# Threshold Based Dynamic and Adaptive Penalty Functions for Constrained Multiobjective Optimization

Muhammad Asif Jan
Department of Mathematics
Kohat University of Science & Technology
Kohat, Pakistan
majan@kust.edu.pk; majan.math@gmail.com

Nasser Tairan
College of Computer Science
King Khalid University
Abha, Saudi Arabia
nmtairan@kku.edu.sa

Rashida Adeeb Khanum
Jinnah College for Women
University of Peshawar
Peshawar, Pakistan
adeeb_maths@yahoo.com

***Abstract*—Penalty functions are frequently used for dealing with constraints in constrained optimization. Among different types of penalty functions, dynamic and adaptive penalty functions seem effective, since the penalty coefficients in them are adjusted based on the current generation number (or number of solutions searched) and feedback from the search. In this paper, we propose dynamic and adaptive versions of our recently proposed threshold based penalty function. They are then implemented in the multiobjective evolutionary algorithm based on decomposition (MOEA/D) framework to solve constrained multiobjective optimization problems (CMOPs). This led to a new algorithm, denoted by CMOEA/D-DE-TDA. The performance of CMOEA/D-DE-TDA is tested on CTP-series test instances in terms of the HV-metric and SC-metric. The experimental results are compared with IDEA and NSGA-II, which show the effectiveness of the proposed algorithm.**

***Index Terms*—Constrained multiobjective optimization; decomposition; MOEA/D; penalty function; threshold.**

## I. Introduction

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

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

where $x = (x_1, \ldots, x_n)^T \in \mathcal{R}^n$ is an $n$ dimensional vector of decision variables, $F$ is the objective vector function that consists of $m$ 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 (called bound constraints) of $x_k$, $k = 1, \ldots, n$, respectively. A solution $x$ is called a feasible solution, if it meets all the bound and inequality constraints in (1); otherwise, it is called infeasible.

Penalty functions are commonly used for dealing with constraints. In penalty function methods, appropriate penalty coefficients are often required to balance objective and penalty functions. However, it is often very hard to find such appropriate coefficients [1]. They are problems dependent. Thus, in order to avoid the difficulty to set penalty coefficients in static penalty functions and to explore infeasible regions, some researchers in the EAs community have developed dynamic and adaptive penalty functions [2]–[4].

In [5], we proposed a novel threshold based penalty function for constrained multiobjective optimization (CMOO). The threshold dynamically controls the penalty to infeasible solutions. It is controlled by a scaling parameter. 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, we utilize a parameterless threshold value and present dynamic and adaptive versions of the proposed penalty function. They are then implemented in one of the improved frameworks of MOEA/D [6], namely MOEA/D-DE [7] to solve CMOPs.

The remainder of this paper is organized as follows. Section II presents the Tchebycheff aggregation function and the proposed threshold based dynamic and adaptive penalty functions. Section III briefly introduces MOEA/D and presents the Pseudo-code of the update scheme of CMOEA/D-DE-TDA to demonstrate the implementation of proposed penalty functions. Section IV discusses the experimental settings. Section V comments on the metrics that will be used for the performance assessment of CMOEA/D-DE-TDA . Section VI presents and discusses experimental results on CTP-series [8], [9] test instances. Section VII compares our experimental results with IDEA [10] and NSGA-II [11]. Finally, Section VIII concludes the paper.

## II. Tchebycheff Aggregation Function and the Proposed Penalty Functions

### A. Tchebycheff Aggregation Function

MOEA/D [6] needs to decompose a multiobjective optimization problem (MOP) into a number of scalar objective subproblems. In this paper, we use the Tchebycheff aggregation function for this purpose. It is defined as follows [12]:

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

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

## *B. The Proposed Penalty Functions*

Our proposed penalty function in [5] uses a threshold value, $\tau$ for dynamically controlling the amount of penalty to infeasible solutions. It is computed as follows.

Suppose MOEA/D [6] decomposes the given MOP into $N$ subproblems. At each generation, MOEA/D maintains $N$ solutions $x^1, \ldots, x^N$, where $x^i$ is the current solution to subproblem $i$. Let $P$ be the mating and update range set in MOEA/D (for details, please see Algorithm 1, Section III). Then we set [5]:

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

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

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, $\tau$ is defined as [5]:

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

where parameter $s$ controls the threshold value, $\tau$.

The proposed penalty function encourages the algorithm to search the feasible region and the infeasible region near the feasible region. It is defined as follows [5]:
For $i = 1, \ldots, m$

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

where $f_p^i(x)$ is the $i$-th penalized objective function value.

In [5], the algorithm is tested with a fixed value of parameter $s$ (we used $s = 0.3$ in Eq. 4). However, because of the changing values of $V_{min}$ and $V_{max}$ during evolution, the resulting threshold value, $\tau$ was still adaptive.

It might happen that if $V_{min}$ and $V_{max}$ are quite apart from each other, then the threshold value, $\tau$ will be greater. As a result, more infeasible solutions will be penalized, and the search might quickly focus on the feasible region. Consequently, optimal solutions of low quality might be obtained. Thus, in order to avoid such situation, in this paper, we set the threshold value, $\tau$ as the average value of the degree of constraint violations of all infeasible solutions in the neighborhood of a solution. That is,

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

where $n_{inf}$ is the number of infeasible solutions in the neighborhood of a solution. This avoids fixing different values of parameter $s$.

The penalty function defined by Eq. 5 also uses two more fixed penalty coeffients/control parameters, $s_1$ and $s_2$, where $s_1 << s_2$, to penalize infeasible solutions. Thus, infeasible solutions with degree of constraint violation less than $\tau$ are less penalized than the infeasible solutions with degree of constraint violation greater than $\tau$. However, in this paper, we adjust $s_1$ and $s_2$ dynamically and adaptively and define the penalty function in two new forms as follows:
For $i = 1, \ldots, m$

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

where $g$ is the current generation number and G is the total number of generations.
When $r_{inf} \neq 0$,

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

where $r_{inf}$ 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.

In Eq. 7, infeasible solutions are initially less penalized with small generation number, $g$. Because of the comparatively small penalty coefficients used, the infeasible solutions with degree of constraint violation less than the threshold value had particularly more chances to evolve. This provides a chance of exploring the infeasible regions well in the beginning of the search. However, as the search progresses, the penalty to infeasible solutions increases with the increase in generation number, $g$. As a result, the search focuses on the feasible region at the end of the algorithmic run.

In Eq. 8, on the other hand, the penalty to infeasible solutions depends upon the number of infeasible solutions in the neighborhood of a solution. If there are fewer infeasible solutions in the neighborhood of a solution, the penalty to neighboring infeasible solutions is small; otherwise, it increases with the increase in the number of infeasible solutions. Here again, because of the adopted penalty coefficients, the infeasible solutions with degree of constraint violation less than the threshold had more chances to evolve.

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

MOEA/D-DE [7] is an updated and efficient version of MOEA/D [6]. The framework of this algorithm is modified in [5], [13] for CMOPs. The modified framework is denoted by CMOEA/D-DE [13].

The penalty functions defined by Eqs. 5, 7, and 8 with threshold value, $\tau$ given by Eq. 6 are employed in the update scheme of CMOEA/D-DE to solve CTP-series [8], [9] test instances. This results 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.

## IV. EXPERIMENTAL SETTINGS

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

### *A. Parameters Settings:*

The parameters' settings are same as in [10], [14]. They are given as follows:

- Population size, $N = 200$;

**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**
```

- Dimensionality of variables, $n = 2$; .
- 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, $G = 200$;
- Stopping criterion: the algorithm stops after $40,000$ function evaluations.
- Number of runs: Each algorithm is run 30 times independently in MATLAB for each test instance.

### *B. Weight Vectors Selection:*

A set of $N$ weight vectors, $W$, is originated by using the following criteria [14]:

1) Form a set $W1$ comprising of uniformly randomly generated 5000 weight vectors.
2) Initialize set $W$ as the set holding all the weight vectors $(1, 0, \ldots, 0, 0)$, $(0, 1, \ldots, 0, 0)$, $\ldots$, $(0, 0, \ldots, 0, 1)$.
3) Find the weight vector in set $W1$ with the largest distance to set $W$, add it to set $W$ and delete it from set $W1$.
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.

## V. Performance Metrics

In this paper, we will use the *hypervolume* metric (HV-metric) [8] statistics for comparing results (Note that the algorithms in comparison have also used it.). It computes the volume in the function space covered by the elements of a set $S$ for problems with minimizing objectives [15].

Suppose $A_1$ and $A_2$ be two approximations to the PF of a CMOP. The *set coverage* metric (SC-metric) [6] gives the percentage of the solutions in $A_2$ that are dominated by at least one solution in $A_1$. We will employ the SC-metric to compare the nondominated solutions obtained by different algorithms.

TABLE I: Comparison of the HV-metric statistics based on 30 independent runs of CMOEA/D-DE-TDA using Eqs. 5, 7, and 8 for CTP1-CTP8. The results in **boldface** and in *italic* indicate the better and the second better results.'-' means that no feasible solution is found.

| | best (highest) | | | mean | | | st. dev. | | |
|---|---|---|---|---|---|---|---|---|---|
| Test Instance | Eq. 5 | Eq. 7 | Eq. 8 | Eq. 5 | Eq. 7 | Eq. 8 | Eq. 5 | Eq. 7 | Eq. 8 |
| CTP1 | **2.7647** | 2.5182 | *2.7151* | **2.7638** | 2.5183 | *2.5377* | *0.0014* | **0.0000*** | 0.0536 |
| CTP2 | **3.0595** | 3.0590 | *3.0594* | **3.0593** | *3.0592* | **3.0593** | 0.0001 | 0.0001 | 0.0001 |
| CTP3 | *3.0273* | 3.0145 | **3.0307** | 3.0236 | *3.0246* | **3.0267** | **0.0019** | 0.0027 | *0.0025* |
| CTP4 | *2.9638* | 2.8268 | **2.9892** | *2.9127* | 2.9069 | **2.9706** | *0.0294* | 0.0385 | **0.0121** |
| CTP5 | *3.0381* | 3.0238 | **3.0385** | *3.0323* | *3.0323* | **3.0325** | **0.0031** | *0.0032* | 0.0035 |
| CTP6 | - | - | - | - | - | - | - | - | - |
| CTP7 | **3.6127** | 3.5358 | *3.6126* | **3.6124** | *3.5768* | 3.5694 | **0.0001** | 0.0383 | *0.0382* |
| CTP8 | - | - | - | - | - | - | - | - | - |

## VI. Experimental Results

In this section, we show and compare the experimental results obtained from CMOEA/D-DE-TDA with Eqs. 5, 7, and 8 on eight CTP-series test instances. This will show the behavior of CMOEA/D-DE-TDA with static, dynamic and adaptively adjusted penalty coefficients. Moreover, for calculating the HV-metric values, the reference point used for test instances CTP6 and CTP8 is (2, 20), while for the rest of the test instances, it is (2,2).

Table I presents the HV-metric statistics based on 30 independent runs of CMOEA/D-DE-TDA with Eqs. 5, 7, and 8 for the CTP-series test instances. These statistics are based on feasible solutions found in the final populations of each algorithmic run and contain the best (i.e., highest), mean, and standard deviation values of the HV-metric.

The results of this table clearly show that CMOEA/D-DE-TDA using Eq. 5 found better statistics for test instances CTP1 and CTP7. The algorithm showed improved statistical performance with Eq. 8 for test instances CTP3 and CTP4. However, the performance of the algorithm with all three Eqs. 5, 7, and 8 is almost same for test instances CTP2 and CTP5.

Moreover, for test instances CTP6 and CTP8, CMOEA/D-DE-TDA with all three Eqs. 5, 7, and 8 fails to find any feasible solution in any of the 30 algorithmic runs. The algorithm traps in the infeasible region and finally converges to a single infeasible solution or to more than one infeasible solutions ending up on the unconstrained PF (see Fig. 1).

Table II presents the average of the SC-metric values of the final nondominated fronts obtained by CMOEA/D-DE-TDA with Eqs. 5, 7, and 8 for the CTP-series test instances.

The results of this table show that, in terms of the SC-metric, the nondominated solutions found by CMOEA/D-DE-TDA with Eqs. 5, 7 are alike for test instance CTP1. Those found with Eq. 5 are better than the ones obtained with Eq. 7 for test instances CTP2, CTP4, and CTP5, while the situation is vice versa for test instance CTP3 and CTP7. Since the SC-metric values are marginally different for test instances CTP2,

TABLE II: The average set coverage amongst Eqs. 5 (S), 7 (D), and 8 (A) used by the algorithm on CTP-series test instances. The results in **boldface** indicate the better results; if not, they are identical.‘-’ means that no feasible solution is found.

| Test Instance | C(S, D) | C(D, S) | C(S, A) | C(A, S) | C(D, A) | C(A, D) |
|---|---|---|---|---|---|---|
| CTP1 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| CTP2 | **0.26** | 0.23 | 0.26 | **0.28** | 0.23 | **0.29** |
| CTP3 | 0.43 | **0.56** | 0.36 | **0.63** | 0.40 | **0.59** |
| CTP4 | **0.43** | 0.31 | 0.24 | **0.62** | 0.26 | **0.59** |
| CTP5 | **0.44** | 0.39 | 0.42 | 0.42 | 0.40 | **0.44** |
| CTP6 | - | - | - | - | - | - |
| CTP7 | 0.05 | **0.07** | 0.05 | 0.05 | 0.06 | 0.06 |
| CTP8 | - | - | - | - | - | - |

CTP5, and CTP7, the nondominated solutions found may be considered as identical for these test instances.

A comparison of the SC-metric values obtained from the algorithm with Eqs. 5, 8 in Table II unveils that the final nondominated solutions are the same for test instances CTP1, CTP5 and CTP7. However, those obtained with Eq. 8 are superior than the ones acquired with Eq. 5 for test instance CTP2, CTP3, and CTP4. In particular, a clear difference in the SC-metric values can be witnessed in the latter two test instances.

Table II also shows that the nondominated solutions found by CMOEA/D-DE-TDA with Eqs. 7, 8 are identical for test instances CTP1 and CTP7. However, the ones achieved with Eq. 8 are better than those attained with Eq. 7 for test instances CTP2-CTP5.

Fig. 1 plots, in the objective space, the nondominated solutions with the best (i.e., highest) HV-metric value found by CMOEA/D-DE-TDA in 30 independent runs when using Eqs. 5, 7, and 8 for CTP-series test instances.

The figure clearly show that CMOEA/D-DE-TDA with all three Eqs. 5, 7, and 8 found good approximations of the PFs for five test instances CTP2-CTP5, and CTP7.

The algorithm approximated well the PF of test instance CTP1 with Eq. 5, but failed with Eqs. 7, 8. More than 90% of the initial population members are feasible for this test instance (see Fig. 3 for generation feasibility graph of CTP1). Thus, due to the presence of few infeasible solutions, the resulted threshold values in the neighborhood of different solutions might be small as well. These small threshold values and the less penalty assigned with small generation number, $g$ in the early generations of the algorithm with Eq. 7 provided a chance to these few infeasible solutions to evolve and produce more infeasible solutions. As a result, the generation feasibility is reduced to 0.5 by generation 10 and onwards (see Fig. 3). This means that the infeasible solutions are unable to beat their feasible counterparts in terms of the aggregation function values. Consequently, the population converges to the upper part of the PF. It then becomes difficult for the algorithm to diversify the population along the whole PF adopting mutation operator only. This happened in all the 30 independent algorithmic runs which resulted in a standard deviation value of 0.0000. Thus, this better standard deviation value obtained with Eq. 7 and listed in Table I is not considerable in comparison to the other two obtained with Eqs. 5, 8. Because the algorithm with Eq. 8

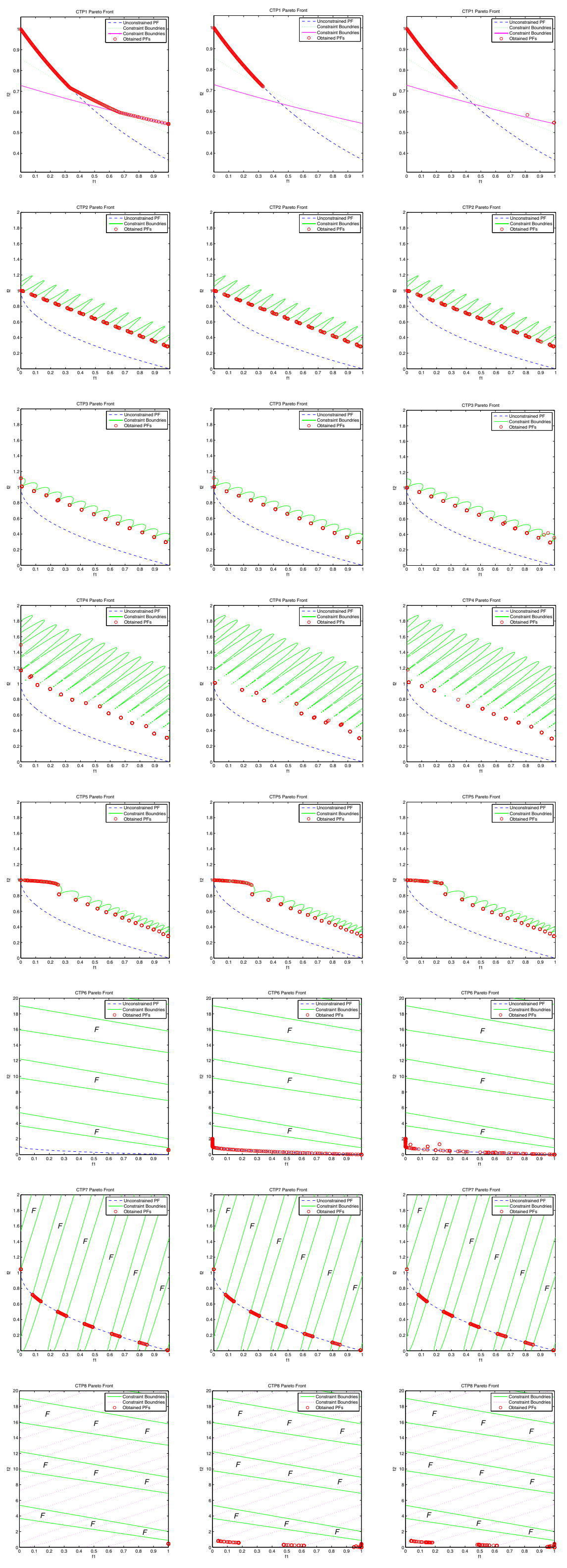

Fig. 1: Plots of the nondominated front with the best HV-metric value found by the algorithm with Eq. 5 (left column), Eq. 7 (middle column), and Eq. 8 (right column) for CTP1-CTP8.

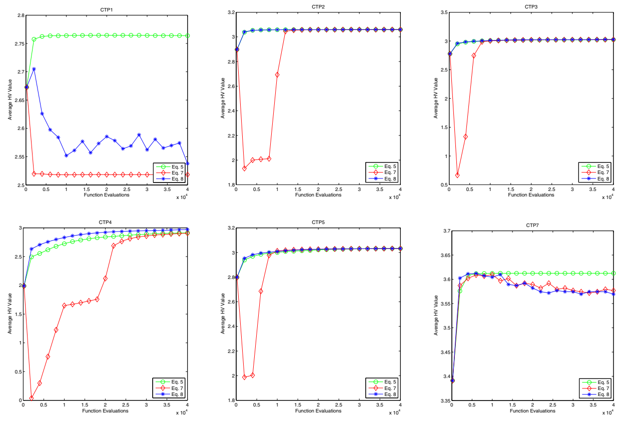

Fig. 2: Evolution of the HV-metric values versus function evaluations when the algorithm uses Eqs. 5, 7, and 8 for CTP1-CTP5 and CTP7.

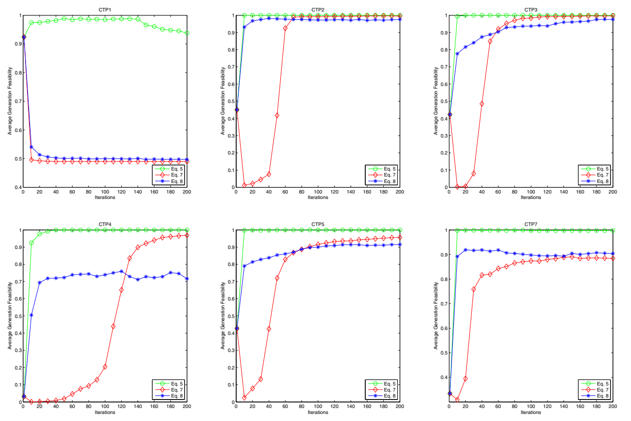

Fig. 3: Evolution of the generation feasibility versus generations when the algorithm uses Eqs. 5, 7, and 8 for CTP1-CTP5 and CTP7.

assigns less penalty to infeasible solutions in the presence of few infeasible solutions, same reasoning is also true for the failure of the algorithm with this equation as well, particularly for the upper part of the PF (see again Fig. 1).

Fig. 1 also shows that CMOEA/D-DE-TDA failed in approximating the PFs of test instance CTP6 and CTP8 with all three Eqs. 5, 7, and 8. It converged to a single infeasible solution in case of Eq. 5 and to more than one infeasible solutions in case of Eqs. 7, 8 for these two test instances. These two test instances contain hard constraints that split the search space into feasible and infeasible bands of varying widths and cause highly infeasible initial populations. The latter factor leads to a high threshold value. As a result of adopting this high threshold value and proposed penalty coefficients, CMOEA/D-DE-TDA with all three Eqs. 5, 7, and 8 gets trapped in the infeasible region. Moreover, due to the replacement and update scheme of the algorithm, these infeasible solutions eventually converge to a single infeasible solution in case of Eq. 5 and to the unconstrained PFs in case of Eqs. 7, 8.

Figs. 2 and 3 show the evolution of the average HV-metric values versus function evaluations and average generation feasibility versus generations in CMOEA/D-DE-TDA when using Eqs. 5, 7, and 8 on six CTP-series test instances.

It is clear from Fig. 2 that CMOEA/D-DE-TDA with all three Eqs. 5, 7, and 8 can find same final average HV-metric values on test instances CTP2, CTP3 and CTP5. The algorithm can find better HV-metric values with Eqs. 5 than those obtained with the other two Eqs. 7, 8 for test instances CTP1 and CTP7. Fig. 2 also shows that the HV-metric values of the algorithm with Eq. 8 are superior than the ones attained with Eqs. 5, 7 on test instance CTP4.

Fig. 3 shows that most of the population members of CMOEA/D-DE-TDA using Eq. 5 become feasible quickly, and the feasibility ratio gets equal to 1 mostly by generations 10 to 20 for test instances CTP2-CTP5, and CTP7. However, it takes long to get a feasibility ratio either equal to 1 or close to 1 with Eqs. 7, 8 for these test instances. These figures also show that the feasibility ratio of the algorithm with Eq. 8 remains low than with Eq. 7 for all test instances except CTP1 and CTP7.

The exceptional case is test instance CTP1. The feasibility ratio of CMOEA/D-DE-TDA with Eqs. 7, 8 remains very smaller than with Eq. 5 right from the beginning of the algorithmic runs even though the algorithm starts with 92% feasible solutions with all three equations. The final populations in CMOEA/D-DE-TDA had on average 50% or slightly more feasible solutions when using Eqs. 7, 8. The existence of more infeasible solutions in the final populations is one of the reasons for the poor performance of the algorithm on test instance CTP1 with these two equations.

Since the algorithm fails to find any feasible solutions with all the three Eqs. 5, 7, and 8 for test instances CTP6 and CTP8, their graphs are not presented.

## VII. Comparison with IDEA and NSGA-II

In this section, the results of CMOEA/D-DE-TDA are compared with those of IDEA [10] and NSGA-II [11] results available in [10] on the seven CTP-series test instances, CTP2-CTP8.

Table III compares the HV-metric statistics, across all 30 runs, obtained from our algorithm, CMOEA/D-DE-TDA with Eqs. 5, 7, and 8, IDEA with $\alpha = 0.2$ [10] ($\alpha$ determines the percentage of infeasible solutions to be retained during the course of evolution) and NSGA-II with constraint domination principle [11]. The statistics for the algorithms IDEA 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 IDEA and NSGA-II. All other parameters remain the same as in Section IV in our experiments.

It is obvious from this table results that our algorithm has found significantly better standard deviation values than IDEA and NSGA-II for five test instances, CTP2-CTP5 and CTP7.

TABLE III: Comparison between CMOEA/D-DE-TDA with Eqs. 5, 7, and 8 (indicated by JZ1, JZ2, JZ3, respectively), 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 second better results. '-' means that no feasible solution is found.

| | best (Highest) | | | | | mean | | | | | st. dev. | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Test Instance | JZ1 | JZ2 | JZ3 | IDEA | NSGA-II | JZ1 | JZ2 | JZ3 | IDEA | NSGA-II | JZ1 | JZ2 | JZ3 | IDEA | NSGA-II |
| CTP2 | **3.0595** | 3.0590 | *3.0594* | 3.0592 | 3.0593 | **3.0593** | *3.0592* | **3.0593** | 3.0114 | 2.8707 | **0.0001** | **0.0001** | **0.0001** | *0.1771* | 0.2701 |
| CTP3 | *3.0273* | 3.0145 | **3.0307** | 3.0160 | 3.0104 | 3.0236 | *3.0246* | **3.0267** | 2.9608 | 2.8281 | **0.0019** | 0.0027 | *0.0025* | 0.1638 | 0.2547 |
| CTP4 | *2.9638* | 2.8268 | **2.9892** | 2.9190 | 2.8485 | *2.9127* | 2.9069 | **2.9706** | 2.7447 | 2.4381 | **0.0294** | 0.0385 | **0.0121** | 0.1393 | 0.3527 |
| CTP5 | *3.0381* | 3.0238 | **3.0385** | 3.0247 | 3.0209 | *3.0323* | *3.0323* | **3.0325** | 2.9529 | 2.7235 | **0.0031** | *0.0032* | 0.0035 | 0.1621 | 0.2926 |
| CTP6 | - | - | - | *36.8191* | **36.8227** | - | - | - | **36.7878** | *36.1829* | - | - | - | **0.0758** | *2.1873* |
| CTP7 | *3.6127* | 3.5358 | 3.6126 | **3.6177** | **3.6177** | **3.6124** | *3.5768* | 3.5694 | 3.4359 | 3.2402 | **0.0001** | 0.0383 | *0.0382* | 0.5945 | 0.5941 |
| CTP8 | - | - | - | **36.1804** | *36.1708* | - | - | - | **35.9706** | *32.0859* | - | - | - | **0.4345** | *5.1763* |

These small values of the standard deviation suggest that the performance of our algorithm is consistent. The best and mean HV-metric values obtained by our algorithm with all three Eqs. 5, 7, and 8 are also better than the two competitors except the best value of CTP7 on these five test instances.

Because our algorithm gets trapped in the infeasible region with all three Eqs. 5, 7, and 8 and converged to a single infeasible solution or more infeasible solutions for test instances CTP6 and CTP8, we do not present any of our statistics for these two test instances. However, for these two test instances, IDEA found better statistics than NSGA-II except the best value for CTP6.

## VIII. Conclusions

In this paper, we run CMOEA/D-DE-TDA with dynamic and adaptive versions of the penalty function proposed in Eq 5. We can conclude the following points from our experiments conducted in this paper.

Firstly, our algorithm with all three penalty functions works well even if there are few feasible solutions in the initial population. Secondly, it with all three penalty functions fails totally if the whole initial population is highly infeasible due to the presence of hard constraints. Thirdly, our algorithm with the dynamic and adaptive penalty functions fails partly if there are few infeasible solutions in the initial population. Fourthly, the comparison of CMOEA/D-DE-TDA with IDEA and NSGA-II unveiled that our algorithm can find better and consistent statistics, in terms of the HV-metric, than the two contestants for five CTP-series test instances, CTP2-CTP5, and CTP7.

In the near future, we intend to run CMOEA/D-DE-TDA on CF-series test instances and compare our results with the three best algorithms of CEC2009 MOEA competition. We will also check the sensitivity of CMOEA/D-DE-TDA to the parameters involved.

## References

[1] 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.

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

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

[4] 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., Amsterdan, The Netherlands. Heidelberg, Germany: Springer-Verlag, September 1998, pp. 211–220, lecture Notes in Computer Science Vol. 1498.

[5] 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.

[6] 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.

[7] 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.

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

[9] K. Deb, A. Pratap, and T. Meyarivan, "Constrained test problems for multi-objective evolutionary optimization," in *First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, ser. LNCS, vol. 1993. Zurich, Switzerland: Springer, 2001, pp. 284–298.

[10] T. Ray, H. K. Singh, A. Isaacs, and W. Smith, "Infeasibility driven evolutionary algorithm for constrained optimization," in *Constraint-Handling in Evolutionary Computation*, E. Mezura-Montes, Ed. Berlin: Springer. Studies in Computational Intelligence, Volume 198, 2009, ch. 7, pp. 145–165, ISBN 978-3-642-00618-0.

[11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, April 2002.

[12] K. M. Miettinen, *Nonlinear Multiobjective Optimization.* Kluwer Academic Publishers, 1999.

[13] M. A. Jan, "Decomposition Based Evolutionary Methods for Constrained Multiobjective Optimization," Ph.D. dissertation, University of Essex, 2011.

[14] Q. Zhang, W. Liu, and 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, 18-21 May 2009.

[15] D. Van Veldhuizen, "Multiobjective evolutionary algorithms: classifications, analyses, and new innovations," in *Evolutionary Computation*. Citeseer, 1999.