Performance of a Constrained Version of MOEA/D on CTP-series Test Instances

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

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

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

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

Abstract—**Constrained multiobjective optimization arises in many real-life applications, and is therefore gaining a constantly growing attention of the researchers. Constraint handling techniques differ in the way infeasible solutions are evolved in the evolutionary process along with their feasible counterparts. Our recently proposed threshold based penalty function gives a chance of evolution to infeasible solutions whose constraint violation is less than a specified threshold value. This paper embeds the threshold based penalty function in the update and replacement scheme of multi-objective evolutionary algorithm based on decomposition (MOEA/D) to find tradeoff solutions for constrained multiobjective optimization problems (CMOPs). The modified algorithm is tested on CTP-series test instances in terms of the hypervolume metric (HV-metric). The experimental results are compared with the two well-known algorithms, NSGA-II and IDEA. The sensitivity of algorithm to the adopted parameters is also checked. Empirical results demonstrate the effectiveness of the proposed penalty function in the MOEA/D framework for CMOPs.**

Keywords—*Decomposition; MOEA/D; threshold based penalty function; constrained multiobjective optimization*

I. INTRODUCTION

This paper handles following type of constrained multiobjective optimization problem (CMOP) [1] [2]:

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

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 real-valued objective functions, and $g_i(x) \geq 0$ are inequality constraints (equality constraints, if any, can be transformed into corresponding inequality constraint as per common practice in literature). 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, which define the search region $\mathcal{S} = \{x = (x_1, \ldots, x_n)^T \mid l_k \leq x_k \leq u_k, k = 1, \ldots, n\}$.

A solution $x \in S$ is called a feasible solution, if it satisfies all the inequality constraints in (1). The set of all feasible solutions is called the feasible region. Mathematically, we can write: $\mathcal{F} = \{x \in \mathcal{S} \subset \mathcal{R}^n \mid g_j(x) \geq 0, j = 1, \cdots, p\}$. However, if a solution is not feasible, we call it infeasible. The set of all infeasible solutions is called the infeasible region. Moreover, the feasible attainable objective set (AOS) can be defined as $\{F(x) \mid x \in \mathcal{F}\}$.

In CMOPs, due to conflicting objectives, a single solution that could optimize all the objectives at the same time is often hard to be found. Therefore, in such problems, normally one has to look for a set of feasible optimal compromising/tradeoff solutions. Pareto-optimality [3], [4] defines the best tradeoffs among the conflicting objectives.

A solution x Pareto-dominates or simply dominates another solution y, mathematically denoted as $x \preceq y$, if $f_i(x) \leq f_i(y)$, $\forall i = 1, \ldots, m$ and $f_j(x) < f_j(y)$ for at least one $j \in \{1, \ldots, m\}^1$. 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 [3].

In most of the constrained optimization problems, the Pareto-optimal solutions lie on the constraints' boundaries Thus, to locate such solutions, a good practice could be to evolve some good infeasible solutions with small degree of constraint violation along with the feasible solutions during the evolutionary process (In [5]–[7] such practice has been adopted). The main goal of evolving infeasible solutions in the search process is to use the information they carry. Since EAs are stochastic search and optimization methods, so ignoring infeasible solutions might guide the EA being stuck in local optima, especially in CMOPs with disconnected search space [8], [9]. Additionally, finding a single feasible solution in some highly constrained optimization problems by itself could be a challenging problem [10], [11]. Therefore, one

---

[1]All the inequalities should be reversed if the purpose is to maximize the objectives in (1)

can distinguish different constraint handling techniques that are used in multiobjective optimization (MOO) by the way the infeasible solutions are mixed up and evolved with their feasible counterparts during the evolutionary process.

In [1], a threshold based penalty function is introduced. In order to penalize infeasible solutions, this penalty function uses a threshold value. The infeasible solutions with degree of constraint violation smaller than the threshold value are less penalized as compared to the ones with degree constraint violation greater than the threshold value. Consequently, a chance is given to some good infeasible solutions with less degree of constraint violation to evolve during the evolutionary process. In [12], the framework of MOEA/D-DE [13], an improved variant of MOEA/D [14] is modified for CMOPs; The modified framework is denoted by CMOEA/D-DE [12]. Two penalty parameterless constraint handling techniques are employed in this framework to solve CTP-series [4], [15] and CF-series [16] test instances. However, in this work, the threshold based penalty function is implanted in the replacement and update scheme of CMOEA/D-DE for handling constraints in CMOPs. As a result, a constrained version of MOEA/D-DE, denoted by CMOEA/D-DE-ATP is introduced. Empirical results have shown the capability of the proposed algorithm for handling hard CTP-series [4], [15] test instances.

The remainder of this paper is organized as follows. Section II presents a few basic concepts and the suggested threshold based penalty function. Section III introduces MOEA/D and adapts the algorithmic framework of MOEA/D-DE for CMOPs. Section IV presents and discusses experimental results on CTP-series [4], [15] test instances. This section also compares our experimental results with those of IDEA [7] and NSGA-II [17]. Section V provides some remarks on the sensitivity of the performance of CMOEA/D-DE-ATP to its parameters. Finally, Section VI concludes this paper.

II. BASIC CONCEPTS AND THE PROPOSED THRESHOLD BASED PENALTY FUNCTION

A. Degree of Constraint Violation

The degree of constraint violation of a solution $x \in S$ can be defined as follows [1], [4]:

$$V(x) = |\sum_{j=1}^{p} \min(g_j(x), 0)|. \qquad (2)$$

Clearly, $V(x) = 0$ implies that x is feasible; otherwise, it is infeasible.

B. Tchebycheff Aggregation Function

MOEA/D [14] decomposes an MOP into a number of scalar objective subproblems. In this paper, the Tchebycheff aggregation function is employed for this purpose, since it is less sensitive to the shape of PF, and it can be utilized to obtain the Pareto-optimal solutions in both convex and nonconvex PFs. It is defined as follows [18]:

$$\text{Minimize} \quad g^{te}(x|\lambda, z^*) = \max_{1 \le i \le m}\{\lambda_i|f_i(x) - z_i^*|\}; (3)$$
$$\text{Subject to} \quad x \in \mathcal{F} \subset \mathcal{R}^n;$$

where $z^* = (z_1^*, \ldots, z_m^*)^T$ 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 $\lambda_i \ge 0 \ \forall i = 1, \ldots, m$ and $\sum_{i=1}^{m} \lambda_i = 1$.

C. The Threshold Based Penalty Function

Let us consider that MOEA/D [14] decomposes the MOP into N subproblems. In each iteration, MOEA/D retains N solutions $x^1, \ldots, x^N$, where x^i is the current solution to subproblem i. Let us consider P to be the mating and update range in MOEA/D. Then define [1] [2]:

$$V_{min} = \min\{V(x^i), i \in P\}, \qquad (4)$$

$$V_{max} = \max\{V(x^i), i \in P\}, \qquad (5)$$

where $V(x^i)$ is the degree of constraint violation of solution x^i. The adopted threshold value, τ is then defined as follows [1] [2]:

$$\tau = V_{min} + s(V_{max} - V_{min}), \qquad (6)$$

where the parameter s controls τ. In [1], we experimented with $s = 0.3$.

The threshold based penalty function not only encourages the algorithm to search the feasible region, but also the infeasible region close the feasible region for optimal solutions. It is defined as follows [2]: For $i = 1, \ldots, m$

$$f_p^i(x) = \begin{cases} f_i(x) + s_1 V^2(x), & \text{if } V(x) < \tau; \\ f_i(x) + s_1\tau^2 + \\ s_2(V(x) - \tau), & \text{otherwise,} \end{cases} \qquad (7)$$

where s_1 and s_2 are two scaling parameters with $s_1 << s_2$. As can be observed from the penalty function, the penalty sharply increases when $V(x)$ is greater than the threshold. This is realized by scaling the degree of constraint violation, $V(x)$ of an infeasible solution by a relatively high value of parameter s_2 than parameter s_1.

III. MULTIOBJECTIVE EVOLUTIONARY ALGORITHM BASED ON DECOMPOSITION

Zhang and Li [14] introduced a simple yet efficient MOEA, named as multi-objective evolutionary algorithm based on decomposition (MOEA/D). In order to approximate the PF, MOEA/D employs an aggregation function to explicitly decompose an MOP into a number of scalar objective optimization subproblems (this work uses the Tchebycheff function to serve the purpose of decomposition). Then, an EA is employed to optimize these subproblems concurrently and collaboratively by evolving population of solutions. The Euclidean distances between the aggregation coefficient vectors of the subproblems are utilized to determine the neighborhood relations among them, which are then used to optimize a subproblem.

An improved version of MOEA/D, MOEA/D-DE is introduced in [13]. It distinguishes from its predecessor (MOEA/D) in the following aspects [13]:

- For maintaining population diversity, MOEA/D-DE takes two measures:
  1) It picks out three parent solutions from the whole population with a low probability $1 - \delta$,

where $\delta = 0.9$ is the probability of selecting the parents from the neighborhood of a solution. Because of the dissimilarity among these parent solutions, a very wide range of child solutions could be brought about and, as a result, the exploration ability of the search could be ameliorated.

2) It limits the maximal number of solutions replaced by a better child solution by a parameter $n_r << T$, where T is the neighborhood size of the child solution. Thus, there is a little chance that a better child solution has many copies in the population. While, in MOEA/D, the maximal number of solutions replaced by a better child solution could be as large as T.

- Since DE [19] operators often surpass other genetic operators in single objective optimization, MOEA/D-DE benefits from DE operator for generating offspring.

The penalty function defined by Eq. 7 is employed in the update scheme of CMOEA/D-DE [12] to solve CTP-series [4], [15] test instances. This produced a new algorithm, denoted by CMOEA/D-DE-ATP. The pseudo-code of the update scheme of CMOEA/D-DE-ATP is given in Algorithm 1.

Algorithm 1 Pseudo-code of the Update Scheme of CMOEA/D-DE-ATP. n_r is the Number of Solutions Updated by a Better Child Solution.

1: Each new offspring y replaces n_r parent 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 $P = \emptyset$ **then**
4:      return;
5: **else**
6:      Choose randomly an index j from P;
7:      Calculate the Tchebycheff aggregation function values of y and x^j with the new objective values of Eq. 7;
8:      **if** $g^{te}(y|\lambda^j, z) \leq g^{te}(x^j|\lambda^j, z)$ (or $g_p^{te}(y|\lambda^j, z) \leq g_p^{te}(x^j|\lambda^j, z)$) **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**

IV. EXPERIMENTAL RESULTS

In all experiments, the same parameters' settings as given in [12] are used. The weight vectors used in Eq. 3 are selected as per the proposed criteria in [16]. If not stated otherwise, Eq. 6 is used with $s = 0.3$ and Eq. 7 with $s_1 = 0.01$ and $s_2 = 20$ in all our experiments.

Hypervolume metric (HV-metric) computes the volume in the function space covered by the elements of a set P for problems with minimizing objectives [20], [21]. The HV-metric statistics are used to compare the experimental results obtained from CMOEA/D-DE-ATP on CTP-series test instances. To calculate the HV-metric values, the reference point (2, 20) is used for test instances CTP6 and CTP8, while it is (2,2) for the remaining test instances, CTP1-CTP5, CTP7.

Table I presents the HV-metric statistics based on 30 independent runs of CMOEA/D-DE-ATP when using Eq. 6 with $s = 0.3, 0.5, 0.7$ for the CTP-series test instances (the three different adopted values are intended to look at the behavior of the algorithm by evolving various percentages of infeasible solutions during the evolutionary process). 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.

For CTP1, the better best value is found by CMOEA/D-DE-ATP with all used parameter s values. However, for better mean and standard deviation values, it needs to be run with $s = 0.3$. For CTP2, the better best and mean values are found by CMOEA/D-DE-ATP with $s = 0.3$, while the better standard deviation value is found with $s = 0.5$. For CTP3, the better best value is found by CMOEA/D-DE-ATP with $s = 0.5$. However, for better mean and standard deviation values, it needs to be run with $s = 0.3$. For CTP4, CTP5, and CTP6, the better best, mean and standard deviation values are found with $s = 0.5$, $s = 0.3$, and $s = 0.7$, respectively. For CTP7 and CTP8, the better best value is found with $s = 0.7$, while the better mean and standard deviation values are found with $s = 0.3, 0.5$. Overall, better performance of the algorithm can be achieved, in terms of mean and standard deviation values, with $s = 0.3$.

Figure 1 plots, in the objective space, the nondominated solutions with the best (i.e., highest) HV-metric value found by CMOEA/D-DE-ATP in 30 independent runs for CTP-series test instances. In order to look at the variation of the nondominated solutions found by CMOEA/D-DE-ATP, all the 30 final nondominated fronts attained are also plotted in the same figure. This figure clearly shows that CMOEA/D-DE-ATP found good approximations of the PFs for all test instances except CTP4, where it can be noted that in some of the runs the obtained Pareto optimal solutions did not completely converge to the PF.

A. Comparison with NSGA-II and IDEA

This section compares the experimental results of CMOEA/D-DE-ATP with those of IDEA [7] and NSGA-II [17] on CTP-series test instances. Table II presents the HV-metric statistics, obtained from 30 independent runs of CMOEA/D-DE-ATP when using Eq. 6 with $s = 0.3$ and Eq. 7 with $s_1 = 0.01$ and $s_2 = 20$, IDEA with $\alpha = 0.2$ (α determines the percentage of infeasible solutions to be retained during evolution and NSGA-II with the constraint domination principle [17] for seven CTP-series test instances, CTP2-CTP8. The statistics of IDEA and NSGA-II are picked from [7], and are rounded to four decimal places. From this table, it can be seen that CMOEA/D-DE-ATP has found best HV-metric values for four test instances, CTP3, CTP5, CTP6 and CTP8 and the second best results for CTP4 and CTP7. NSGA-II and IDEA have found the best HV-metric value for CTP1 and CTP4, respectively, while for CTP7 both algorithms have obtained the same best HV-metric value. The table also shows that even with a small threshold value, CMOEA/D-DE-ATP has found better mean and standard deviation values

TABLE I: The HV-Metric statistics of CMOEA/D-DE-ATP with three different values of parameter s. The results in **boldface** and in *italic* show the better and the second better results. If not, they are same.

	best (highest)			mean			st. dev.		
Test Instance	$s = 0.3$	$s = 0.5$	$s = 0.7$	$s = 0.3$	$s = 0.5$	$s = 0.7$	$s = 0.3$	$s = 0.5$	$s = 0.7$
CTP1	2.7647	2.7647	2.7647	**2.7642**	*2.7638*	2.7615	**0.0006**	*0.0007*	0.0042
CTP2	**3.0591**	3.0564	*3.0580*	**3.0511**	*3.0492*	3.0477	*0.0174*	**0.0169**	0.0177
CTP3	3.0256	**3.0265**	*3.0257*	**3.0137**	*3.0012*	2.9816	**0.0242**	*0.0394*	0.0481
CTP4	2.8688	**2.9327**	*2.8778*	2.4736	**2.5531**	*2.5129*	*0.2058*	**0.2001**	0.2297
CTP5	**3.0313**	*3.0292*	3.0283	**3.0032**	3.0011	*3.0014*	**0.0201**	0.0226	**0.0201**
CTP6	36.8196	*36.8200*	**36.8202**	36.8184	*36.8181*	**36.8184**	*0.0017*	0.0027	**0.0012**
CTP7	*3.6125*	*3.6125*	**3.6128**	**3.6124**	**3.6124**	*3.6099*	**0.0001**	**0.0001**	*0.0138*
CTP8	36.1821	*36.1822*	**36.1825**	**36.1652**	36.1449	*36.1464*	**0.0342**	*0.0532*	0.0578
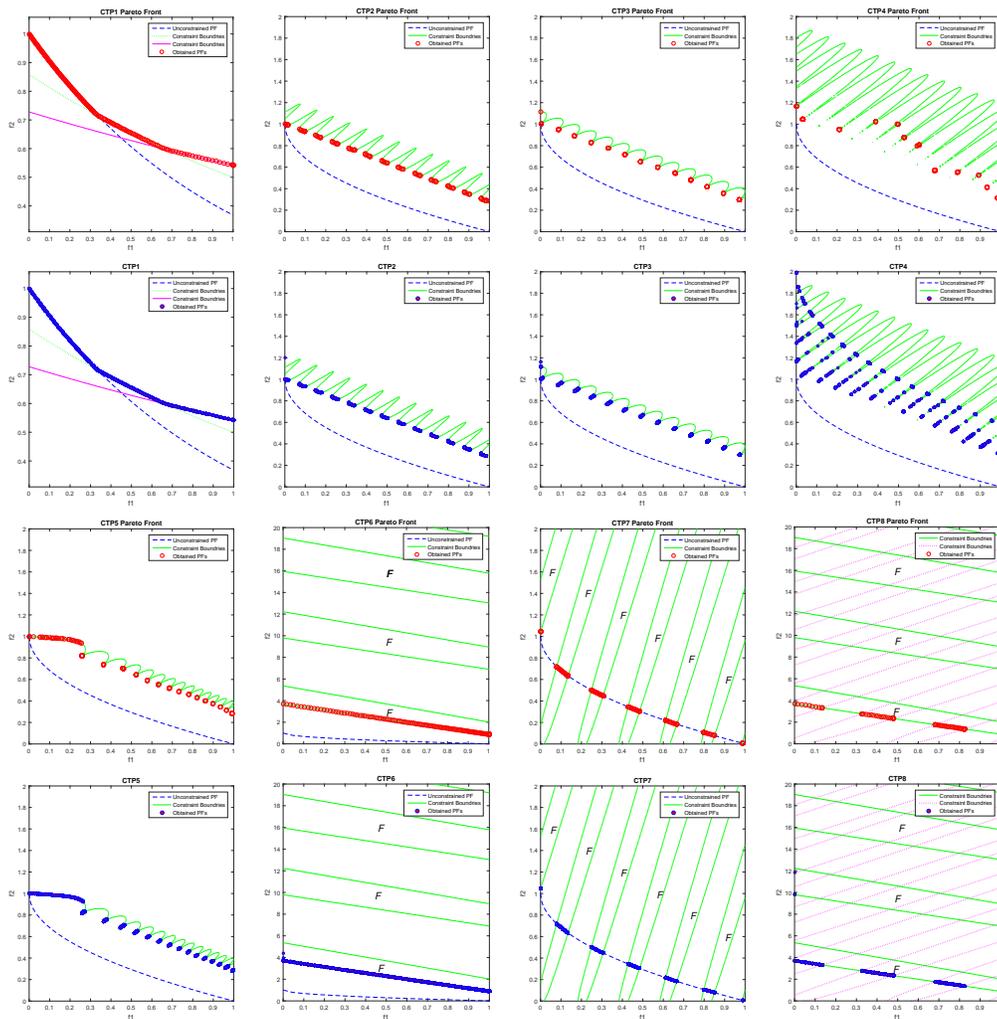
Fig. 1: Plots of the nondominated front with the best HV-metric value and all the 30 final nondominated fronts found by CMOEA/D-DE-ATP for CTP1 to CTP8.

TABLE II: Comparison of CMOEA/D-DE-ATP (indicated by ATP) with IDEA and NSGA-II in terms of the HV-Metric statistics. The results in **boldface** and in *italic* indicate the better and the second better results.

Test Instance	best (Highest)			mean			st. dev.		
	ATP	IDEA	NSGA-II	ATP	IDEA	NSGA-II	ATP	IDEA	NSGA-II
CTP2	3.0591	*3.0592*	**3.0593**	**3.0511**	3.0114	2.8707	**0.0174**	*0.1771*	0.2701
CTP3	**3.0256**	*3.0160*	3.0104	**3.0137**	*2.9608*	2.8281	**0.0242**	*0.1638*	0.2547
CTP4	*2.8688*	**2.9190**	2.8485	*2.4736*	**2.7447**	2.4381	*0.2058*	**0.1393**	0.3527
CTP5	**3.0313**	*3.0247*	3.0209	**3.0032**	*2.9529*	2.7235	**0.0201**	*0.1621*	0.2926
CTP6	**36.8196**	36.8191	*36.8227*	**36.8184**	*36.7878*	36.1829	**0.0017**	*0.0758*	2.1873
CTP7	*3.6125*	**3.6177**	**3.6177**	**3.6124**	*3.4359*	3.2402	**0.0001**	0.5945	*0.5941*
CTP8	**36.1821**	*36.1804*	36.1708	**36.1652**	*35.9706*	32.0859	**0.0342**	*0.4345*	5.1763

than IDEA and NSGA-II for all test instances except CTP4, where IDEA supersede the two competitors. In particular, the standard deviation values obtained with CMOEA/D-DE-ATP are considerably better than the two algorithms. The small values of the standard deviation for CMOEA/D-DE-ATP suggests the consistent performance of it.

V. SENSITIVITY OF CMOEA/D-DE-ATP TO ITS PARAMETERS

This section analyzes the sensitivity of the the performance of CMOEA/D-DE-ATP to the proposed penalty function parameters – s, s_1 and s_2 and the algorithmic parameters– T, n_r, δ, N, CR, and F. In this regard, a comparatively simple test instance CTP2 and a hard test instance CTP4 are used. In CTP2, the PF consists of disconnected continuous Pareto-optimal regions, while in CTP4, an algorithm has to travel through long narrow feasible tunnels to find the lone Pareto-optimal solutions at the end of each tunnel.

Fig. 2: The average HV-metric values vs. parameter s in CTP2 and CTP4.

A. Sensitivity of CMOEA/D-DE-ATP to s

To study the sensitivity of the performance of CMOEA/D-DE-ATP to parameter s that controls the threshold, τ range where infeasible solutions are less penalized, 9 different values

of parameter s (i.e., $s = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$) are used in CMOEA/D-DE-ATP for test instances CTP2 and CTP4. Figure 2 shows the average HV-metric values versus the values of parameter s. The figure clearly indicates the sensitivity of the performance of the algorithm to parameter s on different test instances. As can be seen from Figure 2 (a) that for test instance CTP2 the best (i.e., highest) average value of the HV-metric is found by CMOEA/D-DE-ATP with $s = 0.3$ (one can also see the value obtained with $s = 0.8$ is very close to the one obtained with $s = 0.3$), while for test instance CTP4, it is found with $s = 0.9$ (see Figure 2 (b)). This indicates that for test instances like CTP2 a small value of parameter s could be used to get a small value of τ and, as a result, a small infeasible region close to feasibility boundary is explored. This small value of τ allows a small number of infeasible solutions to survive, which results in achieving a higher HV-metric value. However, for hard test instances like CTP4, a higher value of parameter s could be used to get a larger value of τ and, thereby, a large infeasible region is explored along with the feasible ones in search of the lone Pareto-optimal solutions situated at the end of each disconnected narrow feasible regions. This allows a large number of infeasible solutions to be less penalized. As a result, these infeasible solutions will evolve along with feasible solutions found and thus result in obtaining a higher HV-metric value.

B. Sensitivity of CMOEA/D-DE-ATP to s_1 and s_2

To study the sensitivity of the performance of CMOEA/D-DE-ATP to parameters s_1 and s_2, 25 different combinations of five values of s_1 (i.e., $s_1 = 0.0001, 0.001, 0.01, 0.1, 0.5$) and five values of s_2 (i.e., $s_2 = 10, 20, 30, 40, 50$) are tested in CMOEA/D-DE-ATP on test instances CTP2 and CTP4. Each combination of the two parameters is tried 30 times.

Figure 3 shows the average HV-metric values obtained by the algorithm with 25 different combinations of s_1 and s_2. It is obvious that CMOEA/D-DE-ATP is less sensitive to the settings of these two parameters under the considered ranges for test instances like CTP2 and CTP4.
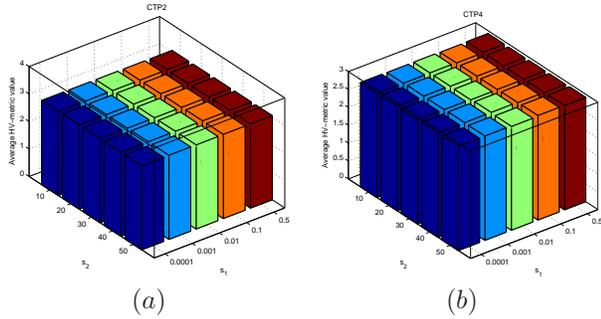
Fig. 3: The average HV-metric values obtained by CMOEA/D-DE-ATP with 25 different combinations of s_1 and s_2 on CTP2 and CTP4.
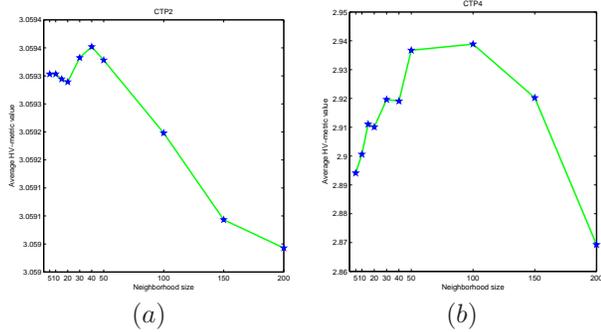
Fig. 4: The average HV-metric values vs. the neighborhood size (T) in CTP2 and CTP4.

C. Sensitivity of CMOEA/D-DE-ATP to T

To study the sensitivity of the performance of CMOEA/D-DE-ATP to T, 10 different values of neighborhood size (i.e., $T = 5, 10, 20, 30, 40, 50, 100, 150, 200$) are used in CMOEA/D-DE-ATP for test instances CTP2 and CTP4.

Figure 4 shows the average HV-metric values versus the neighborhood size T. It is clear from this figure that the performance of CMOEA/D-DE-ATP is sensitive to the setting of neighborhood size T. For test instance CTP2, the best (i.e., highest) HV-metric value is found by CMOEA/D-DE-ATP with $T = 40$. One can see from Figure 4 (a) that CMOEA/D-DE-ATP with $T = 30$ and $T = 50$ can find the average HV-metric values very close to the one obtained with $T = 40$. This indicates that CMOEA/D-DE-ATP can still find the good approximation of the PF of CTP2 when the neighborhood size varies in an appropriate range. Figure 4 (a) also shows that CMOEA/D-DE-ATP with $T < 30$ or $T > 100$ (although, the HV-metric values with $T < 30$ are much better than those with $T > 100$) will be unable to find good approximation of the PF of CTP2 as with $T = 40$, because the average HV-metric values found are smaller than the one obtained with $T = 40$. This can be explained by the exploration and exploitation of the search space in CMOEA/D-DE-ATP. This explains that, on one hand, CMOEA/D-DE-ATP lacks the ability of exploring the search space with small T, while on the other hand, it is unable to exploit the PS well with large T. In the latter case, the mating parents could be apart from each other (i.e., very

dissimilar) in the decision space. However, as can be seen from Figure 4 (b), T can be chosen from the range $50 < T < 100$ for better performance of the algorithm on test instance CTP4.
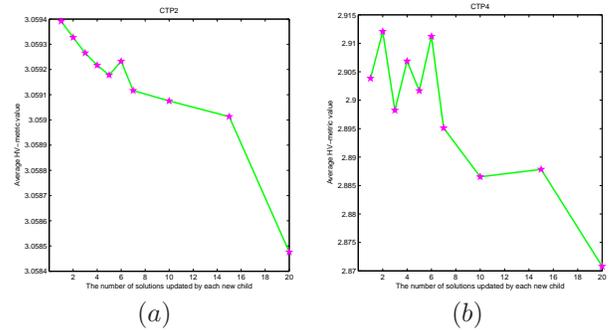
Fig. 5: The average HV-metric values vs. the maximal number of solutions updated by each new offspring (n_r) in CTP2 and CTP4.

D. Sensitivity of CMOEA/D-DE-ATP to n_r

To study the sensitivity of the performance of CMOEA/D-DE-ATP to n_r, 10 values of n_r (i.e., $n_r = 1, 2, 3, 4, 5, 6, 7, 10, 15, 20$) are tested in the algorithm for test instances CTP2 and CTP4.

Figure 5 presents the average HV-metric values versus the ten different values of n_r. As clearly shown in Figure 5 (a), CMOEA/D-DE-ATP with small $n_r = 1, 2, 3$ values can find the good approximation of the PF of CTP2. However, when n_r is large (i.e., ≥ 4), CMOEA/D-DE-ATP performs clearly worse in maximizing the HV-metric values.

For test instance CTP4, higher HV-metric value is achieved when $n_r = 2$ (see Figure 5 (b)), and when $n_r > 2$ except $n_r = 6$ (as the value of HV-metric obtained with $n_r = 6$ is very close the one obtained with $n_r = 2$), the values of the HV-metric get worsen.
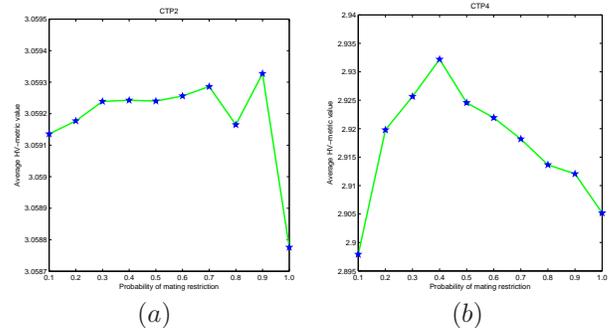
Fig. 6: The average HV-metric values vs. the probability of selecting mating parents from the neighborhood (δ) in CTP2 and CTP4.

E. Sensitivity of CMOEA/D-DE-ATP to δ

In order to study the sensitivity of the performance of CMOEA/D-DE-ATP to the probability of selecting mating parents from the neighborhood, δ, 10 values of δ (i.e., $\delta = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0$) are tried in CMOEA/D-DE-ATP for test instances CTP2 and CTP4.

Figure 6 shows the average HV-metric values versus the values of δ. It is clear from Figure 6 (a) that CMOEA/D-DE-ATP with $\delta = 0.9$ has the best performance in terms of the HV-metric for test instance CTP2. This indicates that selecting 90% solutions from the neighborhood of a solution and 10% of the solutions from the whole population for recombination does improve the performance of CMOEA/D-DE-ATP on CTP2. In the latter case, the solutions that are apart from each other in the search space will get a chance to mate with a low probability. Although this procedure boosts the capability of CMOEA/D-DE-ATP to explore the search space, but still more computational efforts are utilized on the recombination between neighboring solutions in order to exploit the PS efficiently.

However, for better performance on test instance like CTP4, 40% of the solutions are needed to be selected from the neighborhood and 60% from the whole population for recombination (see Figure 6 (b)).
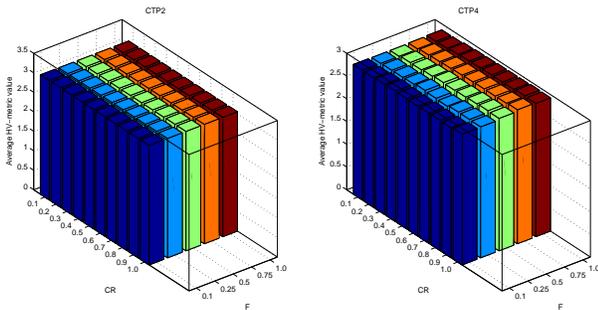
Fig. 7: The average HV-metric values obtained by CMOEA/D-DE-ATP with 50 different combinations of CR and F values on CTP2 and CTP4.

F. Sensitivity of CMOEA/D-DE-ATP 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 behaviour of CMOEA/D-DE-ATP, 50 combinations of five values of F ($F = 0.1, 0.25, 0.5, 0.75, 1.0$) and ten values of CR ($CR = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0$) are tested on test instances CTP2 and CTP4. Each combination of CR and F is tested 30 times.

Figure 7 shows the average HV-metric values with 50 different combinations of CR and F values. It is evident from this figure that CMOEA/D-DE-ATP is less sensitive to the settings of CR and F under the considered ranges and test instances.
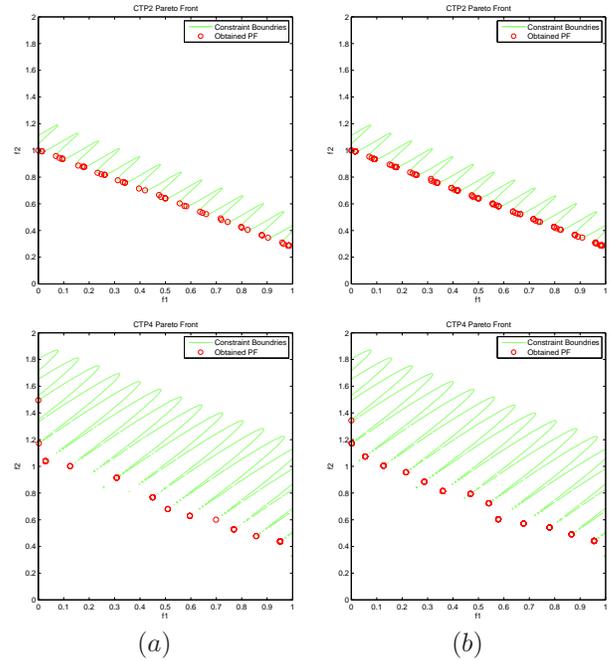
Fig. 8: Plots of the final solutions obtained from CMOEA/D-DE-ATP with population sizes of 50 (column a) and 100 (column b) evolved over 200 generations for CTP2 and CTP4.

G. CMOEA/D-DE-ATP with Small Population

Decision makers (DMs) often show interest in a small evenly distributed population at a low computational cost. In the following, it is shown that CMOEA/D-DE-ATP adopting small population could attain this objective. CMOEA/D-DE-ATP is run with population sizes of $N = 50$ and $N = 100$ and 200 generations on test instances CTP2 and CTP4. Accordingly, the algorithm stops after $10,000$ and $20,000$ function evaluations.

Figure 8 plots the final solutions obtained in a single run of CMOEA/D-DE-ATP with $N = 50$ (column a) and $N = 100$ (column b) for test instances CTP2 and CTP4. It is very evident from this figure that CMOEA/D-DE-ATP found 50 and 100 evenly distributed solutions for test instances CTP2. However, for test instance CTP4, the 100 final solutions are evenly distributed along the PF, but the 50 final solutions, although evenly distributed, misses some Pareto-optimal solutions.

As it can be seen that test instances CTP2 and CTP4 have disconnected continuous/discrete Pareto-optimal regions/solutions. Thus, any Pareto dominance based algorithm that prefers feasible solutions over infeasible solutions in their selection/replacement schemes, like NSGA-II with the constraint-domination principle, is likely to face difficulty in capturing the whole PF until a large population size is used to maintain population diversity [7]. On the other hand, decomposition based algorithm like CMOEA/D-DE-ATP, which tries to keep some good infeasible solutions in its course of evolution can capture the whole PF.
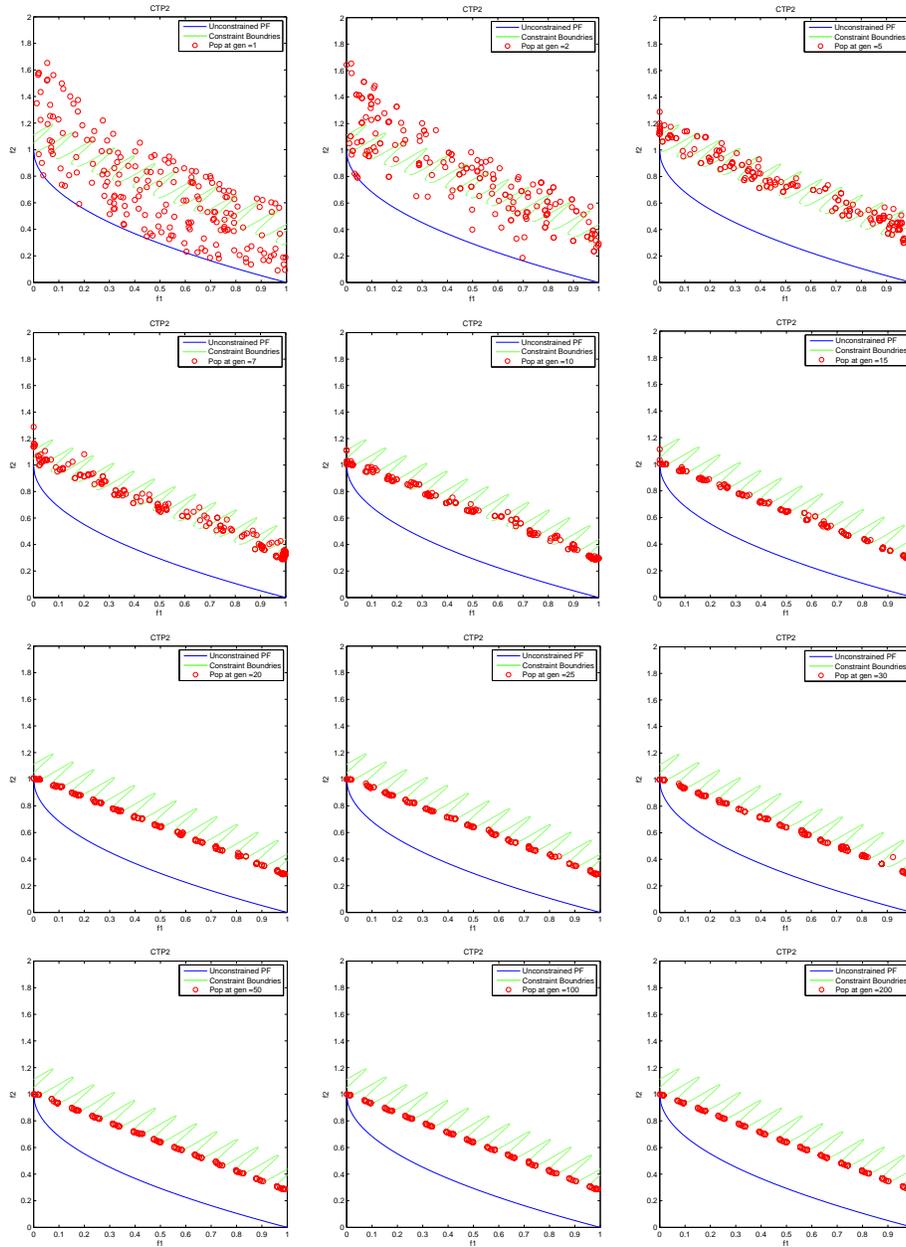
Fig. 9: Evolution of the CMOEA/D-DE-ATP population over generations for CTP2 test run.

H. Population Evolution in CMOEA/D-DE-ATP

Figures 9 and 10 show the progress of CMOEA/D-DE-ATP populations up to 200 generations for test instances CTP2 and CTP4. Both these figures demonstrate that how CMOEA/D-DE-ATP first converges the initial population to different disconnected constricted feasible regions near the PF in a few initial generations and then attains the Pareto-optimal solutions at the end of each one of them.

VI. CONCLUSIONS

A penalty function that penalizes infeasible solutions based on an adaptive threshold value has been introduced into the update and replacement scheme of CMOEA/D-DE. This

resulted in a new algorithm, CMOEA/D-DE-ATP for CMOO. The performance of CMOEA/D-DE-ATP is tested on CTP-series test instances in terms of the HV-metric values.

From the experimental results in this paper, we can make the following conclusions.

- The comparison of CMOEA/D-DE-ATP with IDEA and NSGA-II unveiled that CMOEA/D-DE-ATP can find better and consistent statistics, in terms of the HV-metric, than the two contestant algorithms for all CTP-series test instances except CTP4.

- The study of the sensitivity of CMOEA/D-DE-ATP to the proposed penalty function and algorithmic param-
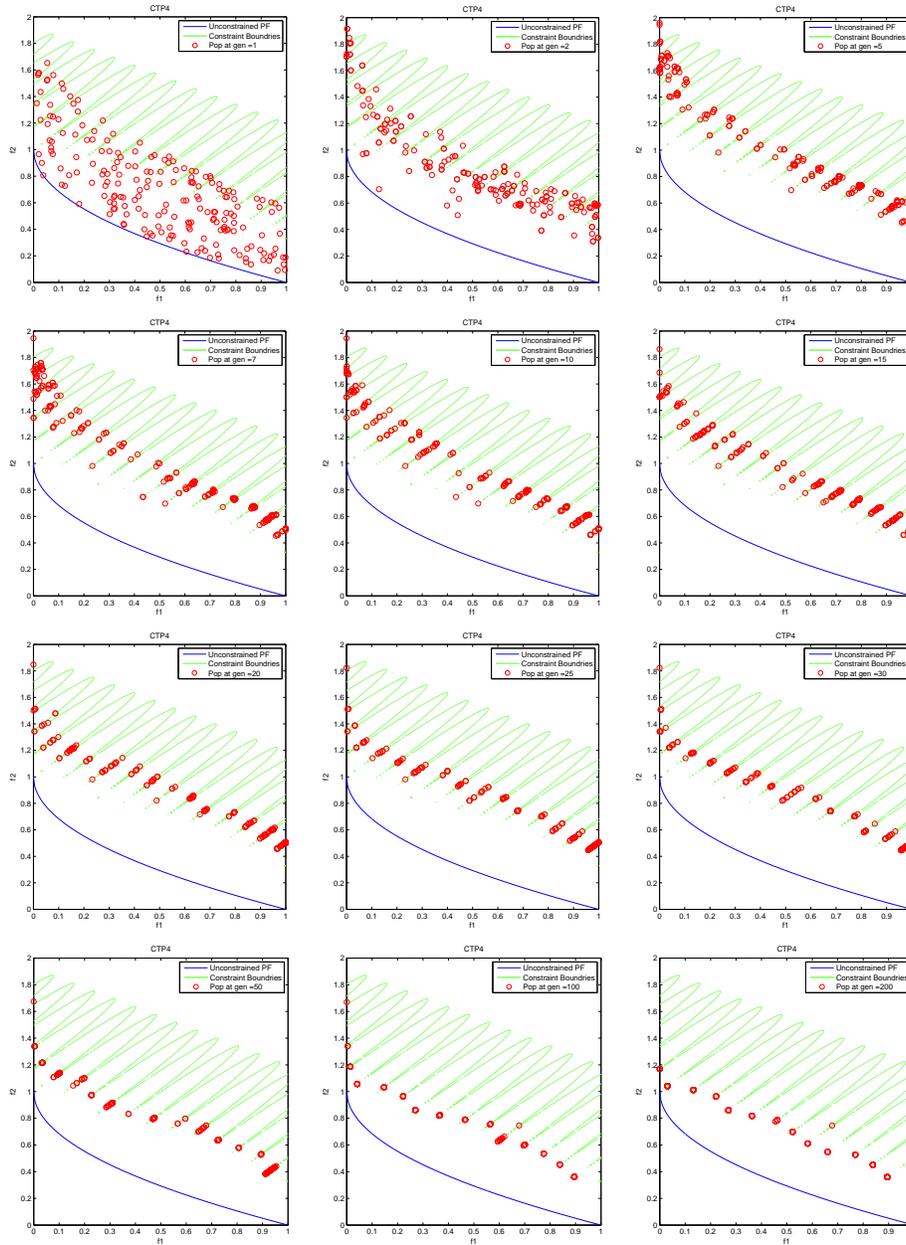
Fig. 10: Evolution of the CMOEA/D-DE-ATP population over generations for CTP4 test run.

eters disclosed that various parameters' settings are required for different test instances except s_1 and s_2 and CR and F.

- The proposed algorithm can find evenly distributed optimal solutions with a small population size, N.

REFERENCES

[1] 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.

[2] M. A. Jan, N. M. Tairan, R. A. Khanum, and W. K. Mashwani, "A new threshold based penalty function embedded MOEA/D," *International Journal of Advanced Computer Science and Applications(IJACSA)*, vol. 7, no. 2, pp. 647–655, 2016.

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

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

[5] M. Gen and R. Cheng, "Interval Programming using Genetic Algorithms," in *Proceedings of the Sixth International Symposium on Robotics and Manufacturing*, Montpellier, France, 1996.

[6] E. M. Montes and C. A. Coello Coello, "A simple multimembered evolution strategy to solve constrained optimization problems: Smes," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 1, pp. 1–17, February 2005.

[7] 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.

[8] Y. G. Woldesenbet, G. G. Yen, and B. G. Tessema, "Constraint handling in multiobjective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, June 2009.

[9] G. Yen, "An adaptive penalty function for handling constraint in multi-objective evolutionary optimization," *Constraint-Handling in Evolutionary Optimization*, pp. 121–143, 2009.

[10] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W H Freeman and Co San Francisco, 1979.

[11] S. S. Rao, *Engineering Optimization*, 3rd ed. John Wiley & Sons, 1996.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

[19] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, ser. Natural Computing. Secaucus, NJ, USA: Springer-Verlag, 2005.

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

[21] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithmsa comparative case study," in *Parallel Problem Solving from NaturePPSN V*. Springer, 1998, pp. 292–301.