

Threshold Based Penalty Functions for Constrained Multiobjective Optimization

Muhammad Asif Jan
Department of Mathematics
Kohat University of Science & Technology
Khyber Pakhtunkhwa, Pakistan

Nasser Mansoor Tairan
College of Computer Science
King Khalid University Abha,
Saudi Arabia

Rashida Adeeb Khanum
Jinnah College for Women
University of Peshawar Khyber
Pakhtunkhwa, Pakistan

Wali Khan Mashwani
Department of Mathematics
Kohat University of Science & Technology
Khyber Pakhtunkhwa, Pakistan

Abstract—This paper compares the performance of our recently proposed threshold based penalty function against its dynamic and adaptive variants. These penalty functions are incorporated in the update and replacement scheme of the multiobjective evolutionary algorithm based on decomposition (MOEA/D) framework to solve constrained multiobjective optimization problems (CMOPs). As a result, the capability of MOEA/D is extended to handle constraints, and a new algorithm, denoted by CMOEA/D-DE-TDA is proposed. The performance of CMOEA/D-DE-TDA is tested, in terms of the values of IGD-metric and SC-metric, on the well known CF-series test instances. The experimental results are also compared with the three best performers of CEC 2009 MOEA competition. Empirical results show the pitfalls of the proposed penalty functions.

Keywords—Constrained multiobjective optimization; decomposition; MOEA/D; dynamic and adaptive penalty functions; threshold.

I. INTRODUCTION

In this paper, we consider the following constrained multiobjective optimization problem (CMOP) [1]:

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

where $x = (x_1, \dots, x_n)^T \in \mathcal{R}^n$ is an n dimensional vector of decision variables, F is the objective vector function that consists of m real-valued objective functions, and $g_i(x) \geq 0$ 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 (also called bound constraints) of x_k , $k = 1, \dots, n$, respectively, which define the search region, $\mathcal{S} = \{x = (x_1, \dots, x_n)^T \mid l_k \leq x_k \leq u_k, k = 1, \dots, n\}$.

A solution $x \in \mathcal{S}$ satisfying all the inequality constraints in (1) is called a feasible solution; otherwise, we call it an infeasible solution. The set of all feasible solutions is called the feasible region, denoted by \mathcal{F} , and the set of all infeasible solutions is called the infeasible region. Also, we define feasible attainable objective set (AOS) by $\{F(x) \mid x \in \mathcal{F}\}$.

Because the objectives in (1) often contradict one another, a single solution in the feasible search region could not be found that minimizes all the objectives simultaneously. Therefore, 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 [2], [3].

A solution x is said to Pareto-dominate or simply dominate another solution y , mathematically denoted as $x \preceq y$, if $f_i(x) \leq f_i(y)$, $\forall i = 1, \dots, m$ and $f_j(x) < f_j(y)$ 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 $F(x) \preceq F(x^*)$. $F(x^*)$ is then called a Pareto-optimal (objective) vector. 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 [2].

A common way to deal with constraints in constrained optimization is to use penalty functions. In a penalty function approach, the penalty coefficients balance the objective and penalty functions. However, finding appropriate penalty coefficients to strike the right balance is a big challenge in itself [4]. They depend on the problems in hand. Thus, some researchers suggested dynamic penalty functions [5]–[7] to avoid the difficulty of setting penalty coefficients in static penalty functions and to explore infeasible regions.

In dynamic penalty functions, the current generation number (or the number of solutions searched) is considered in the calculation of the penalty coefficients. The penalty to infeasible solutions is small at the beginning of the search due to the initial small generation numbers used in the formulation, and it then increases by the increase in the generation number. As a result, these methods converge to feasible solution (s) at the end of evolution. Generally, dynamic penalty functions are effective, but they are not adaptive to the ongoing success (or failure thereof) of the search and cannot guide the search

¹One has to reverse all the inequalities if the goal is to maximize the objectives in (1). By the term “dominate” we mean “better than”

to particularly promising regions or away from unpromising regions based on what has already been observed [6]. Furthermore, like static penalty functions, they also need problem specific tuning to perform well [6].

Adaptive penalty functions not only incorporate the current generation number (or search length), but they also consider the feedback from the search in their formulations. Thus, they are benefited from the search history. The penalty coefficients in such methods are adjusted based on what has already been accomplished.

In [1], [8], we proposed a novel threshold-based penalty function for handling constraints in constrained multiobjective optimization (CMOO). The threshold using the minimum and maximum constraint violation in the neighborhood of a solution and controlled by a scaling parameter dynamically adjusts the penalty to infeasible solutions. Moreover, Infeasible solutions with constraint violation less than the threshold are less penalized than the ones with constraint violation greater than the threshold. For this purpose, two additional parameters are used.

In this paper, first we propose a parameterless threshold value contrary to the one used in [8]. Secondly, we define the dynamic and adaptive versions of the threshold based penalty function [8]. These penalty functions are then implemented in one of the improved frameworks of MOEA/D [9], MOEA/D-DE [10] to solve hard CF-series [11] test instances.

The remainder of this paper is organized as follows. Section II presents the Tchebycheff aggregation function and the threshold-based penalty function and its dynamic and adaptive variants. Section III briefly introduces MOEA/D and modifies the algorithmic framework of MOEA/D-DE [10] for CMOPs. Section IV discusses the experimental settings. Section V presents and discusses experimental results on CF-series [11] test instances. Section VI compares our experimental results with the three best performers [12]–[14] of CEC 2009 MOEA competition. Finally, Section VII outlines a summary of the paper.

II. TCHEBYCHEFF AGGREGATION FUNCTION AND THRESHOLD BASED PENALTY FUNCTIONS

A. Tchebycheff Aggregation Function

MOEA/D [9] needs to decompose a multiobjective optimization problem (MOP) into a number of scalar objective subproblems. For this purpose, this work uses the Tchebycheff aggregation function. The reasons for choosing this function include: first, it is less sensitive to the shape of PF. Second, it can be used to find the Pareto-optimal solutions in both convex and nonconvex PFs. It is defined as follows [15]:

$$\begin{aligned} \text{Minimize} \quad & g^{te}(x|\lambda, z^*) = \max_{1 \leq i \leq m} \{\lambda_i |f_i(x) - z_i^*|\}; \\ \text{Subject to} \quad & x \in \mathcal{F} \subset \mathcal{R}^n; \end{aligned}$$

where $z^* = (z_1^*, \dots, z_m^*)^T$ is the reference point, i.e., $z_i^* = \min\{f_i(x) | x \in \mathcal{F}\} \forall i = 1, \dots, m$ and $\lambda = (\lambda_1, \dots, \lambda_m)^T$ is a weight vector such that $\lambda_i \geq 0 \forall i = 1, \dots, m$ and $\sum_{i=1}^m \lambda_i = 1$.

Theorems describing the Pareto-optimality conditions of Tchebycheff aggregation function are available in [2].

B. Threshold Based Penalty Functions

The proposed penalty function in [1], [8] uses a threshold value, τ for dynamically controlling the amount of penalty to infeasible solutions. It is computed as follows.

Assume that MOEA/D [9] decomposes the given MOP into N subproblems. In each iteration, MOEA/D keeps N individual solutions x^1, \dots, x^N , where x^i is the current solution to subproblem i . If P is supposed to be the mating and update range in MOEA/D (for details, please see Algorithm 1, Section III). Then we define [1], [8]:

$$V_{min} = \min\{V(x^i), i \in P\}. \quad (2)$$

$$V_{max} = \max\{V(x^i), i \in P\}. \quad (3)$$

Where $V(x^i) = |\sum_{j=1}^p \min(g_j(x^i), 0)|$ is the degree of constraint violation of solution x^i . Note that the constraints are normalized before calculating this value.

The threshold value, τ is defined as [1], [8]:

$$\tau = V_{min} + s(V_{max} - V_{min}), \quad (4)$$

where parameter s controls the threshold value, τ .

The proposed threshold based penalty function encourages the algorithm to search the feasible region and the infeasible region near the feasible region. It is defined as follows [1], [8]: For $i = 1, \dots, m$

$$f_p^i(x) = \begin{cases} f_i(x) + s_1 V^2(x), & \text{if } V(x) < \tau; \\ f_i(x) + s_1 \tau^2 + s_2 (V(x) - \tau), & \text{otherwise.} \end{cases} \quad (5)$$

In [8], we have tested the algorithm with a fixed value of parameter s (i.e., we used $s = 0.3$ in Eq. 4). However, due to the changing values of V_{min} and V_{max} during the evolutionary process, the resulting threshold value, τ still remains adaptive.

It might also possible that if the values of V_{min} and V_{max} are quite away from one another, then the resulting threshold value, τ will be large. Consequently, more infeasible solutions will have a chance to propagate in the evolution process, and the search might quickly converge to the feasible region. As a result, the low quality optimal solutions might be obtained. Thus, in order to avoid such situation, this paper sets the threshold value, τ equal to the mean value of the degree of constraint violations of all infeasible solutions in the neighborhood of a solution. That is [1]:

$$\tau = \frac{1}{n_{inf}} \sum_{i \in P} V(x^i), \quad (6)$$

where n_{inf} is the number of infeasible solutions in the neighborhood of a solution. This reduces the effort to choose various values for parameter s .

Eq. 5 employs two additional fixed penalty parameters, s_1 and s_2 , where $s_1 \ll s_2$, to penalize infeasible solutions. Infeasible solutions whose degree of constraint violation is smaller than τ are less penalized than the ones with degree of constraint violation greater than τ . This is realized by scaling the respective violations by parameters s_1 and s_2 . However, this paper adjusts s_1 and s_2 dynamically and adaptively. As

a result, the dynamic and adaptive variants of Eq. 5 are established. These variants are defined as follows [1]
For $i = 1, \dots, m$

$$f_p^i(x) = \begin{cases} f_i(x) + (g/G)^2 V^2(x), & \text{if } V(x) < \tau; \\ f_i(x) + (g/G)^2 \tau^2 + \\ (g/G)(V(x) - \tau), & \text{otherwise,} \end{cases} \quad (7)$$

where $f_p^i(x)$ is the i -th penalized objective function value, g is the current generation number and G is the total number of generations.

When $r_{inf} \neq 0$,

$$f_p^i(x) = \begin{cases} f_i(x) + (r_{inf})^2 V^2(x), & \text{if } V(x) < \tau; \\ f_i(x) + (r_{inf})^2 \tau^2 + \\ (r_{inf})(V(x) - \tau), & \text{otherwise,} \end{cases} \quad (8)$$

where $f_p^i(x)$ is the i -th penalized objective function value and r_{inf} is the infeasibility ratio (i.e., the ratio of the number of infeasible solutions to the total number of neighboring solutions) in the neighborhood of a solution.

Initially in Eq. 7, infeasible solutions are less penalized due to small g values. Thus, infeasible solutions with degree of constraint violation less than τ get particularly more chances to evolve. As a result, infeasible regions are well explored in the beginning of the search. However, later on with the increase in generation number, g , the penalty to infeasible solutions increases. As a result, the search converges to the feasible region at the later stage of the algorithmic run.

On the other hand in Eq. 8, the penalty to infeasible solutions depends on the number of infeasible solutions in the neighborhood of a solution. Thus, the less is the number of infeasible solutions in the neighborhood of a solution, the smaller the penalty is; otherwise, it will increase with the increase in the number of infeasible solutions. Here again, because of the employed penalty parameters, infeasible solutions with degree of constraint violation less than τ get more opportunities to evolve.

Algorithm 1 Pseudo-code of Update Scheme of CMOEA/D-DE-TDA. n_r is the number of solutions updated by a better child solution.

- 1: Each new child solution y updates n_r solutions from the set P of its neighboring solutions 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 with the new objective values of Eqs. 5, 7, and 8;
 - 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**
-

III. MOEA/D AND CMOEA/D-DE-TDA

Zhang and Li [9] proposed a simple but efficient MOEA called MOEA/D. It approximates the PF by explicitly decomposing an MOP into a number of scalar objective optimization subproblems (e.g., Tchebycheff aggregation function is employed for this purpose). These subproblems are then optimized concurrently and collaboratively by evolving a population of solutions. An EA is employed for this purpose. 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-DE [10] is an updated and efficient version of MOEA/D. The framework of this algorithm is altered for CMOPs. The modified framework is denoted by CMOEA/D-DE (please see [16] for more details).

We employ the penalty functions defined by Eqs. 5, 7, and 8 with τ given by Eq. 6 in the replacement and update scheme of CMOEA/D-DE to solve CF-series [11] test instances. This resulted in a new algorithm, denoted by CMOEA/D-DE-TDA. The pseudo-code of the update scheme of CMOEA/D-DE-TDA is given in Algorithm 1 [1].

IV. EXPERIMENTAL SETTINGS

In our experiments, the same parameters' settings are used as in [16]. The weight vectors used in Eq. (2) are set as per criteria mentioned in [11]. We employ the inverted generational distance metric (IGD-metric) [9], [17] statistics for comparing the obtained results. For calculating the IGD-metric values, we select 100 feasible nondominated solutions in the case of 2-objective and 150 in the case of 3-objective test instances from each final population. The final solution set P is selected as per criteria given in [11]. We also employ the set coverage metric (SC-metric) [9] to compare the obtained nondominated solutions. One of the reasons for choosing these metrics is that they could measure both convergence and diversity of the approximated solutions in a sense. Secondly, the algorithms in comparison have also used these performance metrics.

V. EXPERIMENTAL RESULTS

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

Table I presents the best (i.e., lowest), mean, and standard deviation of the IGD-metric values based on 30 independent runs found by CMOEA/D-DE-TDA with Eqs. 5, 7, and 8 for CF-series test instances. As it can be seen from this table that CMOEA/D-DE-TDA with Eq. 5 found better statistics for three test instances CF1, CF6, and CF7 except the standard deviation value on CF7. The table also shows that the algorithm with Eq. 7 found better results than those obtained with Eqs. 5, 8 for test instances CF2, CF4 and CF5 except the best values on CF2 and CF5. Similarly, it showed superior performance with Eq. 8 for the 3-objective test instances except the standard deviation value and best value for test instances CF9 and CF10, respectively. The performance of the algorithm may be considered as comparable with all three formulations for test instances CF2, CF8, and CF9, as there is a marginal difference in the IGD-metric statistics for these test instances.

TABLE I: COMPARISON OF THE IGD-METRIC STATISTICS OF THE ALGORITHM FOR CF1-CF10. THE RESULTS IN **BOLDFACE** AND IN *ITALIC* INDICATE THE BETTER AND THE SECOND BETTER RESULTS.

Test Instance	best (lowest)			mean			st. dev.		
	Eq. 5	Eq. 7	Eq. 8	Eq. 5	Eq. 7	Eq. 8	Eq. 5	Eq. 7	Eq. 8
CF1	0.0003	<i>0.0034</i>	0.0035	0.0005	<i>0.0051</i>	0.0076	0.0002	<i>0.0017</i>	0.0027
CF2	0.0026	<i>0.0028</i>	0.0026	<i>0.0038</i>	0.0037	0.0045	<i>0.0014</i>	0.0011	0.0020
CF3	0.0632	0.0632	0.0632	0.1382	0.1382	0.1382	0.0441	0.0441	0.0441
CF4	<i>0.0056</i>	0.0054	0.0062	<i>0.0083</i>	0.0076	0.0106	<i>0.0018</i>	0.0016	0.0041
CF5	0.0326	0.0526	<i>0.0328</i>	0.1560	0.1175	<i>0.1356</i>	0.0932	0.0348	<i>0.0921</i>
CF6	0.0069	0.0423	<i>0.0387</i>	0.0220	0.1667	<i>0.1241</i>	0.0160	0.1162	<i>0.0303</i>
CF7	0.0374	<i>0.0551</i>	0.0556	0.1227	<i>0.1474</i>	0.1567	0.0787	0.0684	<i>0.06590</i>
CF8	0.0344	<i>0.0336</i>	0.0335	0.0380	<i>0.0359</i>	0.0357	0.0017	<i>0.0016</i>	0.0013
CF9	<i>0.0431</i>	0.0433	0.0429	0.0475	<i>0.0473</i>	0.0471	<i>0.0023</i>	0.0022	0.0025
CF10	0.1020	<i>0.1054</i>	0.1062	0.1602	<i>0.1487</i>	0.1430	0.0464	<i>0.0380</i>	0.0308

Overall, the algorithm is to be run with Eq.5 for better best IGD-metric values on most of the CF-series test instances. However, it is to be run with Eq. 7 or with Eq. 8 for the better mean and standard deviation values.

Moreover, the small values for the mean of IGD-metric for test instances CF1, CF2, CF4, CF8, CF9 show that the final nondominated solutions obtained by the algorithm with all three Eqs. 5, 7, and 8 approximate the PF very well for these test instances, in a sense.

TABLE II: THE AVERAGE SET COVERAGE AMONGST EQs. 5 (S), 7 (D), 8 (A) WHEN USED BY THE ALGORITHM. THE RESULTS IN **BOLDFACE** INDICATE THE BETTER RESULTS; IF NOT, THEY ARE IDENTICAL.

Test Instance	C(S, D)	C(D, S)	C(S, A)	C(A, S)	C(D, A)	C(A, D)
CF1	0.01	0.94	0.11	0.84	0.82	0.16
CF2	0.19	0.22	0.18	0.20	0.19	0.22
CF3	0.65	0.65	0.65	0.65	0.65	0.65
CF4	0.26	0.32	0.30	0.27	0.34	0.26
CF5	0.22	0.18	0.21	0.21	0.21	0.21
CF6	0.03	0.08	0.03	0.06	0.09	0.04
CF7	0.26	0.23	0.22	0.21	0.25	0.22
CF8	0.03	0.06	0.04	0.06	0.05	0.05
CF9	0.03	0.04	0.02	0.03	0.03	0.03
CF10	0.28	0.39	0.31	0.37	0.36	0.31

Table II presents the average set coverage among the nondominated solutions of CMOEA/D-DE-TDA obtained with Eqs.5, 7, and 8. The results of this table disclose that, in terms of the SC-metric, the nondominated solutions found by CMOEA/D-DE-TDA with Eq. 7 are better than those obtained with Eq.5 except for test instances CF5 and CF7. However, due to a small difference in the SC-metric values, the nondominated solutions found by the algorithm with both Eqs. 5, 7 may be considered as comparable for all CF-series test instances except CF1 and CF10. The difference in the SC-metric values is much bigger in case of test instance CF1, while it is reasonably big in case of test instance CF10. Similarly, by the same reason, the nondominated solutions obtained with Eqs. 5, 8 may be considered as comparable except CF1 and CF10, and the ones obtained with Eqs. 7, 8 may be thought as similar except CF1.

Figures 1–5 show, in the objective space, the distributions of the 100 and 150 nondominated population members for the seven 2-objective, CF1-CF7, and the three 3-objective, CF8-

CF10, CF-series test instances, respectively. These solutions are selected based on the criteria given in [11] 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 nondominated fronts of these selected 100 and 150 nondominated solutions.

From these figures, it is very clear that CMOEA/D-DE-TDA with all three Eqs. 5, 7, and 8 found good approximations of the PFs for the three 2-objective, CF1, CF2, and CF4 and two 3-objective, CF8, CF9, test instances. However, it performed poorly on three 2-objective, CF3, CF5, CF7, and one 3-objective, CF10 test instances. The algorithm found good approximation of the PF for test instance CF6 with Eq.5 than those obtained with the other two equations Eqs.7, 8. In the latter case, solutions in the lower part of the PF are not found. Furthermore, it is also evident from the plots of 30 nondominated fronts of test instance CF4 that CMOEA/D-DE-TDA fails to find the whole PF in some runs.

Since the PF of CF3 is discontinuous and concave, so, it might be harder as compared to all other 2-objective test instances for the algorithm. Furthermore, although the PFs of CF4 and CF5, CF6 and CF7, and CF9 and CF10 are identical, the poor performance of CMOEA/D-DE-TDA with all three Eqs. 5, 7, and 8 on test instances CF5, CF7 and CF10 could be due to the presence of relatively harder objective and constraint functions in these test instances than test instances CF4, CF6, and CF9 (see [11]).

Figure 6 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-TDA with Eq.5 converges faster, in terms of IGD-metric values, than with Eqs. 7, 8 for test instances CF1, CF6 and CF7.

For test instances CF2 and CF4, the IGD-metric values obtained by the algorithm with Eq. 8 are higher than those obtained with Eqs. 5, 7. Although initially different, the algorithm with latter two equations performs similarly. Particularly, the IGD-metric values obtained by the algorithm are almost the same in the later generations.

For test instance CF5, the algorithm with Eq.8 can converge faster in terms of IGD-metric values than with the other two Eqs. 5, 7.

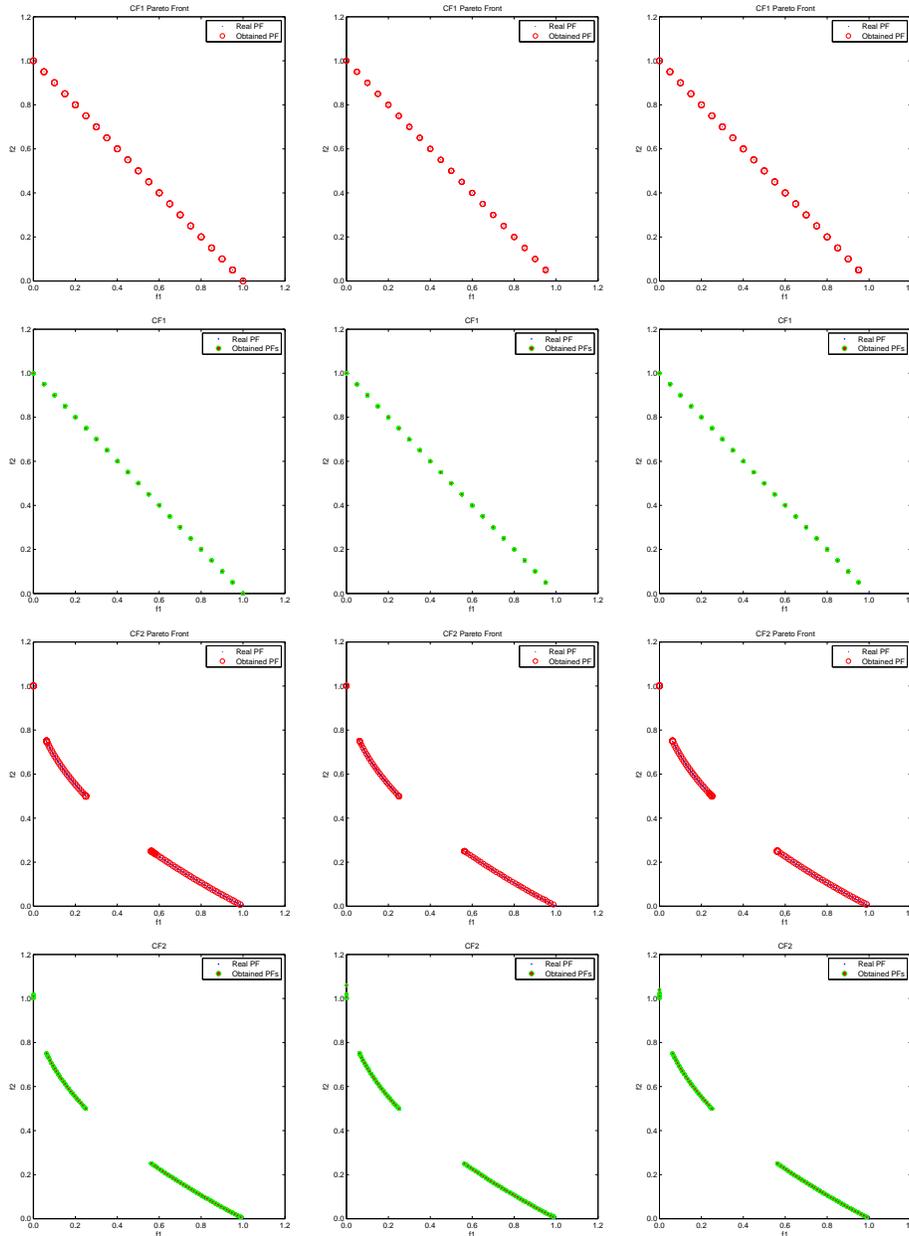


Fig. 1: Plots of the nondominated front with the best IGD value and all the 30 final nondominated fronts found by the algorithm with Eq. 5 (left column), Eq. 7 (middle column), and Eq. 8 (right column) for CF1 and CF2.

Figure 6 also shows that the algorithm with all three Eqs. 5, 7, and 8 converges at the same rate in terms of IGD-metric values for the three 3-objective test instances, CF8-CF10.

Figure 7 shows the average generation feasibility versus generations's graphs. This figure shows that CMOEA-DE-TDA with Eq. 5 converges to the feasible region faster than with Eqs. 7, 8 for all test instances except CF10, where it converges to the feasible region at the same rate with all three Eqs. 5, 7, and 8. Specifically, the feasibility ratio equates to 1 by generations 10 to 50 for test instances CF1-CF7.

The algorithm approaches to the feasible region faster with Eq. 7 than with Eq. 8 for test instances CF1, CF5, CF7, while

the situation is vice versa for test instances CF4 and CF6. Moreover, it converges to the feasible region at the same rate for the test instances CF2, CF8, and CF9 with both Eqs. 7, 8.

Figure 7 shows that 50 % or more of the initial populations for test instances CF1-CF5 is feasible. These feasible solutions propagate quickly in the subsequent generations of the algorithm with Eq. 5, and the feasibility ratio becomes 1 by generations 10-50. However, the algorithm retains infeasible solutions until the end of the algorithmic runs with Eqs. 7, 8, and the final feasibility ratios remain in the range (0.6 - 1) for these test instances. Additionally, with Eq. 7, the feasibility ratio initially remains small even in some cases drops to very

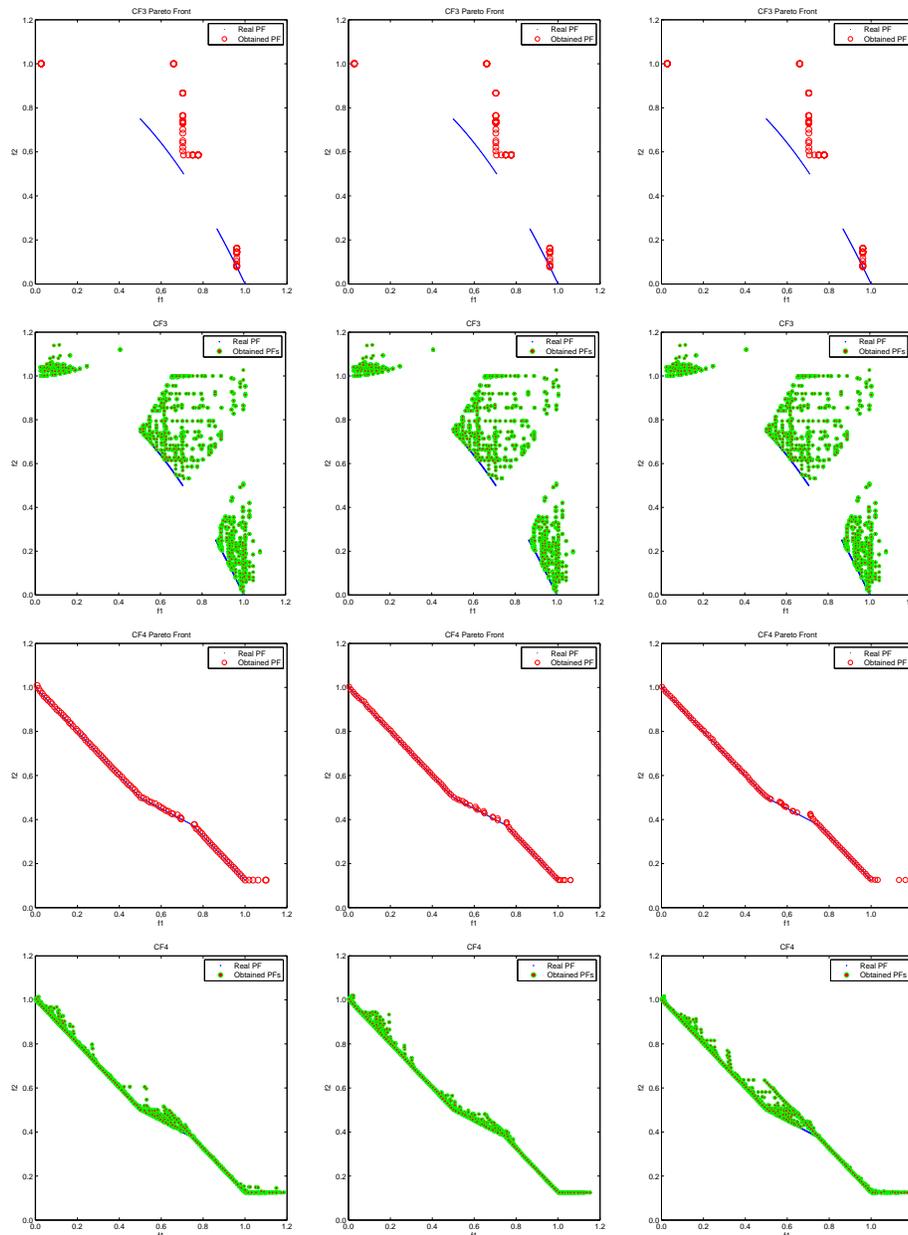


Fig. 2: Plots of the nondominated front with the best IGD value and all the 30 final nondominated fronts found by the algorithm with Eq. 5 (left column), Eq. 7 (middle column), and Eq. 8 (right column) for CF3 and CF4.

low value and then goes on increasing and approaching 1 from there (see Figure 7 for feasibility ratio graphs of test instances CF1 and CF2). The quick convergence to the feasible region is advantageous for test instances like CF1, CF2, and CF4, but can cause problems for harder test instance like CF5, where the PF is a piecewise continuous curve with three pieces like CF4, but its objective and constraint functions are quite different and harder than CF4 (see [11]).

On the other hand, about 25 % or below of the initial populations for test instances CF6-CF10 is feasible. Because of the threshold defined in this paper and individually defined scaling factors in the three penalty functions, the algorithm

has more chances to evolve better infeasible solutions during the evolutionary process in these test instances. Particularly, in the two 3-objective test instances CF8 and CF9, the average feasibility ratio at the last generations of the algorithmic runs is about 0.6 and about 0.85 with Eqs. 7, 8 and about 0.7 and about 0.9 with Eq. 5, respectively (see Figure 7). This way the infeasible regions near the feasibility boundaries in these two instances are well explored and could be a reason for the better performance of the algorithm on these two instances with all three equations.

Moreover, in test instance CF6, the feasibility ratio of CMOEA/D-DE-TDA with Eq. 5 becomes 1 after the initial 20

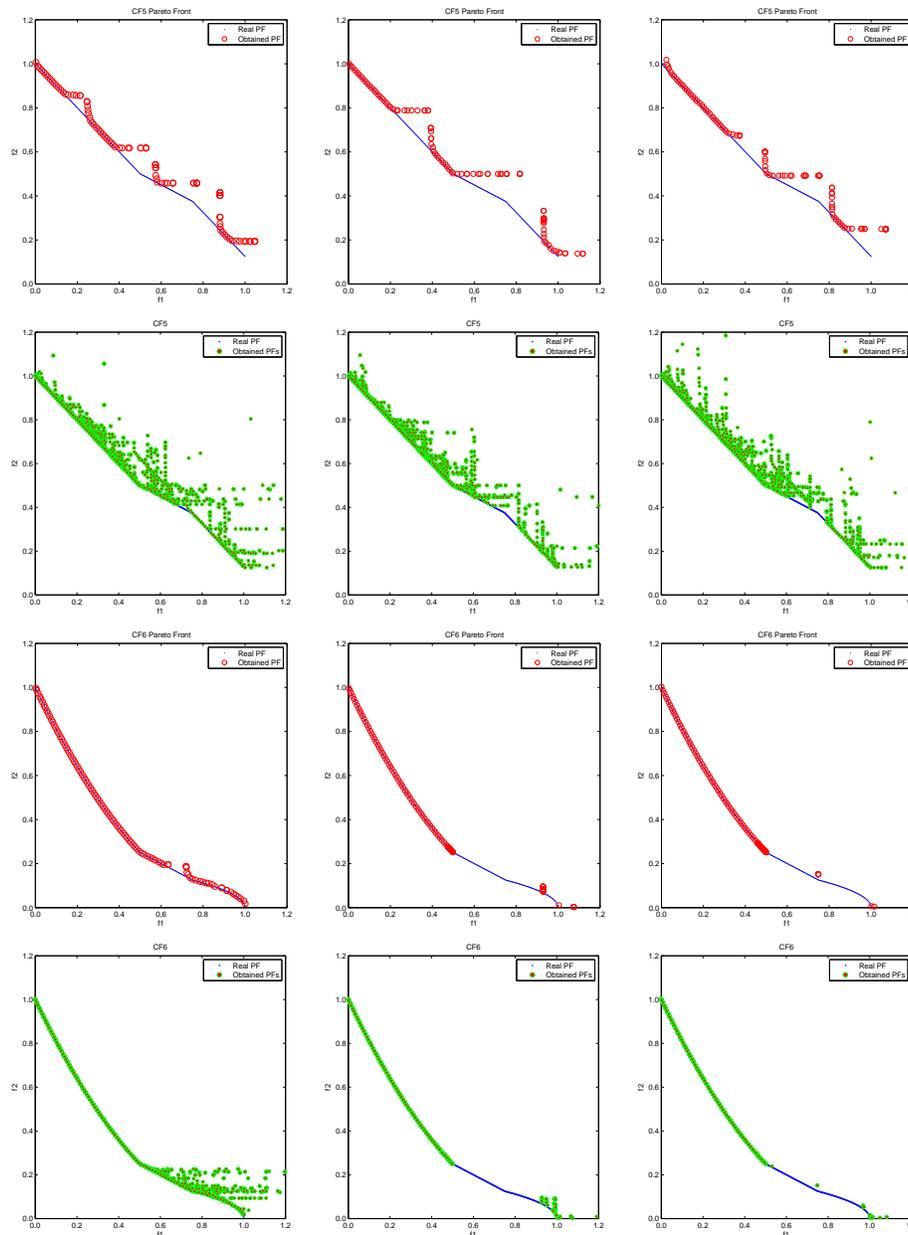


Fig. 3: Plots of the nondominated front with the best IGD value and all the 30 final nondominated fronts found by the algorithm with Eq. 5 (left column), Eq. 7 (middle column), and Eq. 8 (right column) for CF5 and CF6.

generations, and it remains mostly in the range (0.60 0.80) with Eqs. 7, 8, while in test instance CF7, it takes 50 generations of CMOEA/D-DE-TDA with Eq. 5 to become 1 and finally approaches to about 0.72 and about 0.8 with Eq. 8 and Eq. 7, respectively. Thus, the quick convergence to the feasible region and as a result more exploration of it is beneficial in case of Eq. 5 for test instances CF6 and CF7. While, retaining more infeasible solutions could be a reason for a comparatively poor performance of CMOEA/D-DE-TDA with Eqs. 7, 8 on these two test instances.

In test instance CF10, the feasibility ratio takes about 200 generations of CMOEA/D-DE-TDA with Eq. 5 and about 250 generations of CMOEA/D-DE-TDA with Eqs. 7, 8 to become

1. Here, the less exploration of the feasible region could be the reason for the poor performance of CMOEA/D-DE-TDA on this test instance.

VI. COMPARISON WITH THE THREE BEST PERFORMERS OF CEC 2009 MOEA COMPETITION

In this section, the results of CMOEA/D-DE-TDA are compared with the three best performers [12]–[14] in CEC 2009 MOEA competition on the CF-series test instances.

Table III compares the best, mean, and standard deviation values of the IGD-metric obtained from our algorithm, CMOEA/D-DE-TDA with Eqs. 5, 7, and 8 and the three best

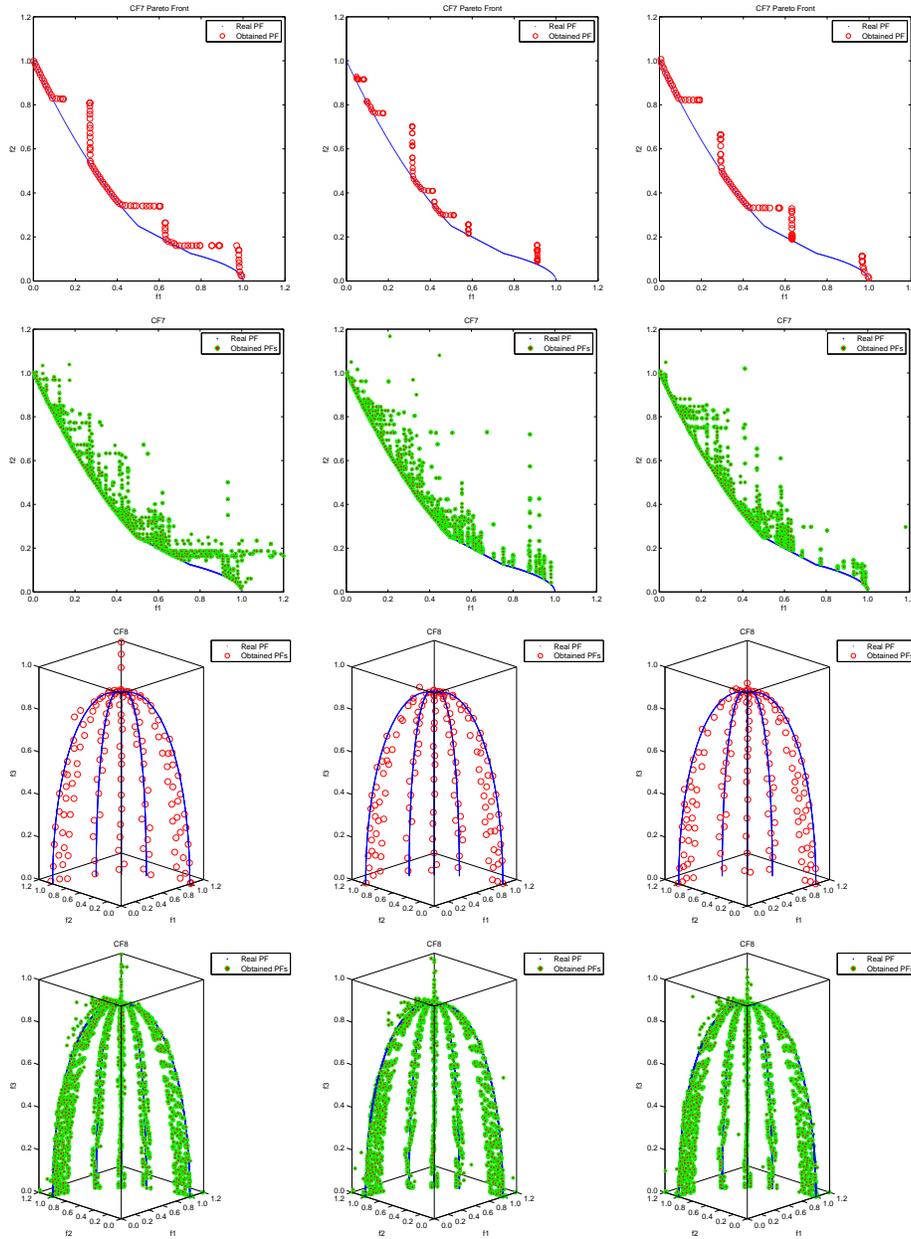


Fig. 4: Plots of the nondominated front with the best IGD value and all the 30 final nondominated fronts found by the algorithm with Eq. 5 (left column), Eq. 7 (middle column), and Eq. 8 (right column) for CF7 and CF8.

performers [12]–[14] in CEC 2009 MOEA competition for the CF-series test instances. It is clear that CMOEA/D-DE-TDA has achieved the best (i.e., lowest) IGD-metric value with Eq. 5 for test instance CF1, with Eq. 7 for test instance CF4, and with Eq. 8 for test instances CF8 and CF9. The table also shows that the algorithm has found the second best values for three test instances CF2, CF6, and CF10. Particularly, better statistics are found by our algorithm for test instances CF1, CF8 and CF9 except the standard deviation value on CF1 (It may be noted that our standard deviation value obtained with Eq. 5 is very close to the best standard deviation value). Further, our algorithm is the second best performer in terms of the mean and standard deviation IGD-metric values with Eq. 8 for test

instance CF10.

VII. CONCLUSIONS

In this paper, the performance of our proposed algorithm, CMOEA/D-DE-TDA is evaluated with three penalty functions given by Eqs. 5, 7, and 8 on the CF-series test instances. Here, Eqs. 7, 8 are the dynamic and adaptive versions of Eq. 5. The performance metrics used for comparison of the obtained results were IGD-metric and SC-metric. Moreover, the experimental results are compared with the three best performers of CEC 2009 MOEA competition.

We can conclude the following points from our experiments

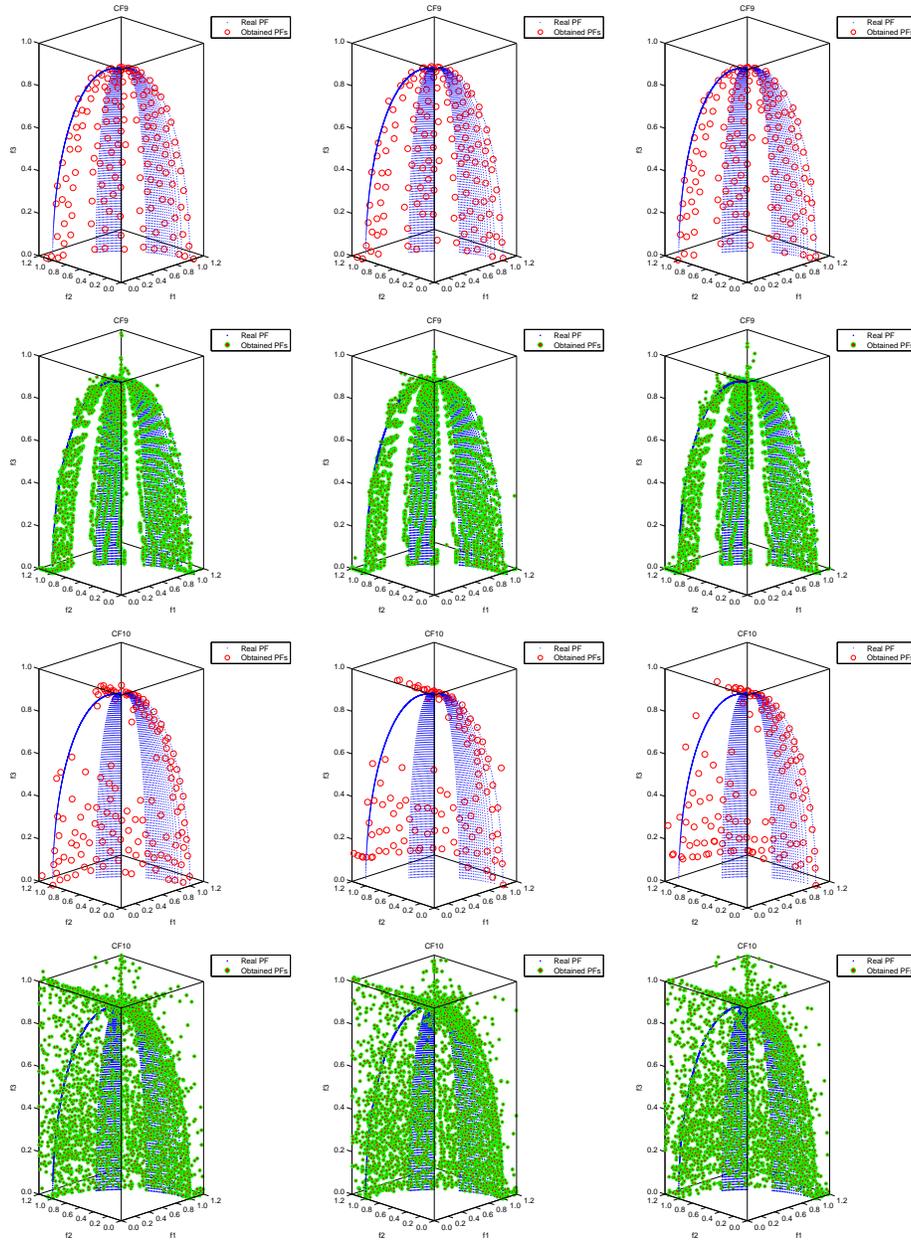


Fig. 5: Plots of the nondominated front with the best (lowest) IGD value and all the 30 final nondominated fronts found by the algorithm with Eq. 5 (left column), Eq. 7 (middle column), and Eq. 8 (right column) for CF9 and CF10.

conducted in this paper.

- CMOEA/D-DE-TDA with all three penalty functions works well even if there are few feasible solutions in the initial population.
- CMOEA/D-DE-TDA with the dynamic and adaptive penalty functions fails partly if there are few infeasible solutions in the initial population.
- The comparison of CMOEA/D-DE-TDA with the three best performers in CEC 2009 special session and competition indicated that CMOEA/D-DE-TDA has attained the best (i.e., lowest) IGD-metric value

with Eq. 5 for test instance CF1, with Eq. 7 for test instance CF4, and with Eq. 8 for test instances CF8 and CF9. The algorithm has also found the second best values for three test instances CF2, CF6, and CF10. Specifically, our algorithm overall found better statistics for test instances CF1, CF8, and CF9.

REFERENCES

- [1] M. A. Jan, "Decomposition Based Evolutionary Methods for Constrained Multiobjective Optimization," Ph.D. dissertation, University of Essex, 2011.
- [2] K. Miettinen, *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1999.

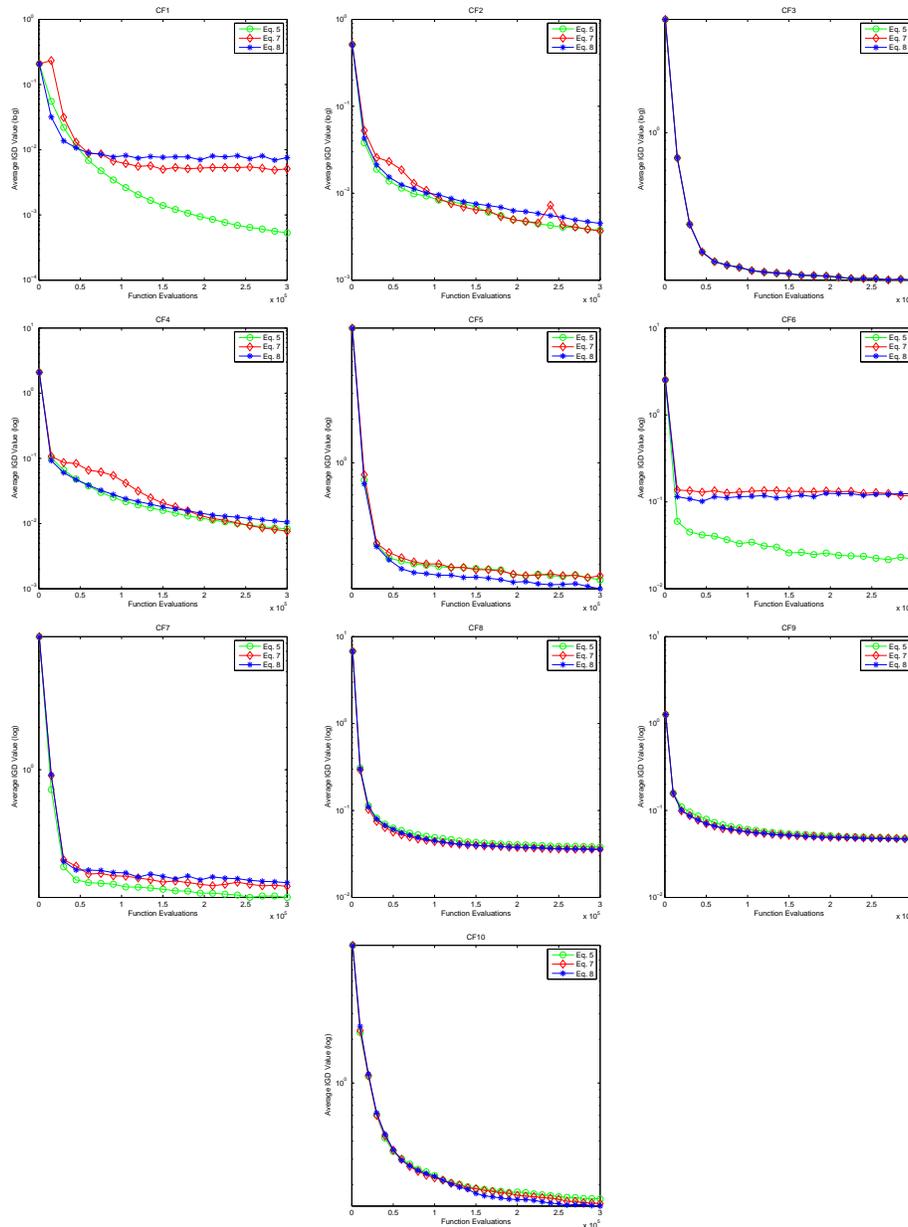


Fig. 6: Evolution of the IGD-metric values versus function evaluations when CMOEA/D-DE-TDA uses Eqs. 5, 7, and 8 for CF1-CF10.

[3] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons LTD, 2001.

[4] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, September 2000.

[5] J. Joines and C. Houck, "On the use of nonstationary penalty functions to solve nonlinear constrained optimization problems with gas," in *Proc. 1st IEEE Conf. Evol. Program.*, E. D. Fogel, Ed., Orlando, FL, 1994, pp. 579–584.

[6] A. E. Smith and D. W. Coit, "Constraint handling techniques penalty functions," In *Back, D.B. Fogel, T. Z. Michalewicz. eds. Handbook of Evolutionary Computation*, 1997.

[7] S. Kazarlis and V. Petridis, "Varying Fitness Functions in Genetic Algorithms: Studying the Rate of Increase of the Dynamic Penalty Terms," in *Proceedings of the 5th Parallel Problem Solving from Nature (PPSN V)*, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds., Amsterdam, The Netherlands. Heidelberg, Germany: Springer-Verlag, September 1998, pp. 211–220, lecture Notes in Computer Science Vol. 1498.

[8] M. A. Jan and Q. Zhang, "MOEA/D for constrained multiobjective optimization: Some preliminary experimental results," in *UK Workshop on Computational Intelligence (UKCI)*. IEEE, 2010, pp. 1–6.

[9] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[10] H. Li and Q. Zhang, "Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, April 2009.

[11] Q. Zhang, W. Liu, and H. Li, "The Performance of a New Version of MOEA/D on CEC09 Unconstrained Mop Test Instances," in *Special*

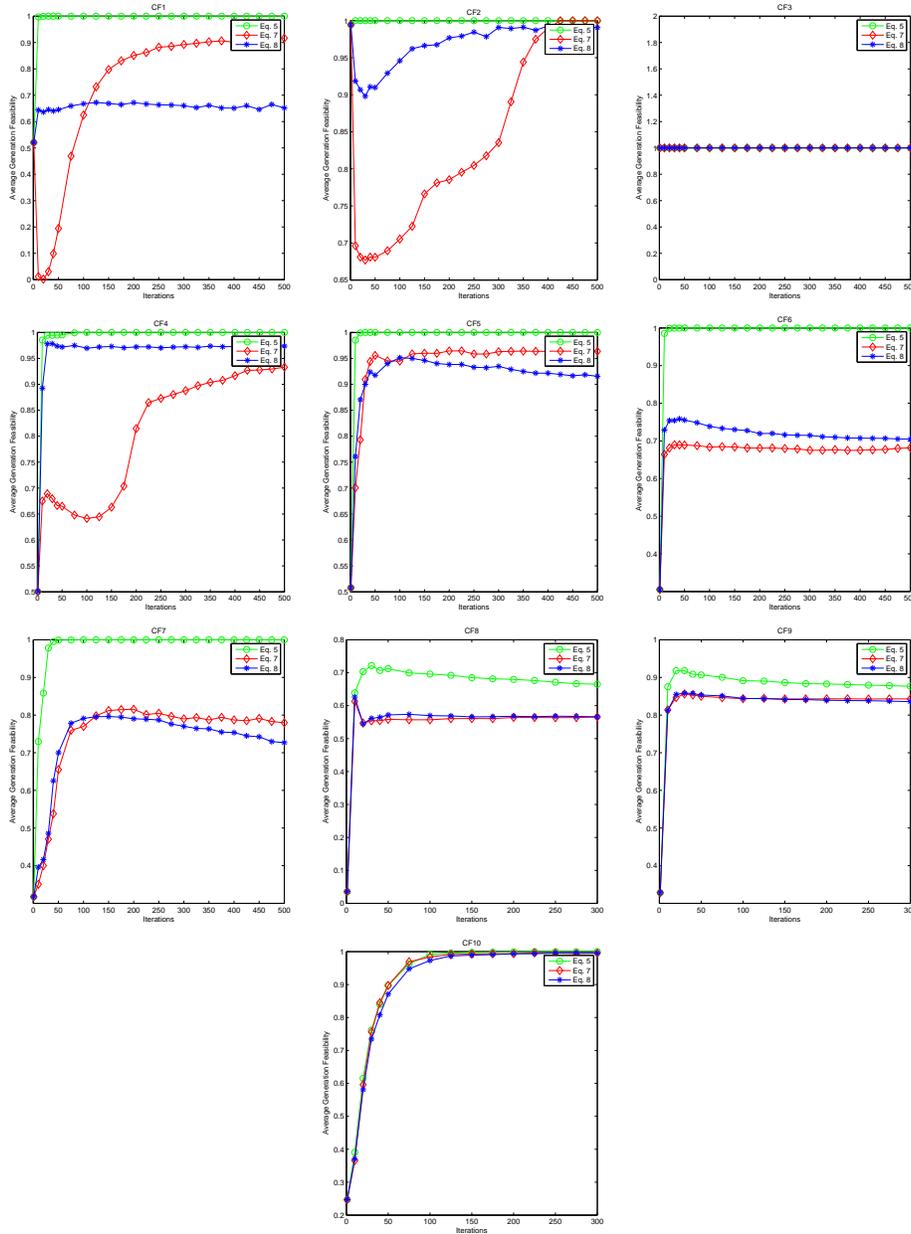


Fig. 7: Evolution of the generation feasibility versus generations when CMOEA/D-DE-TDA uses Eqs. 5, 7, and 8 for CF1-CF10.

TABLE III: COMPARISON BETWEEN CMOEA/D-DE-TDA WITH EQS. 5, 7, AND 8 (INDICATED BY JZ1, JZ2, JZ3, RESPECTIVELY), TSENG AND CHEN'S [12] (INDICATED BY TC), LIU AND LI'S [13] (INDICATED BY LL), AND LIU ET. AL'S [14] (INDICATED BY LI) ALGORITHMS IN TERMS OF THE IGD VALUES BASED ON 30 INDEPENDENT RUNS. THE RESULTS IN **BOLDFACE** INDICATE THE BEST VALUES AND THE RESULTS IN *ITALIC* INDICATE THE SECOND BEST RESULTS.

Test Instance	best (lowest)						mean						st. dev.					
	JZ1	JZ2	JZ3	TC	LL	LI	JZ1	JZ2	JZ3	TC	LL	LI	JZ1	JZ2	JZ3	TC	LL	LI
CF1	0.0003	0.0034	0.0035	0.0007	0.0071	<i>0.0005</i>	0.0005	0.0051	0.0076	0.0192	<i>0.0009</i>	0.0113	<i>0.0002</i>	0.0017	0.0027	0.0026	0.0001	0.0028
CF2	<i>0.0026</i>	0.0028	<i>0.0026</i>	0.0041	0.0027	0.0016	0.0038	<i>0.0037</i>	0.0045	0.0192	0.0009	0.0113	0.0014	<i>0.0011</i>	0.0020	0.0147	0.0026	0.0005
CF4	0.0056	0.0054	0.0062	0.0089	0.0090	<i>0.0055</i>	0.0083	<i>0.0076</i>	0.0106	0.0111	0.0142	0.0070	0.0018	0.0016	0.0041	0.0014	0.0033	<i>0.0015</i>
CF5	0.0326	0.0526	0.0328	<i>0.0176</i>	0.0588	0.0079	0.1560	0.1175	0.1356	<i>0.0208</i>	0.1097	0.0158	0.0932	0.0348	0.0921	0.0024	0.0307	<i>0.0067</i>
CF6	<i>0.0069</i>	0.0423	0.0387	0.0096	0.0090	0.0062	0.0220	0.1667	0.1241	0.0162	0.0139	<i>0.0150</i>	0.0160	0.1162	0.0303	<i>0.0060</i>	0.0026	0.0065
CF7	0.0374	0.0551	0.0556	<i>0.0187</i>	0.0535	0.0104	0.1227	0.1474	0.1567	<i>0.0247</i>	0.1045	0.0191	0.0787	0.0684	0.06590	0.0047	0.0351	<i>0.0061</i>
CF8	0.0344	<i>0.0336</i>	0.0335	0.6220	0.0473	0.0388	0.0380	<i>0.0359</i>	0.0357	1.0854	0.0607	0.0475	0.0017	<i>0.0016</i>	0.0013	0.2191	0.0130	0.0064
CF9	<i>0.0431</i>	0.0433	0.0429	0.0721	0.0460	0.1191	0.0475	<i>0.0473</i>	0.0471	0.0851	0.0505	0.1434	<i>0.0023</i>	0.0022	0.0025	0.0082	0.0034	0.0214
CF10	<i>0.1020</i>	0.1054	0.1062	0.1173	0.1055	0.0984	0.1602	0.1487	<i>0.1430</i>	0.1376	0.1974	0.1621	0.0464	0.0380	<i>0.0308</i>	0.0092	0.0760	0.0316

Session on Performance Assessment of Multiobjective Optimization Algorithms/CEC 09 MOEA Competition, Norway, 18-21 May 2009.

- [12] L. Tseng and C. Chen, "Multiple trajectory search for unconstrained/constrained multi-objective optimization," in *IEEE Congress on Evolutionary Computation, CEC2009*. IEEE, 2009, pp. 1951–1958.
- [13] H. Liu and X. Li, "The multiobjective evolutionary algorithm based on determined weight and sub-regional search," in *IEEE Congress on Evolutionary Computation, CEC2009*. IEEE, pp. 1928–1934.
- [14] M. Liu, X. Zou, Y. Chen, and 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.
- [15] K. M.iettinen, *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1999.
- [16] M. A. Jan and R. A. Khanum, "A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D," *Appl. Soft Comput.*, vol. 13, no. 1, pp. 128–148, 2013.
- [17] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," University of Essex and Nanyang Technological University, Tech. Rep. CES-487, 2008.