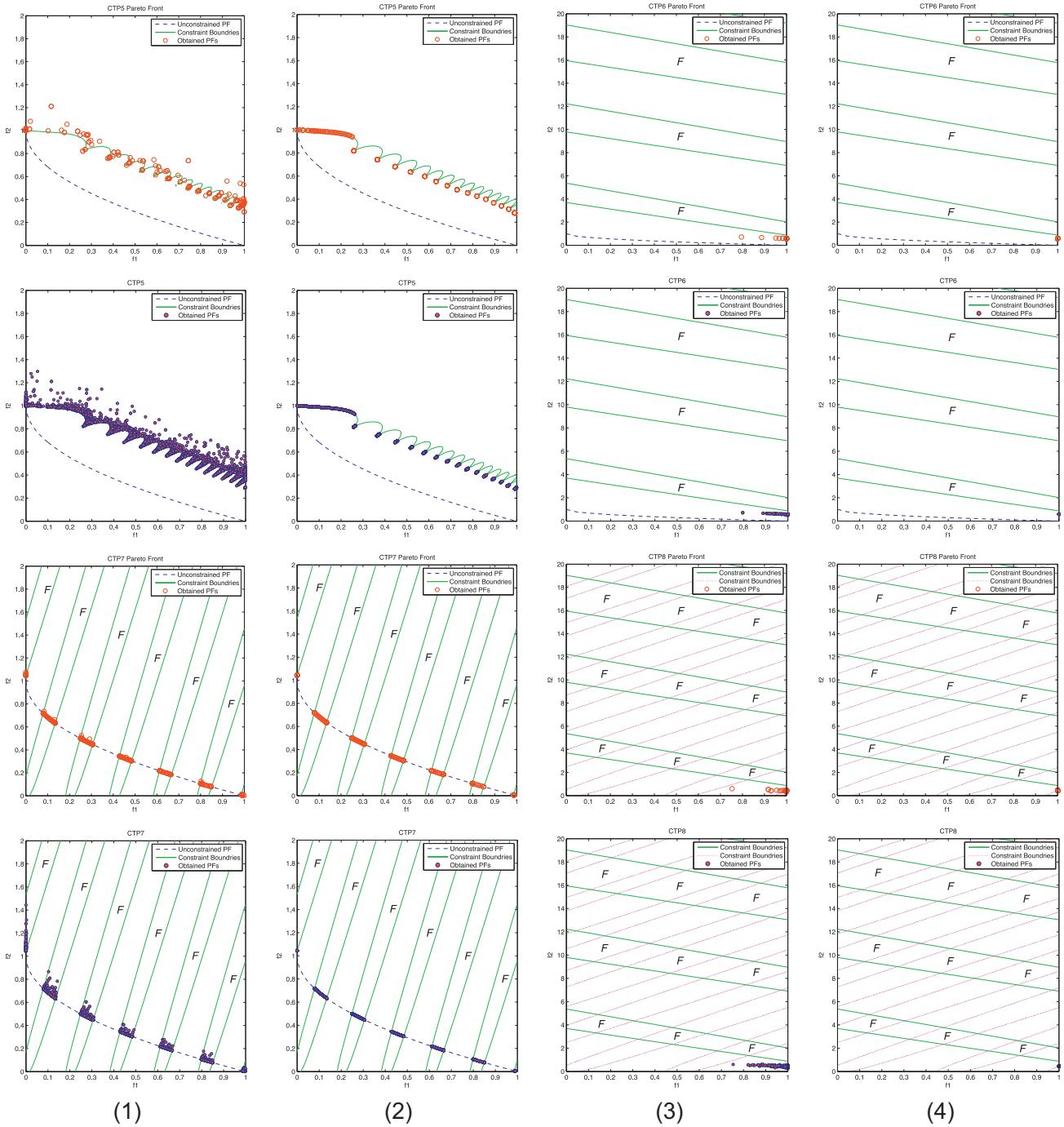


Fig. 2. Plots of the nondominated front with the best HV-metric value and all the 30 final nondominated fronts found by CMOEA/D-DE-SR (columns 1 and 3) and CMOEA/D-CDP (columns 2 and 4) for CTP5–CTP8.

Table 4 presents the average set coverage between the nondominated solutions of CMOEA/D-DE-SR with $p_f=0.05$ and CMOEA/D-DE-CDP for CF-series test instances. The results of this table unveil that, in terms of the SC-metric, the nondominated solutions found by CMOEA/D-DE-CDP are better than those obtained by CMOEA/D-DE-SR for six test instances CF1, CF3–CF5, CF8, and CF10, while the situation is vice versa for two test instances CF2 and CF7. This table also shows that the nondominated solutions found by both algorithms for test instances CF6 and CF9 are same. However, looking at the results of this table, it can be concluded that the performance of both algorithms is similar in terms of the SC-metric

for all CF-series test instances except CF1–CF3, since there is no big difference in the SC-metric values.

Figs. 5–7 show, in the function space, the distributions of the 100 and 150 feasible nondominated population members for the seven 2-objective, CF1–CF7, and three 3-objective, CF8–CF10, CF-series test instances. These solutions are selected based on the criteria of Section 6.2.1 from the final population of the run with the best (i.e., lowest) IGD-metric value among the 30 independent runs. These figures also show all the 30 final non-dominated fronts of these selected 100 and 150 nondominated solutions.

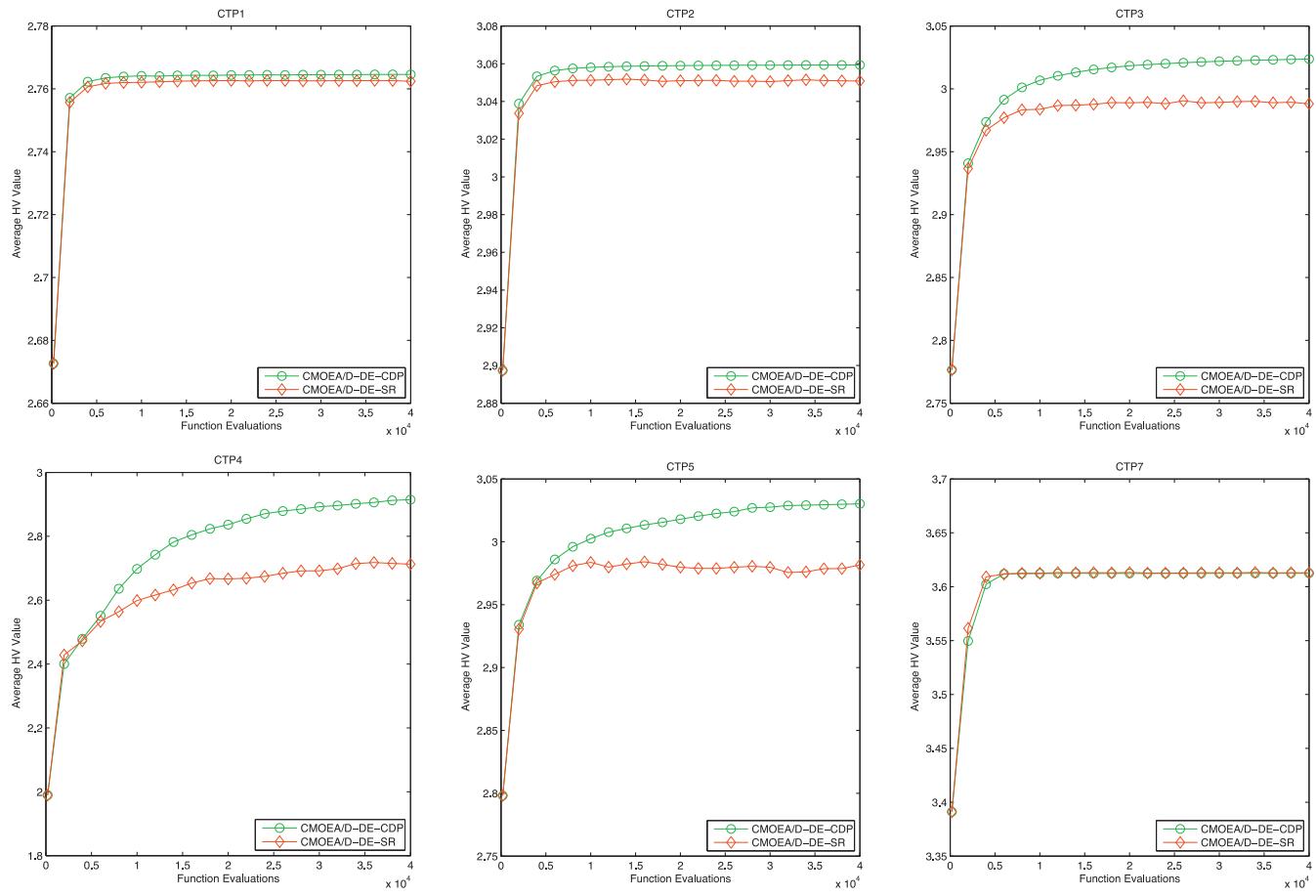


Fig. 3. Evolution of the mean HV-metric values versus function evaluations in CMOEA/D-DE-SR with $p_f = 0.05$ and CMOEA/D-DE-CDP for CTP1–CTP5 and CTP7.

From these figures, it is very clear that both algorithms found good approximations for the four 2-objective test instances, CF1, CF2, CF4, and CF6, and two 3-objective test instances CF8, CF9. However, they performed comparatively poorly on test instances CF3, CF5, CF7, and CF10. In these test instances, the nondominated solutions converge to the PFs, but they are not uniformly distributed along the entire PF. It is also obvious from the plots of 30 nondominated fronts of test instances CF4 and CF6 that both algorithms fail to find the whole PF in some runs.

The PF of CF3 is concave and discontinuous. Therefore, it could be harder than all other 2-objective test instances for both algorithms. Although the PFs of CF4 and CF5, CF6 and CF7, and CF9 and CF10 are identical, the poor performance of CMOEA/D-DE-SR

and CMOEA/D-DE-CDP on test instances CF5, CF7 and CF10 could be due to the presence of harder objective and constraint functions in these test instances than test instances CF4, CF6, and CF9.

Fig. 8 presents the evolution of the average IGD-metric values versus function evaluations of the nondominated solutions in the current population. This figure shows that CMOEA-DE-CDP converges faster in terms of IGD-metric values than CMOEA-DE-SR for five test instances CF1, CF3, CF6, CF7, and CF10, while, the case is vice versa for two test instances CF2 and CF5. However, both algorithms converge almost at the same rate for three test instances CF4, CF8, and CF9.

Fig. 9 shows average generation feasibility versus generations' graphs. This figure shows that both CMOEA/D-DE-SR and CMOEA/D-DE-CDP converged to the feasible regions quickly and at the same rate for seven CF-series test instances, CF4–CF10. However, in case of test instances CF1–CF3, the generation feasibility remains lower in the former than the latter. In test instance CF1, the feasibility ratio in CMOEA-DE-SR initially increases, then starts decreasing and finally ends up at the last generation with a value 0.65. In test instances CF2 and CF3, it oscillates in the ranges [0.997, 0.9999], [0.9995, 1], respectively.

Fig. 9 also demonstrates that about 50% or more and about 25% or below of the initial populations are feasible for test instances CF1–CF5 and CF6–CF10, respectively. These feasible solutions are quickly propagated in the subsequent generations by the replacement and update scheme of the algorithms. For example, for test instances CF4–CF10, the feasibility ratio becomes 1 mostly by generation 10. The reason for the quick convergence to feasible region

Table 4

The average set coverage between CMOEA/D-DE-SR (indicated by SR) and CMOEA/D-DE-CDP (indicated by CDP) on CF-series test instances. The results in boldface indicate the better results; if not, they are identical.

Test instance	C(CDP, SR)	C(SR, CDP)
CF1	0.64	0.29
CF2	0.16	0.23
CF3	0.40	0.31
CF4	0.30	0.26
CF5	0.22	0.19
CF6	0.18	0.18
CF7	0.20	0.24
CF8	0.07	0.06
CF9	0.03	0.03
CF10	0.25	0.23

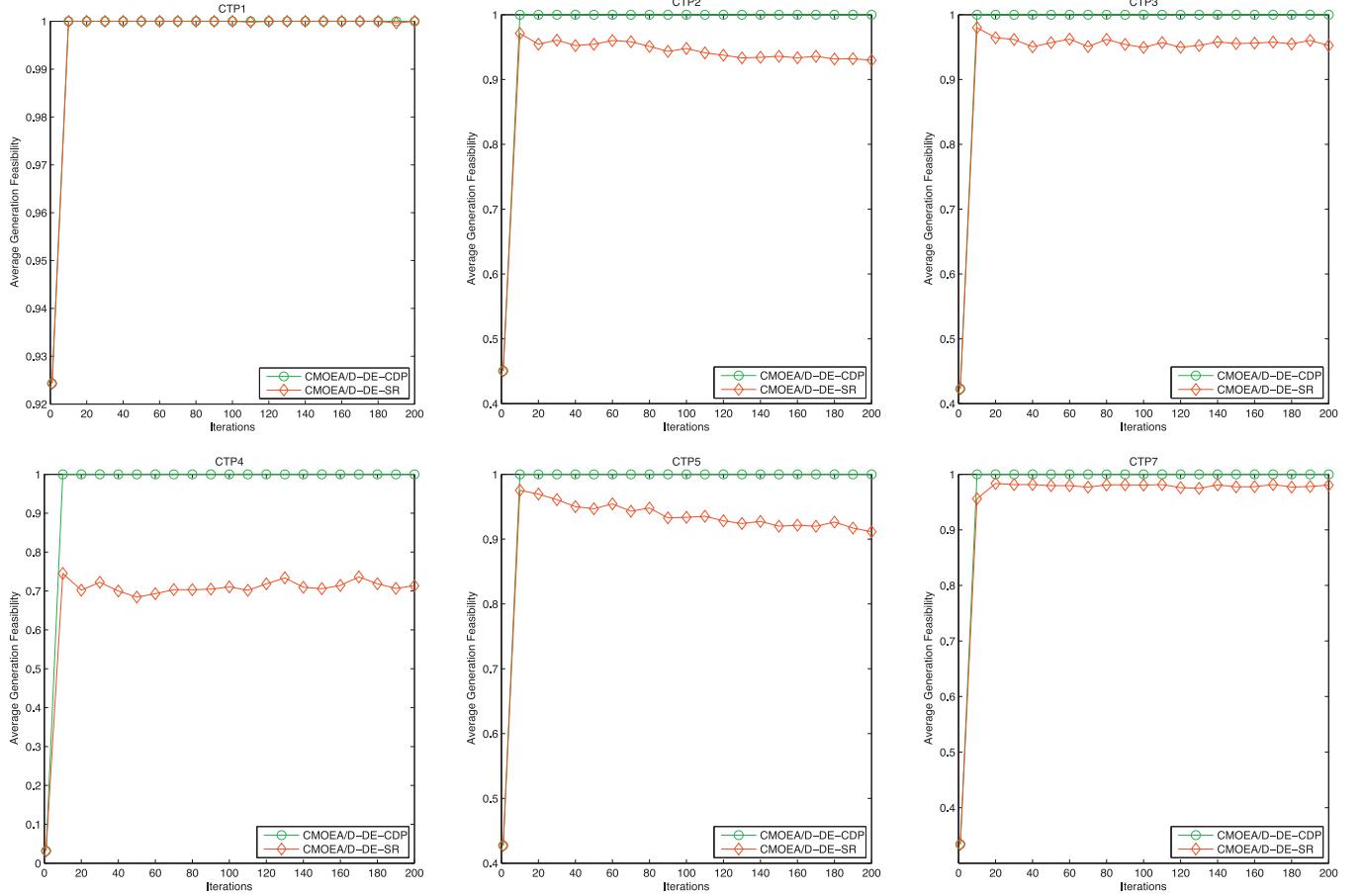


Fig. 4. Evolution of the mean generation feasibility versus generations in CMOEA/D-DE-SR with $p_f = 0.05$ and CMOEA/D-DE-CDP for CTP1–CTP5 and CTP7.

is the higher adopted update number of neighboring parent solutions (as in our settings $n_r = 6$ when $T = 60$ and $n_r = 10$ when $T = 100$) that are replaced by a better child solution in the update scheme of the algorithms. This quick convergence facilitates more exploration of the feasible region by CMOEA/D-DE, and is good for test instances like CF4, CF6, CF8 and CF9. However, it is causing problems for hard test instance like CF5, CF7, and CF10. Moreover, as shown above, CMOEA/D-DE-SR can retain some infeasible solutions during the evolution for test instances CF1–CF3, which are beneficial in case of test instance CF2, but not in case of other two test instances.

From the experimental results on CTP-series and CF-series test instances, it could be concluded that comparing infeasible solutions based on constraint violation only, as is done in CMOEA/D-DE-CDP, or a small percentage based on aggregation function values and the rests based on constraint violation, as is done in CMOEA/D-DE-SR, leads to better performance for some test instances. However, the former strategy works comparatively better than the latter one in the framework CMOEA/D-DE.

8. Comparison with NSGA-II, IDEA, and the three best performers of CEC 2009 MOEA competition

In this section, we compare the experimental results of CMOEA/D-DE-SR and CMOEA/D-DE-CDP with NSGA-II [9] and IDEA [19] results on seven CTP-series, CTP2–CTP8 [2,17] and with the three best performers [20–22] of CEC 2009 MOEA competition on CF-series [18] test instances.

Table 5 compares the HV-metric statistics (across all 30 runs) obtained from our algorithms, CMOEA/D-DE-SR with $p_f = 0.05$ and CMOEA/D-DE-CDP, NSGA-II with the constraint domination principle [9] and IDEA with $\alpha = 0.2$ [19] (α determines the percentage of infeasible solutions to be retained during evolution) for seven CTP-series test instances, CTP2–CTP8. The statistics for the algorithms NSGA-II and IDEA are taken from [19] and have been rounded to four decimal places. The genetic operators SBX with distribution index value of 15 for generating offspring and polynomial mutation with distribution index value of 20 are used in NSGA-II and IDEA. All other parameters remain the same as in Section 5.1 in our experiments.

From this table, it can be found that both of our algorithms have found substantially better mean and standard deviation values than NSGA-II and IDEA for five test instances, CTP2–CTP5 and CTP7. In particular, the standard deviation values obtained with CMOEA/D-DE-CDP are notably better than all the three algorithms. The small values of the standard deviation for CMOEA/D-DE-SR and CMOEA/D-DE-CDP suggest that the performance of our both algorithms is persistent. The table also shows that CMOEA/D-DE-CDP found the best HV-metric values for four test instances, CTP2–CTP5, while NSGA-II and IDEA found it for test instances CTP6 and CTP8, respectively. Moreover, the best HV-metric values found by NSGA-II and IDEA for test instance CTP7 are same.

Because both of our algorithms get trapped in the infeasible regions and converged to a single infeasible solution or more than one infeasible solutions for test instances CTP6 and CTP8, we do not present any statistics for these two test instances. However,

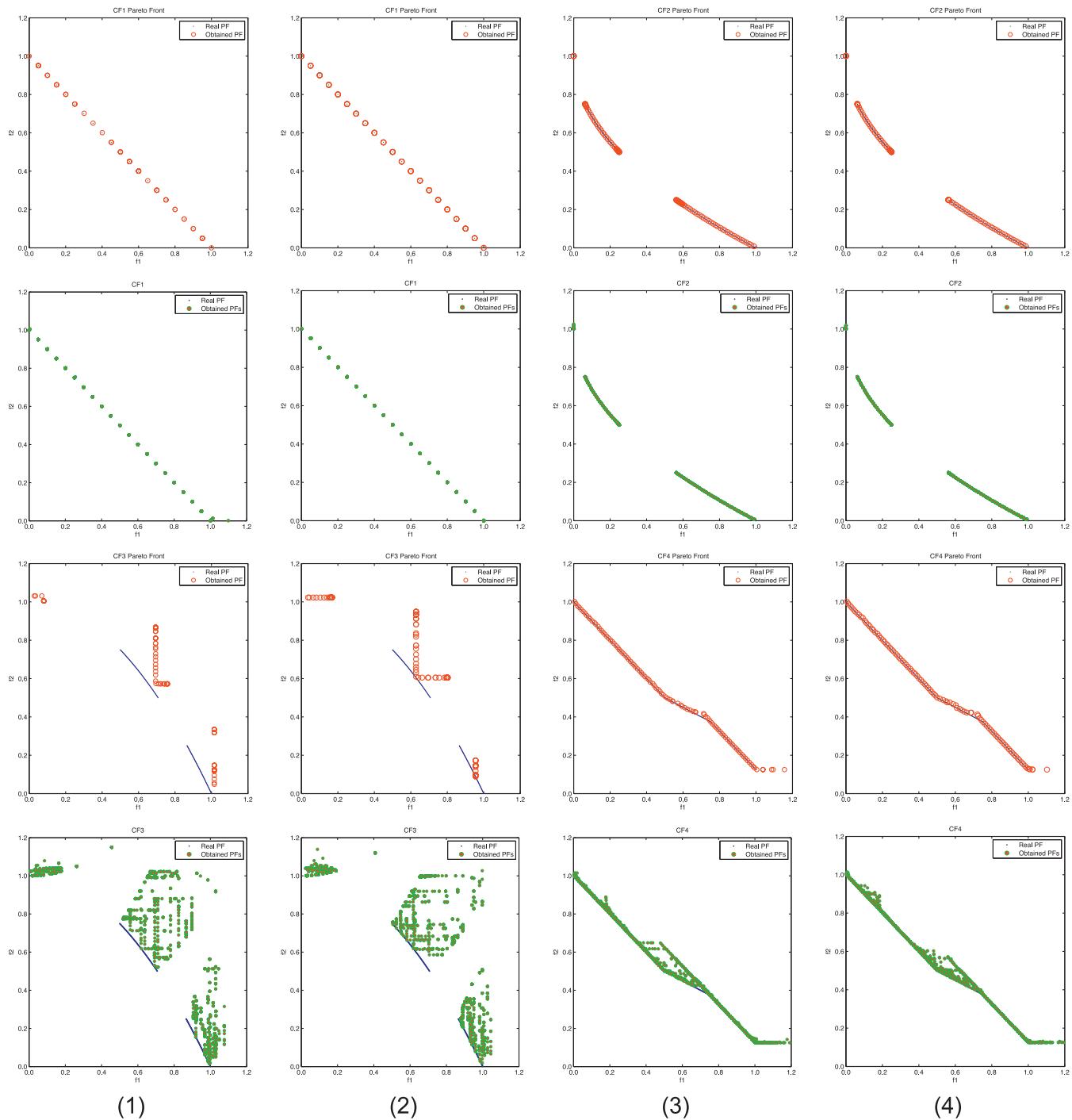


Fig. 5. Plots of the nondominated front with the best IGD-metric value and all the 30 final nondominated fronts found by CMOEA/D-DE-SR (columns 1 and 3) and CMOEA/D-DE-CDP (columns 2 and 4) for CF1–CF4.

Table 5

Comparison between our algorithms, CMOEA/D-DE-SR (indicated by SR) and CMOEA/D-DE-CDP (indicated by CDP), IDEA and NSGA-II in terms of the HV-metric statistics based on 30 independent runs. The results in boldface and in italic indicate the better and the second better results. ‘–’ means that no feasible solution is found.

Test instance	Best (highest)				Mean				St. dev.			
	SR	CDP	IDEA	NSGA-II	SR	CDP	IDEA	NSGA-II	SR	CDP	IDEA	NSGA-II
CTP2	3.0528	3.0595	3.0592	3.0593	3.0508	3.0594	3.0114	2.8707	0.0014	0.0001	0.1771	0.2701
CTP3	3.0008	3.0282	3.0160	3.0104	2.9881	3.0238	2.9608	2.8281	0.0059	0.0024	0.1638	0.2547
CTP4	2.8266	2.9761	2.9190	2.8485	2.7129	2.9147	2.7447	2.4381	0.0457	0.0342	0.1393	0.3527
CTP5	3.0064	3.0382	3.0247	3.0209	2.9816	3.0302	2.9529	2.7235	0.0167	0.0062	0.1621	0.2926
CTP6	–	–	36.8191	36.8227	–	–	36.7878	36.1829	–	–	0.0758	2.1873
CTP7	3.6162	3.6125	3.6177	3.6177	3.6128	3.6124	3.4359	3.2402	0.0011	0.0000	0.5945	0.5941
CTP8	–	–	36.1804	36.1708	–	–	35.9706	32.0859	–	–	0.4345	5.1763

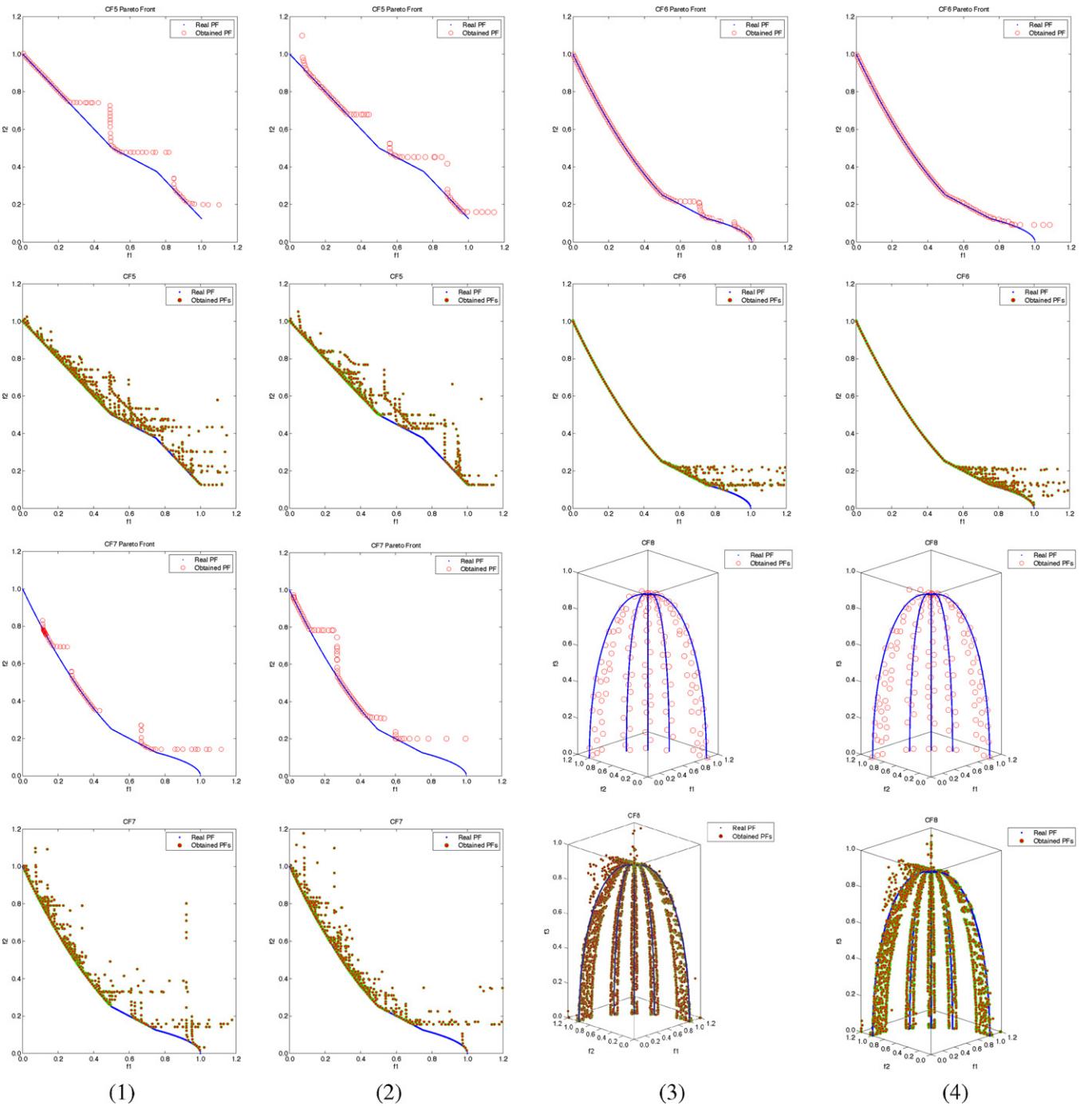


Fig. 6. Plots of the nondominated front with the best IGD-metric value and all the 30 final nondominated fronts found by CMOEA/D-DE-SR (columns 1 and 3) and CMOEA/D-DE-CDP (columns 2 and 4) for CF5–CF8.

for both of them, IDEA found better mean and standard deviation HV-metric values than NSGA-II.

Table 6 compares the best (i.e., lowest), mean, and standard deviation values of the IGD-metric obtained from our algorithms, CMOEA/D-DE-SR with $p_f=0.05$ and CMOEA/D-DE-CDP and the three best performers [20–22] of CEC 2009 MOEA competition for the CF-series test instances. Clearly, CMOEA/D-DE-SR has found the better statistics for two 3-objective test instances CF8 and CF9 except the best value for test instance CF8. It also obtained the second best results for test instances CF1, CF2, CF4, and CF8. CMOEA/D-DE-CDP, on the other hand, found the best results for test instances CF1 and CF4 and the second best results for test instances

CF3, CF6 and CF9. In particular, for test instances CF1, CF8, and CF9, better statistics are found by our algorithms except the best value on CF8 and standard deviation value on CF1 (It can be observed that our best and standard deviation values are very close to better results on these two test instances.).

9. Population evolution and sensitivity to parameters values

In this section, first we discuss the evolution of the population members across 200 generations of CMOEA/D-DE-SR and CMOEA/D-DE-CDP for test instance CTP2. We then analyze the

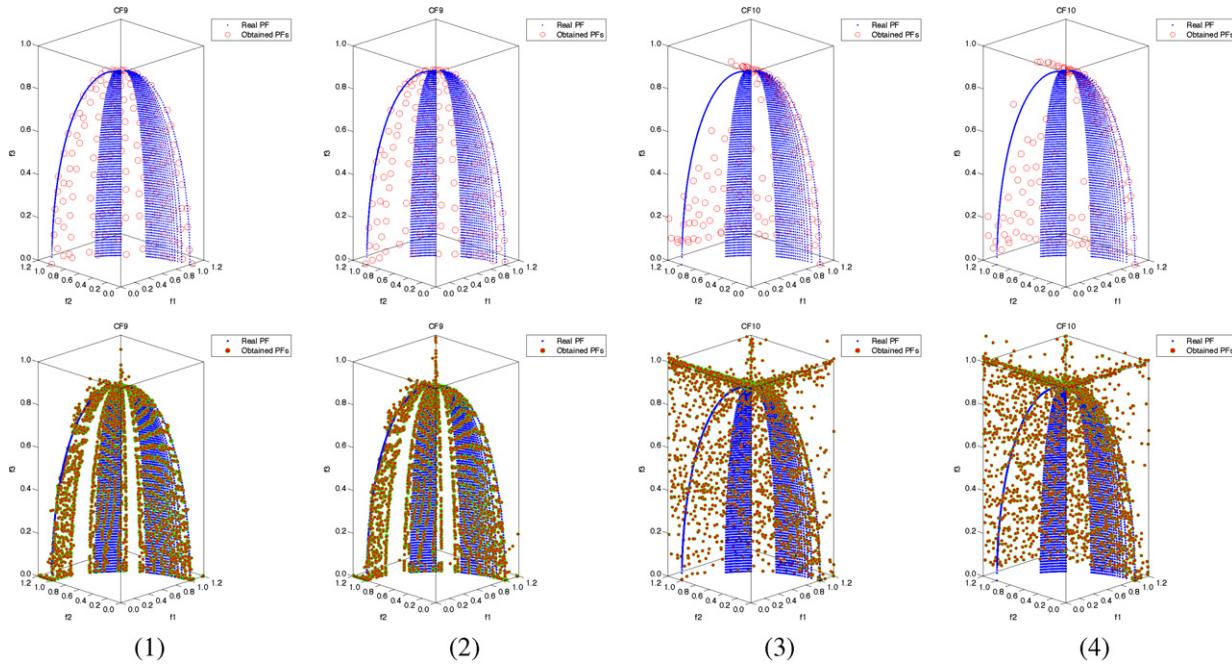


Fig. 7. Plots of the nondominated front with the best IGD-metric value and all the 30 final nondominated fronts found by CMOEA/D-DE-SR (columns 1 and 3) and CMOEA/D-DE-CDP (columns 2 and 4) for CF9 and CF10.

sensitivity of the performance of both algorithms to parameters.

9.1. Population evolution in CMOEA/D-DE-SR and CMOEA/D-DE-CDP

Figs. 10 and 11 show the progress of CMOEA/D-DE-SR and CMOEA/D-DE-CDP populations up to 200 generations for test instance CTP2. These figures demonstrate that how in case of test instance CTP2 both algorithms first converge the initial population to different feasible regions near the PF in a few early generations and then reach the disconnected continuous parts of the PF at the end of each one of them. However, CMOEA/D-DE-CDP very evenly distributes the Pareto-optimal solutions along the different parts of the PF than CMOEA/D-DE-SR. In the latter case, a few infeasible solutions in the final population can be observed from Fig. 10.

9.2. Sensitivity of CMOEA/D-DE-SR to p_f

In order to study the sensitivity of the performance of CMOEA/D-DE-SR to the probability of comparing solutions based on the aggregation function values, p_f , we have tried 9 values of p_f (i.e., $p_f = 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.40, 0.45, 0.50$) in it for test

instances CTP2, CTP3, and CTP4. All other parameters remain the same as in Section 5.1 of our experiments.

Fig. 12 shows the average HV-metric values versus the different values of p_f . It is clear from this figure that CMOEA/D-DE-SR with $p_f = 0.05$ has the best performance in terms of the HV-metric for all three test instances CTP2, CTP3, and CTP4. However, as this probability increases, the HV-metric values get smaller and smaller for all three test instances. This suggests that a comparison based on aggregation function values with small probability is beneficial for the better performance, in terms of the HV-metric values, of CMOEA/D-DE-SR on the listed three test instances.

9.3. Sensitivity of CMOEA/D-DE-SR and CMOEA/D-DE-CDP to T

To study the sensitivity of the performances of CMOEA/D-DE-SR and CMOEA/D-DE-CDP to the neighborhood size, T , we have examined 10 values of neighborhood size (i.e., $T = 5, 10, 20, 30, 40, 50, 100, 150, 200$) in both algorithms for test instance CTP2. All other parameters remain the same as in Section 5.1 in our experiments.

Fig. 13 shows the average HV-metric values versus the neighborhood size T . It is clear from this figure that the performance of both algorithms is sensitive to the setting of the neighborhood size T .

Table 6
Comparison between our's (indicated by SR and CDP), Tseng and Chen's [20] (indicated by TC), Liu and Li's [21] (indicated by LL), and Liu et al.'s [22] (indicated by LI) algorithms in terms of the IGD-metric values based on 30 independent runs. The results in boldface and in italic indicate the better and the second better results.

Test instance	Best (lowest)					Mean					St. dev.				
	SR	CDP	TC	LL	LI	SR	CDP	TC	LL	LI	SR	CDP	TC	LL	LI
CF1	0.0005	0.0002	0.0139	0.0007	0.0071	0.0015	0.0006	0.0192	0.0009	0.0113	0.0016	0.0002	0.0026	0.0001	0.0028
CF2	0.0024	0.0026	0.0041	0.0027	0.0016	0.0033	0.0040	0.0268	0.0042	0.0021	0.0016	0.0018	0.0147	0.0026	0.0005
CF3	0.0767	0.0632	0.0753	0.0908	0.0381	0.1437	0.1382	0.1045	0.1829	0.0563	0.0448	0.0441	0.0156	0.0421	0.0076
CF4	0.0054	0.0051	0.0089	0.0090	0.0055	0.0103	0.0092	0.0111	0.0142	0.0070	0.0046	0.0030	0.0014	0.0033	0.0015
CF5	0.0343	0.0353	0.0176	0.0588	0.0079	0.1388	0.1843	0.0208	0.1097	0.0158	0.1109	0.1278	0.0024	0.0307	0.0067
CF6	0.0091	0.0074	0.0096	0.0090	0.0062	0.0292	0.0261	0.0162	0.0139	0.0150	0.0199	0.0166	0.0060	0.0026	0.0065
CF7	0.0494	0.0381	0.0187	0.0535	0.0104	0.1894	0.1718	0.0247	0.1045	0.0191	0.0917	0.0957	0.0047	0.0351	0.0061
CF8	0.0397	0.0404	0.6220	0.0473	0.0388	0.0435	0.0441	1.0854	0.0607	0.0475	0.0024	0.0028	0.2191	0.0130	0.0064
CF9	0.0447	0.0449	0.0721	0.0460	0.1191	0.0489	0.0497	0.0851	0.0505	0.1434	0.0029	0.0030	0.0082	0.0034	0.0214
CF10	0.1169	0.1123	0.1173	0.1055	0.0984	0.2911	0.2563	0.1376	0.1974	0.1621	0.1651	0.1453	0.0092	0.0760	0.0316

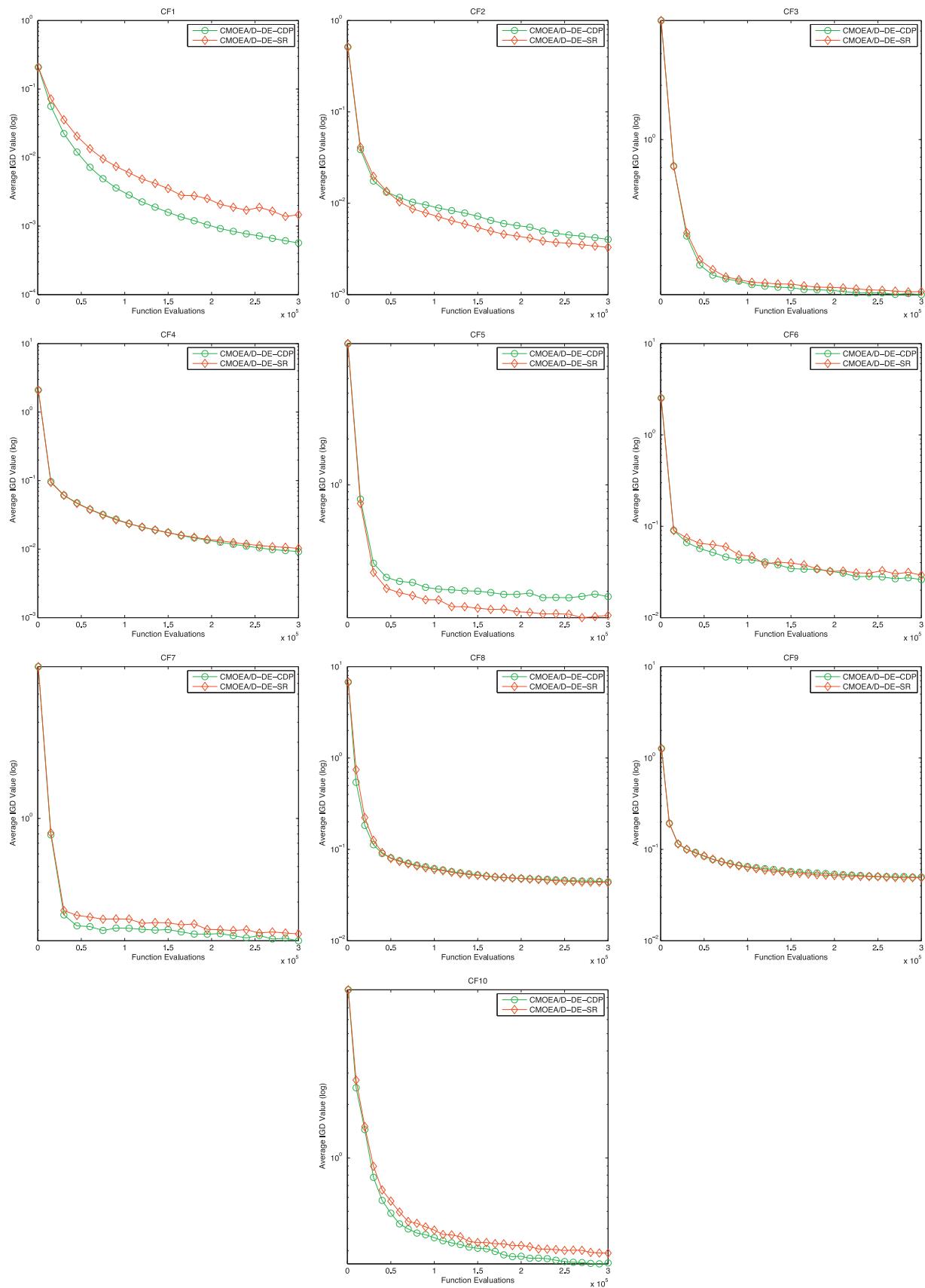


Fig. 8. Evolution of the IGD-metric values versus function evaluations in CMOEA/D-DE-SR and CMOEA/D-DE-CDP for CF1–CF10.

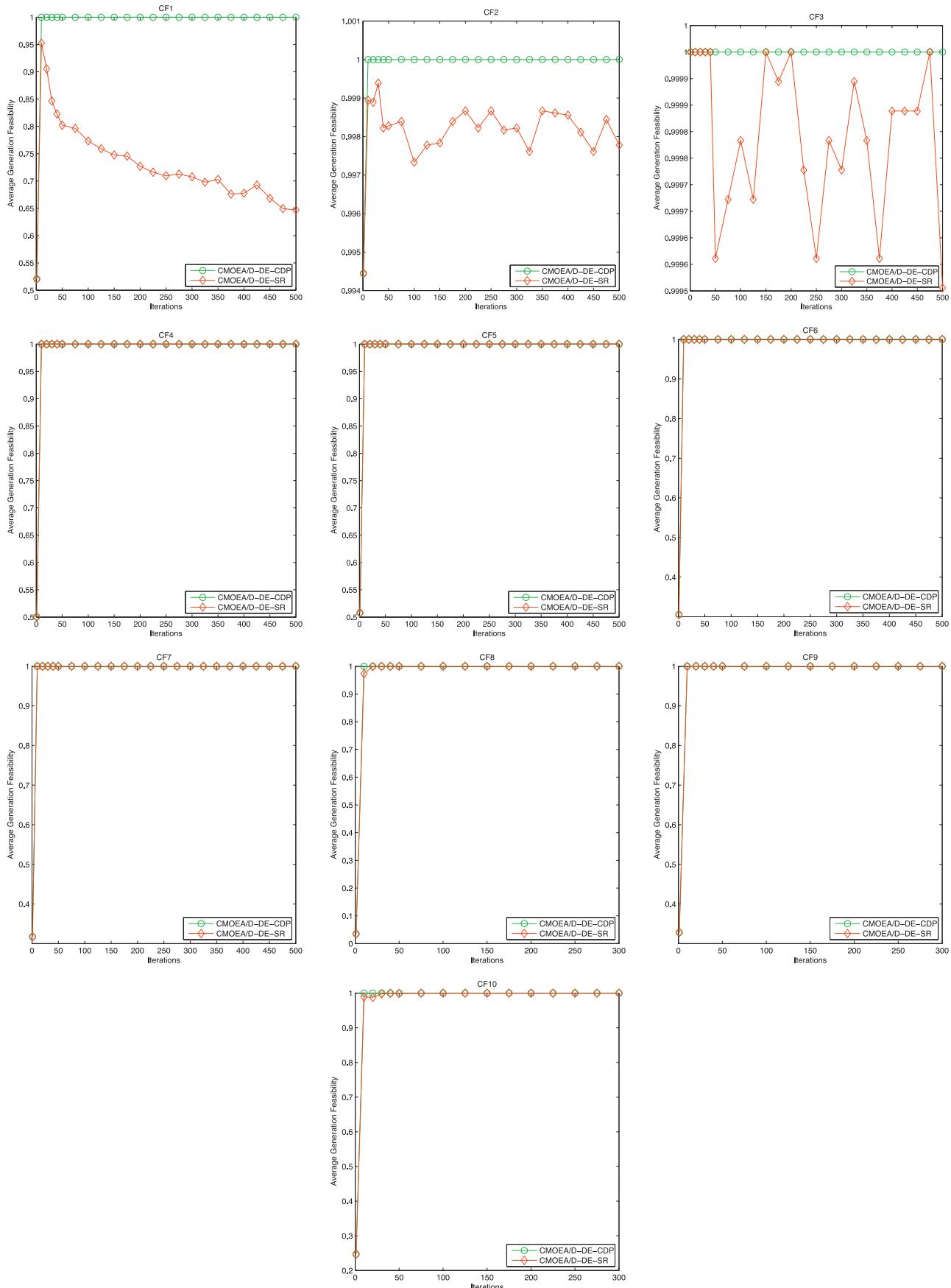


Fig. 9. Evolution of the generation feasibility versus generations in CMOEA/D-DE-SR and CMOEA/D-DE-CDP for CF1–CF10.

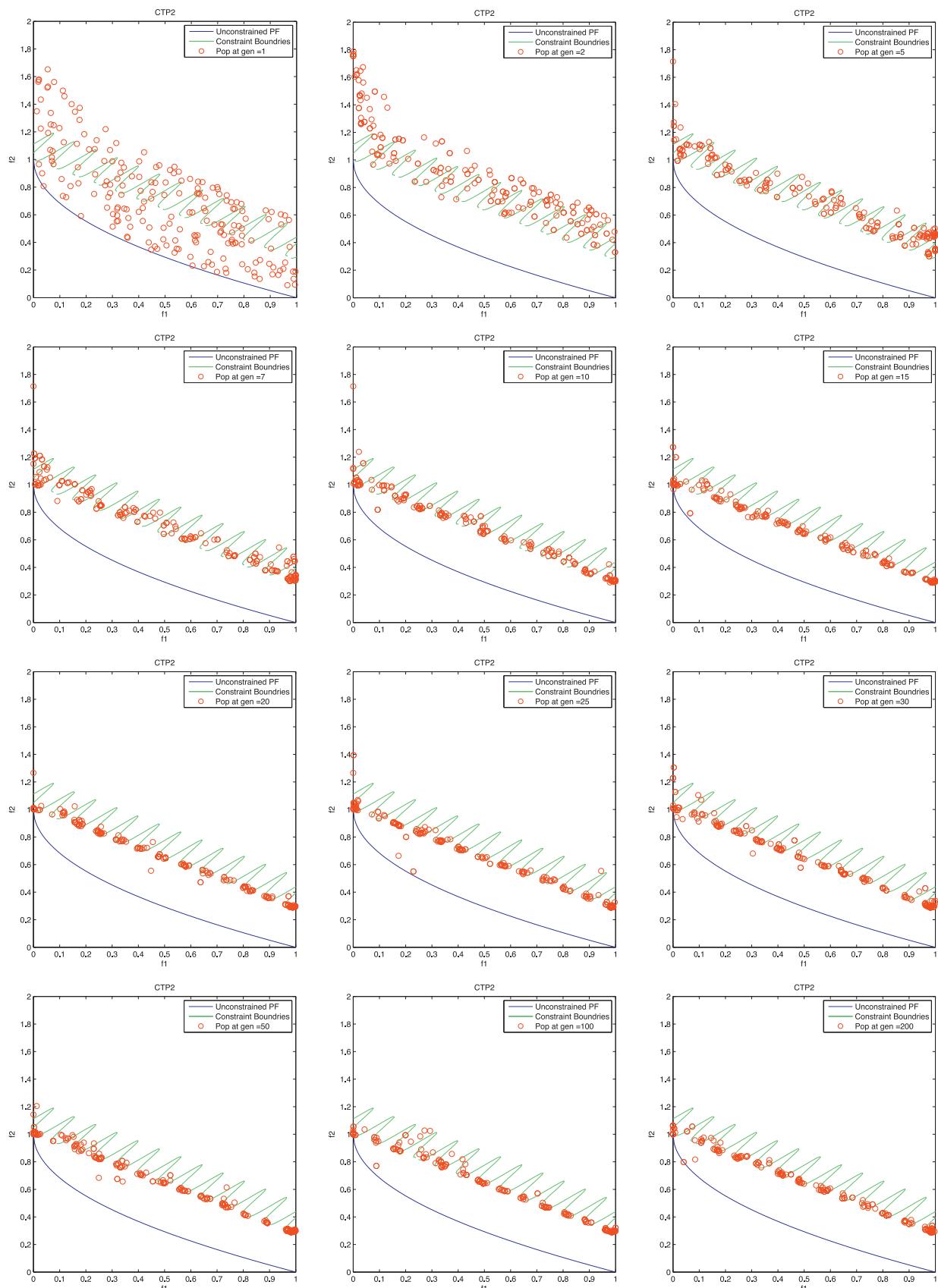


Fig. 10. Evolution of CMOEA/D-DE-SR population over generations for CTP2 test run.

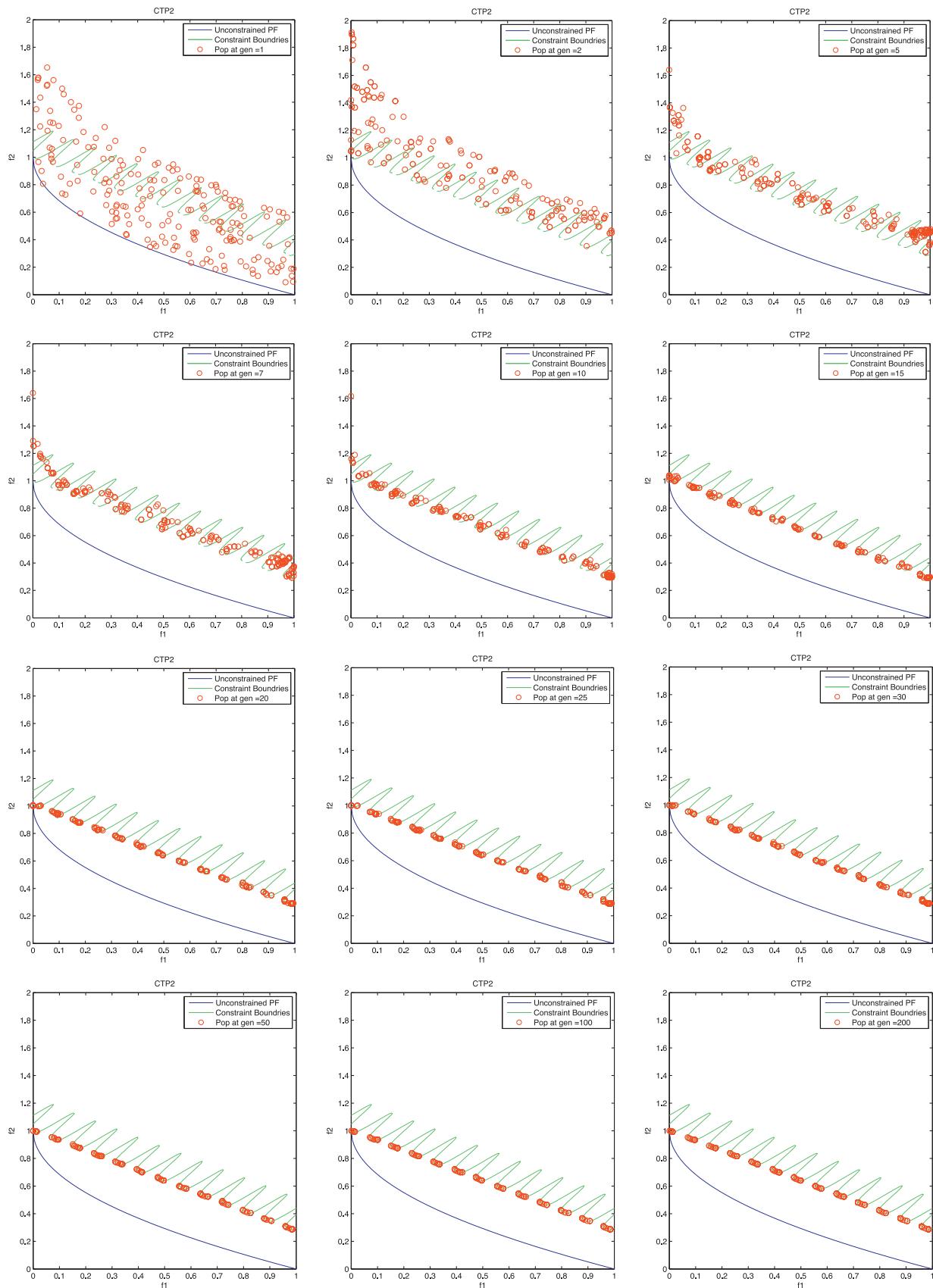


Fig. 11. Evolution of CMOEA/D-DE-CDP population over generations for CTP2 test run.

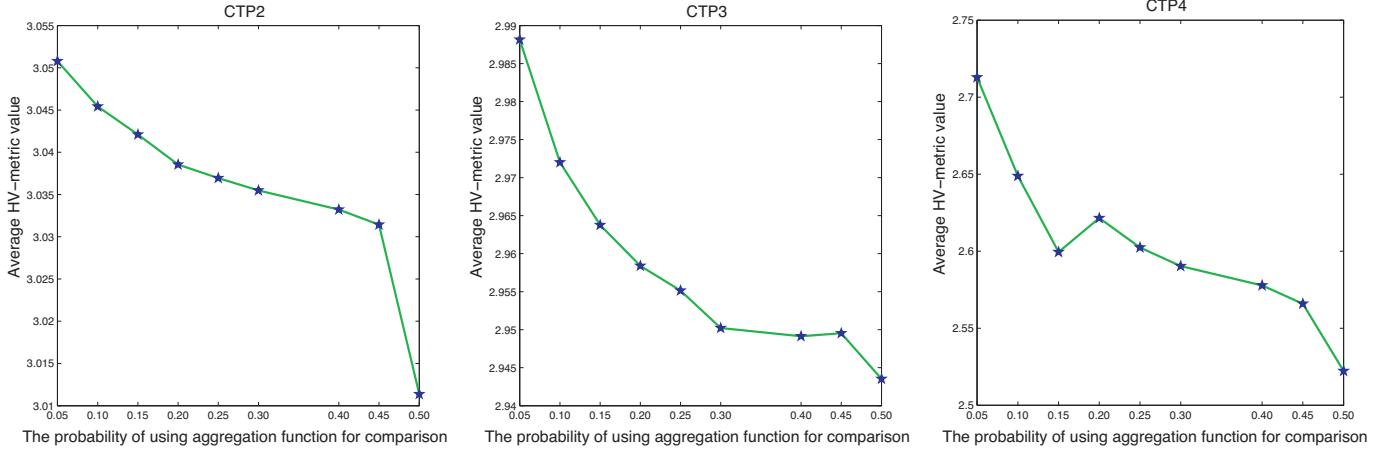


Fig. 12. The average HV-metric values versus the probability of comparing solutions based on the aggregation function values (p_f) found by CMOEA/D-DE-SR in CTP2, CTP3, and CTP4.

Fig. 13(a) shows that CMOEA/D-DE-SR found the best (i.e., highest) average value of the HV-metric with $T=5$. However, as the value of T increases, the approximation of the PF of CTP2 by CMOEA/D-DE-SR gets worsen as compared to that of $T=5$.

Fig. 13(b) shows that, for CTP2, the best (i.e., highest) average value of the HV-metric is found by CMOEA/D-DE-CDP with $T=50$. We can also see from this figure that CMOEA/D-DE-CDP can find the average HV-metric values in the range $30 < T < 50$ very close to the one obtained with $T=50$. This points out that CMOEA/D-DE-CDP can still find the good approximation of the PF of CTP2 when the neighborhood size varies in an appropriate range.

9.4. Sensitivity of CMOEA/D-DE-SR and CMOEA/D-DE-CDP to n_r

To study the sensitivity of the performances of CMOEA/D-DE-SR and CMOEA/D-DE-CDP to the update number, n_r , we have tested 10 values of n_r (i.e., $n_r = 1, 2, 3, 4, 5, 6, 7, 10, 15, 20$) in both algorithms for test instance CTP2. All other parameters remain the same as in Section 5.1 in our experiments.

Fig. 14 presents the average HV-metric values versus the ten different values of n_r . This figure clearly shows that both algorithms with small $n_r = 1, 2$ values can find good approximation of the PF of

CTP2. However, when n_r is large, both algorithms perform worse in maximizing the HV-metric values.

9.5. Sensitivity of CMOEA/D-DE-SR and CMOEA/D-DE-CDP to δ

In order to study the sensitivity of the performances of CMOEA/D-DE-SR and CMOEA/D-DE-CDP to the probability of selecting mating parents from the neighborhood, δ , we have tried 10 values of δ (i.e., $\delta = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0$) in both algorithms for test instance CTP2. All other parameters remain the same as in Section 5.1 of our experiments.

Fig. 15 shows the average HV-metric values versus the values of δ . It is clear from this figure that both algorithms with $\delta = 1.0$ has the best performance in terms of the HV-metric for test instance CTP2. This indicates that both algorithms perform well when 100% solutions are selected from the neighborhood. However, as it can be seen from this figure, the HV-metric value obtained with $\delta = 0.9$ in both algorithms is very close to the one acquired with $\delta = 1.0$. Thus, if 90% solutions from the neighborhood of a solution and 10% from the whole population for recombination are selected, the performance of both algorithms in terms of the HV-metric values can still be improved on CTP2. In the latter case, the solutions that are apart from each other in the search space will get a chance to

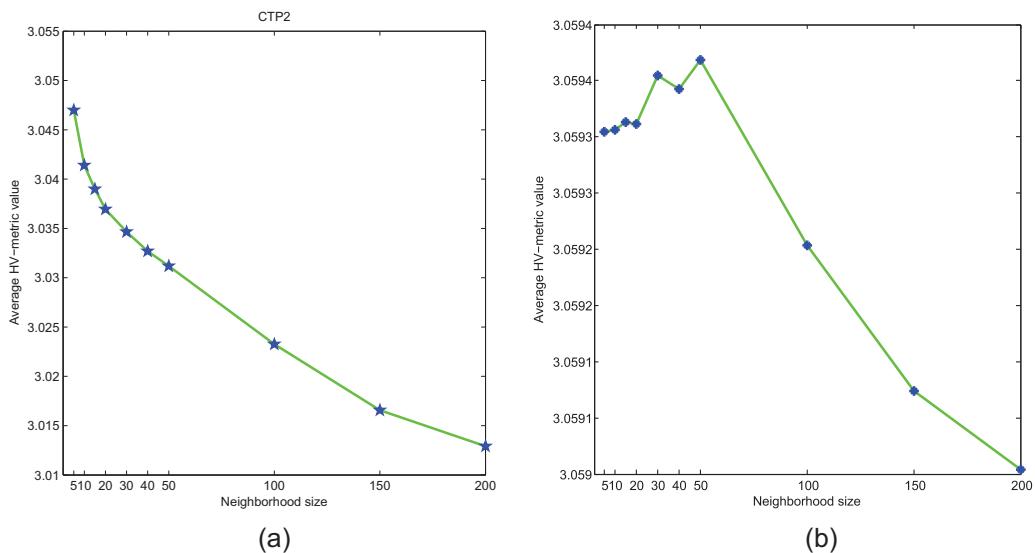


Fig. 13. The average HV-metric values versus the neighborhood size (T) found by CMOEA/D-DE-SR (a) and CMOEA/D-DE-CDP (b) in CTP2.

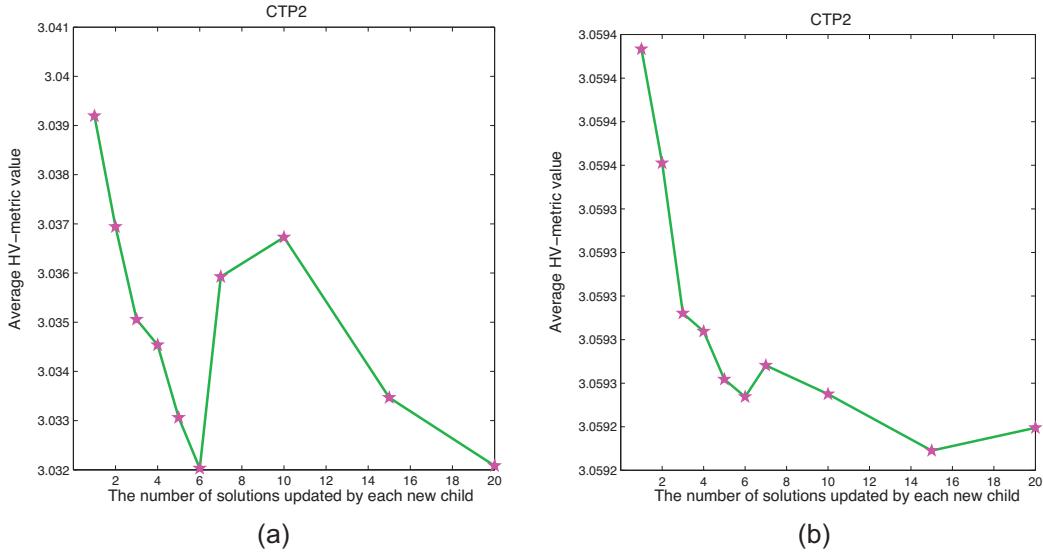


Fig. 14. The average HV-metric values versus the maximal number of solutions updated by each new offspring (n_r) found by CMOEA/D-DE-SR (a) and CMOEA/D-DE-CDP (b) in CTP2.

mate with a low probability. This strategy increases the ability of both algorithms to explore the search space. Nonetheless, more computational efforts are spent on the recombination between neighboring solutions in order to exploit the PS efficiently.

9.6. Sensitivity of CMOEA/D-DE-SR and CMOEA/D-DE-CDP to CR and F

The genetic operator DE employs two control parameters CR and F for generating offspring. To study the effects of these two parameters on the behaviors of CMOEA/D-DE-SR and CMOEA/D-DE-CDP, we have tested in both algorithms 50 combinations of five values of F (i.e., $F=0.1, 0.25, 0.5, 0.75, 1.0$) and ten values of CR (i.e., $CR=0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0$) on test instance CTP2. Each combination of CR and F is tested 30 times. All other parameters remain the same as in Section 5.1 in our experiments.

Fig. 16 shows the average HV-metric values with 50 different combinations of CR and F values. It is evident from this figure

that both CMOEA/D-DE-SR and CMOEA/D-DE-CDP are less sensitive to the settings of CR and F under the considered ranges and test instance.

9.7. CMOEA/D-DE-SR and CMOEA/D-DE-CDP with small population

To study the behaviors of CMOEA/D-DE-SR and CMOEA/D-DE-CDP with a small population, we run them with population size of $N=50$ and 200 generations on test instance CTP2. Accordingly, the algorithm terminates after 10,000 function evaluations. All other parameters remain the same as in Section 5.1 in our experiments.

Fig. 17 plots the final solutions obtained in a single run of CMOEA/D-DE-SR and CMOEA/D-DE-CDP with $N=50$ for test instance CTP2. Although both algorithms found Pareto-optimal solutions in the disconnected parts of the PF, CMOEA/D-DE-CDP found 50 very evenly distributed solutions than CMOEA/D-DE-SR for CTP2. This is a matter of interest for *decision makers* (DMs) to

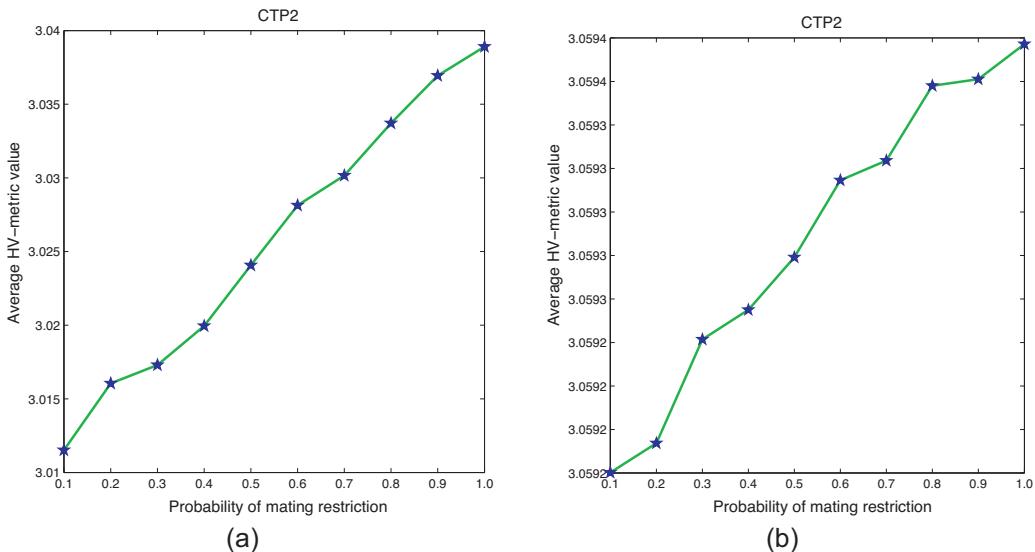


Fig. 15. The average HV-metric values versus the probability of selecting mating parents from the neighborhood (δ) found by CMOEA/D-DE-SR (a) and CMOEA/D-DE-CDP (b) in CTP2.

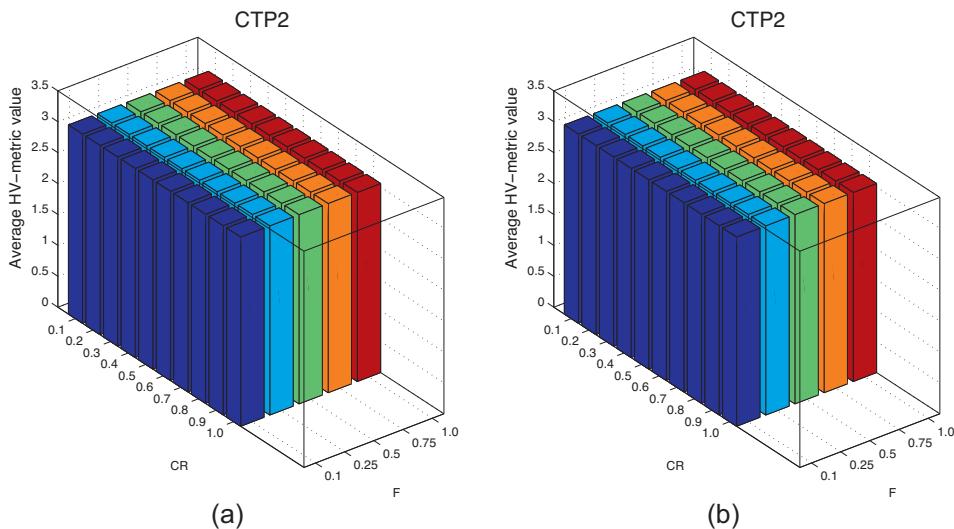


Fig. 16. The average HV-metric values obtained by CMOEA/D-DE-SR (a) and CMOEA/D-DE-CDP (b) with 50 different combinations of CR and F values on CTP2.

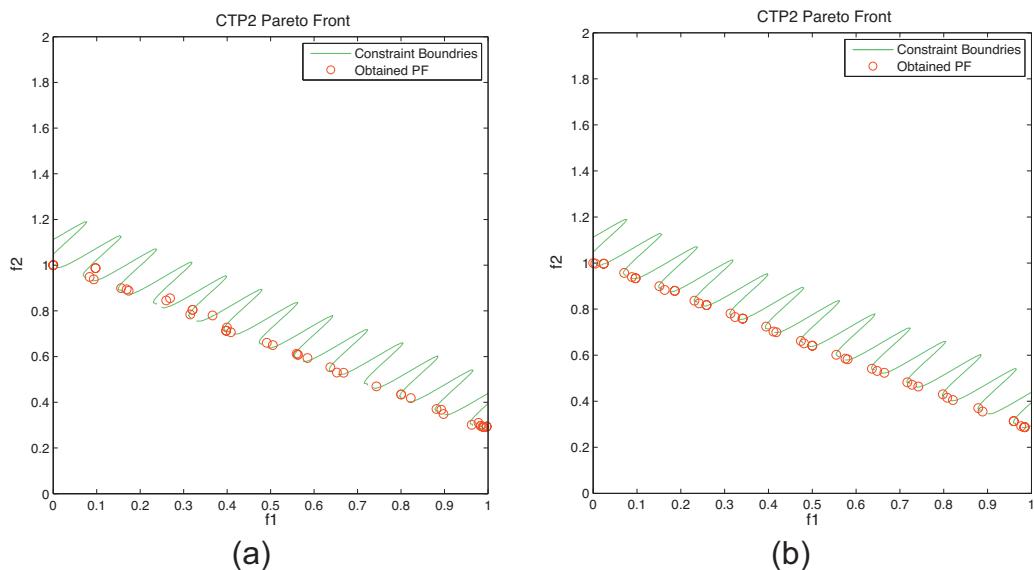


Fig. 17. Plots of the final solutions obtained from CMOEA/D-DE-SR (a) and CMOEA/D-DE-CDP (b) with population size of 50 evolved over 200 generations for CTP2.

have small evenly distributed population at a low computational cost.

Since test instance CTP2 has disconnected continuous Pareto-optimal regions, any Pareto-dominance based algorithm that prefers feasible solutions over infeasible solutions in their selection/replacement schemes, like NSGA-II with CDP, is likely to face difficulty in capturing the whole PF unless a large population size is used to maintain population diversity [19]. However, our decomposition based proposed algorithm with CDP, CMOEA/D-DE-CDP, which defines a separate aggregation function for each individual of the population can attain the whole PF even with a small population.

10. Conclusions

In this paper, the two well known penalty-parameterless constraint handling techniques SR and CDP are adapted for the CMOEA/D-DE framework. This led to two new algorithms CMOEA/D-DE-SR and CMOEA/D-DE-CDP.

From the experimental results on CTP-series and CF-series test instances, the following points can be concluded.

- The comparison of infeasible solutions based on constraint violation only, as is done in CMOEA/D-DE-CDP, or of a small percentage of infeasible solutions based on aggregation function values and the rests based on constraint violation, as is done in CMOEA/D-DE-SR, leads to better performance for most of the CTP-series and CF-series test instances. However, the former technique works comparatively better than the latter one in the framework CMOEA/D-DE.
 - CMOEA/D-DE-CDP mostly surpasses CMOEA/D-DE-SR in terms of converging to the PFs, uniformity of the nondominated solutions, and small variance in the objective space on both test suits.
 - The comparison of CMOEA/D-DE-CDP and CMOEA/D-DE-SR with NSGA-II and IDEA revealed that both of our algorithms can find better and consistent statistics, in terms of the HV-metric, than the two contenders for five CTP-series test instances, CTP2–CTP5.

- CTP7. This better performance could be attributed to algorithmic framework CMOEA/D-DE.
- The comparison of CMOEA/D-DE-CDP and CMOEA/D-DE-SR with the three best performers in CEC 2009 special session and competition indicated that CMOEA/D-DE-SR found the better best (i.e., lowest) IGD-metric value for CF9 and the second best IGD-metric values for test instances CF1, CF2, CF4, and CF8. CMOEA/D-DE-CDP found the better best results for test instances CF1 and CF4 and the second best results for test instances CF3, CF6 and CF9. Particularly, for test instances CF1, CF8, and CF9, better statistics are found by our algorithms.
 - The experimental results also suggest that our algorithms could work well in the presence of a small portion of feasible solutions in the initial population and comparatively lenient constraints. However, if the initial population is highly infeasible due to the presence of hard constraints, then the algorithms could fail to converge to the feasible region and trap in the infeasible region.
 - Both algorithms are sensitive to the settings of their various parameters except CR and F.
 - Both algorithms can find evenly distributed optimal solutions with a small population size, N.

References

- [1] K. Miettinen, Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, 1999.
- [2] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, John Wiley & Sons Ltd., 2001.
- [3] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, *IEEE Transactions on Evolutionary Computation* 11 (6) (2007) 712–731.
- [4] H. Li, Combination of evolutionary algorithms with decomposition techniques for multiobjective optimization, Ph.D. Thesis, University of Essex, 2008.
- [5] C.M. Fonseca, P.J. Fleming, Genetic algorithms for multiobjective optimization: formulation, discussion and generalization, in: Proceedings of the Fifth International Conference Genetic Algorithms, Morgan Kaufmann, Urbana-Champaign, IL, USA, 1993, pp. 416–423.
- [6] J. Horn, N. Nafpliotis, D.E. Goldberg, A niched Pareto genetic algorithm for multiobjective optimization, in: Proceedings of the First IEEE Conference on Evolutionary Computation, vol. 1, IEEE Service Center, Piscataway, NJ, 1994, pp. 82–87.
- [7] J.D. Knowles, D.W. Corne, The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation, in: IEEE Congress on Evolutionary Computation (CEC 1999), IEEE, Washington, DC, USA, 1999, pp. 98–105.
- [8] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: improving the strength Pareto evolutionary algorithm for multiobjective optimization, in: Evolutionary Methods for Design, Optimisation and Control, CIMNE, Barcelona, Spain, 2002, pp. 95–100.
- [9] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation* 6 (2) (2002) 182–197.
- [10] H. Ishibuchi, T. Murata, A multi-objective genetic local search algorithm and its application to flowshop scheduling, *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews* 28 (3) (1998) 392–403.
- [11] Y.W. Leung, Y. Wang, Multiobjective programming using uniform design and genetic algorithm, *IEEE Transactions on Systems, Man and Cybernetics Part C* 30 (3) (2000) 293–304.
- [12] T. Murata, H. Ishibuchi, M. Gen, Specification of genetic search directions in cellular multi-objective genetic algorithms, in: Evolutionary Multi-Criterion Optimization, Springer, 2001, pp. 82–95.
- [13] A. Jaszkiewicz, On the performance of multiple-objective genetic local search on the 0/1 knapsack problem – a comparative experiment, *IEEE Transactions on Evolutionary Computation* 6 (4) (2002) 402–412.
- [14] E.J. Hughes, Multiple single objective Pareto sampling, in: IEEE Congress on Evolutionary Computation (CEC 2003), IEEE Press, Canberra, 2003, pp. 2678–2684.
- [15] T.P. Runarsson, X. Yao, Stochastic ranking for constrained evolutionary optimization, *IEEE Transactions on Evolutionary Computation* 4 (3) (2000) 284–294.
- [16] H. Li, Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, *IEEE Transactions on Evolutionary Computation* 13 (2) (2009) 284–302.
- [17] K. Deb, A. Pratap, T. Meyarivan, Constrained test problems for multi-objective evolutionary optimization, in: First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001), LNCS, vol. 1993, Springer, Zurich, Switzerland, 2001, pp. 284–298.
- [18] Q. Zhang, W. Liu, H. Li, The performance of a new version of MOEA/D on CEC09 unconstrained mop test Instances, in: Special Session on Performance Assessment of Multiobjective Optimization Algorithms/CEC 09 MOEA Competition, Norway, 2009.
- [19] T. Ray, H.K. Singh, A. Isaacs, W. Smith, in: E. Mezura-Montes (Ed.), Constraint-Handling in Evolutionary Computation, Studies in Computational Intelligence, vol. 198, Springer, Berlin, 2009, ISBN 978-3-642-00618-0, pp. 145–165 (Chapter 7).
- [20] L. Tseng, C. Chen, Multiple trajectory search for unconstrained/constrained multi-objective optimization, in: IEEE Congress on Evolutionary Computation, CEC2009, IEEE, 2009, pp. 1951–1958.
- [21] H. Liu, X. Li, The multiobjective evolutionary algorithm based on determined weight and sub-regional search, in: IEEE Congress on Evolutionary Computation, CEC2009, IEEE, 2009, pp. 1928–1934.
- [22] M. Liu, X. Zou, Y. Chen, Z. Wu, Performance assessment of DMOEA-DD with CEC 2009 MOEA competition test instances, in: IEEE Congress on Evolutionary Computation, CEC2009, IEEE, 2009, pp. 2913–2918.
- [23] R. Steuer, R. Steuer, Multiple Criteria Optimization: Theory, Computation, and Application, vol. 233, Wiley, 1986.
- [24] S. Gass, T. Saaty, The computational algorithm for the parametric objective function, *Naval Research Logistics Quarterly* 2 (1–2) (1955) 39–45.
- [25] L. Zadeh, Optimality and non-scalar-valued performance criteria, *IEEE Transactions on Automatic Control* 8 (1) (1963) 59–60.
- [26] I. Das, J.E. Dennis, Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems, *SIAM Journal on Optimization* 8 (3) (1998) 631–657.
- [27] A. Messac, A. Ismail-Yahaya, C.A. Mattson, The normalized normal constraint method for generating the Pareto frontier, *Structural and Multidisciplinary Optimization* 25 (2) (2003) 86–98.
- [28] K. Price, R.M. Storn, J.A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, Natural Computing, Springer-Verlag, Secaucus, NJ, USA, 2005.
- [29] K. Deb, An efficient constraint handling method for genetic algorithms, *Computer Methods in Applied Mechanics and Engineering* 186 (2–4) (2000) 311–338.
- [30] T. Okabe, Y. Jin, B. Sendhoff, A critical survey of performance indices for multi-objective optimization, in: IEEE Congress on Evolutionary Computation, vol. 2, IEEE, Canberra, Australia, 2003, pp. 878–885.
- [31] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V.G. da Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, *IEEE Transactions on Evolutionary Computation* 7 (2) (2003) 117–131.
- [32] Q. Zhang, A. Zhou, S. Zhao, P.N. Suganthan, W. Liu, S. Tiwari, Multiobjective optimization test instances for the CEC 2009 special session and competition, Tech. Rep. CES-487, University of Essex and Nanyang Technological University, 2008.
- [33] D. Van Veldhuizen, Multiobjective evolutionary algorithms: classifications, analyses, and new innovations, in: Evolutionary Computation, Citeseer, 1999.
- [34] E. Zitzler, L. Thiele, Multiobjective optimization using evolutionary algorithms— a comparative case study, in: Parallel Problem Solving from Nature-PPSN V, Springer, 1998, pp. 292–301.