# A Note on Constrained Multi-Objective Optimization Benchmark Problems

Ryoji Tanabe
Institute of Space and Astronautical Science
Japan Aerospace Exploration Agency
Email: rt.ryoji.tanabe@gmail.com

Akira Oyama
Institute of Space and Astronautical Science
Japan Aerospace Exploration Agency
Email: oyama@flab.isas.jaxa.jp

***Abstract*—We herein investigate the properties of widely used constrained multi-objective optimization benchmark problems. A number of multi-objective evolutionary algorithms (MOEAs) for constrained multi-objective optimization problems (CMOPs) have been proposed in the past few years. The C-DTLZ functions and real-world-like problems (RWLPs) have frequently been used to evaluate the performance of MOEAs on CMOPs. In this paper, however, we show that the C-DTLZ functions and widely used RWLPs have some unnatural problem features. The experimental results show that an MOEA *without* any constraint handling techniques (CHTs) can successfully find well-approximated nondominated feasible solutions on the C1-DTLZ1, C1-DTLZ3, and C2-DTLZ2 functions. It is widely believed that RWLPs are MOEA-hard problems, and finding feasible solutions on these problems is a very difficult task. However, we show that an MOEA without any CHTs can find feasible solutions on widely used RWLPs, such as the speed reducer design problem, the two-bar truss design problem, and the water problem. Moreover, the infeasible solution seldom simultaneously violates multiple constraints in the RWLPs. For the above reasons, we conclude that constrained multi-objective optimization benchmark problems require a careful reconsideration.**

## I. Introduction

A constrained multi-objective continuous optimization problem (CMOP), which frequently appears in engineering problems [1], [2], [3], [4], can be formulated as follows:

$$\begin{aligned} \text{minimize} \quad & f_i(\boldsymbol{x}), i \in \{1, ..., M\} \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \leq 0, i \in \{1, ..., p\} \\ & h_i(\boldsymbol{x}) = 0, i \in \{p+1, ..., N\} \end{aligned} \tag{1}$$

where $\boldsymbol{f} : \mathbb{S} \rightarrow \mathbb{R}^M$ is an objective function vector consisting of $M$ conflicting objective functions, and $\mathbb{R}^M$ is the objective function space. $\boldsymbol{x} = (x_1, ..., x_D)^{\mathrm{T}}$ is a $D$-dimensional solution vector, and $\mathbb{S} = \Pi_{j=1}^{D}[a_j, b_j]$ is the bound-constrained search space, where $a_j \leq x_j \leq b_j$ for each $j \in \{1, ..., D\}$. The feasible solution satisfies $p$ inequality constraint functions $\{g_1, ..., g_p\}$ and $N - p$ equality constraint functions $\{h_1, ..., j_{N-p}\}$. The set of all feasible solutions is called the feasible region $\mathbb{F} \subseteq \mathbb{S}$, whereas $\boldsymbol{x} \notin \mathbb{F}$ in $\mathbb{S}$ is the infeasible solution. In the same manner, the set of all infeasible solutions is called the infeasible region $\overline{\mathbb{F}}$.

For the $N$ inequality and equality constraint functions in Equation (1), the constraint violation value $c_i(\boldsymbol{x})$, $i \in \{1, ..., N\}$ of the solution $\boldsymbol{x}$ can be defined as follows:

$$c_i(\boldsymbol{x}) = \begin{cases} \max(0, g_i(\boldsymbol{x})), & i \in \{1, ..., p\} \\ \max(0, |h_i(\boldsymbol{x}) - \delta|), & i \in \{p+1, ..., N\} \end{cases} \tag{2}$$

where the tolerance value $\delta$ is generally used for relaxing the equality constraints to the inequality constraints, and $\delta$ should be set to a sufficiently small value (e.g., $\delta = 10^{-6}$). If the solution $\boldsymbol{x}$ is feasible, $\sum_{i=1}^{N} c_i(\boldsymbol{x}) = 0$.

For $\boldsymbol{x}^1$, $\boldsymbol{x}^2 \in \mathbb{F}$, we say that $\boldsymbol{x}^1$ dominates $\boldsymbol{x}^2$ and denote $\boldsymbol{x}^1 \prec \boldsymbol{x}^2$ if and only if $f_i(\boldsymbol{x}^1) \leq f_i(\boldsymbol{x}^2)$ for all $i \in \{1, ..., M\}$ and $f_i(\boldsymbol{x}^1) < f_i(\boldsymbol{x}^2)$ for at least one index $i$. $\boldsymbol{x}^*$ is a Pareto optimal solution if there exists no $\boldsymbol{x} \in \mathbb{F}$ such that $\boldsymbol{x} \prec \boldsymbol{x}^*$. $\boldsymbol{f}(\boldsymbol{x}^*)$ is also called a Pareto optimal objective function vector. The set of all $\boldsymbol{x}^*$ is the Pareto optimal solution set (PS), and the set of all $\boldsymbol{f}(\boldsymbol{x}^*)$ is the Pareto frontier (PF). The goal of CMOPs is to find a set of nondominated feasible solutions that are well-distributed and close to the PF in the objective function space.

A multi-objective evolutionary algorithm (MOEA) is one of most promising approaches for solving (C)MOPs [5]. Since MOEAs use a set of individuals (solutions of a given problem) for the search, it is possible that they can find good nondominated feasible solutions in a single run. Unfortunately, as pointed out in [6], the evolutionary computation community has focused primarily on studies of MOEAs for unconstrained or bound-constrained MOPs, and the number of studies on CMOPs is much smaller. However, a number of novel, efficient MOEAs for CMOPs has been proposed in the past few years. Recently proposed representative MOEAs for (C)MOPs include NSGA-III [7], U-NSGA-III [8], I-DBEA [9], MOEA/DD [10], and RVEA [11]. Surrogate-assisted MOEAs for CMOPs have also been studied [12].

In general, the performance of MOEAs (and EAs) is difficult to evaluate theoretically and thus is instead evaluated experimentally through computational simulation. Suppose that we apply an MOEA to a specific real-world problem and evaluate its performance. In this case, reproducing experiments conducted by other researchers is often very difficult because the computational simulation of real-world problems often requires special hardware or software. Therefore, artificially designed benchmark functions have been widely used to evaluate the performance of MOEAs.

Typical constrained multi-objective benchmark functions include the SRN function [13], the TNK function [14], the OSY function [15], the CTP functions [16], and the CF functions [17]. However, most of these functions are two-objective CMOPs, in which $M$ cannot be set to an arbitrary number. In other words, the benchmark functions described above are not suitable for evaluating the scalability of MOEAs with respect to $M$. On the other hand, the number of objectives of the recently proposed C-DTLZ functions [7] can be set to an arbitrary number, where the C-DTLZ functions are extended variants of the DTLZ functions [18]. Therefore, the C-DTLZ functions have frequently been used to evaluate the performance of MOEAs in recent studies [7], [8], [10], [11], [12]. While all of the benchmark functions described above are artificially designed functions, real-world-like problems (RWLPs) have also been used in comparative studies. Typical RWLPs include the two-bar truss design problem [19], the car side impact problem [7], and the water problem [20]. It is widely believed that finding feasible solutions for RWLPs is a very difficult task because most RWLPs have many complex constraints. Thus, RWLPs have been considered to be challenging, difficult problems for MOEAs. In summary, there are a number of benchmark problems for evaluating the performance of MOEAs on CMOPs. However, as far as we know, their properties have been poorly investigated, and, with some few exceptions (e.g., [16], [21]), their suitability as benchmark problems for MOEAs has barely been discussed.

In this paper, we investigate the properties of the C-DTLZ functions and widely used RWLPs, and show that they have some issues. For the C-DTLZ functions, we demonstrate that an MOEA *without* any constraint handling techniques (CHTs) can find well-approximated nondominated feasible solutions for the C1-DTLZ1, C1-DTLZ3, and C2-DTLZ2 functions. As mentioned above, many researchers in the evolutionary computation community have considered RWLPs to be MOEA-hard problems, and finding feasible solutions to such problems is a difficult task. However, the experimental results obtained in this paper reveal that an MOEA without any CHTs can find feasible solutions for widely used RWLPs. Nevertheless, infeasible solutions often simultaneously violate multiple constraints for the widely used RWLPs. For these reasons, although the C-DTLZ functions and RWLPs have frequently been used to evaluate the performance of MOEAs in recent studies [7], [8], [10], [11], [12], they may require careful reconsideration.

The remainder of this paper is organized as follows. Section II introduces typical constrained multi-objective benchmark problems. Section III describes the experimental settings, and we investigate the properties of constrained multi-objective benchmark problems in Section IV. Finally, Section V concludes this paper and discusses our future work.

## II. Review of constrained multi-objective optimization benchmark problems

Table I shows some properties of typical constrained multi-objective optimization benchmark problems described in this section. It is widely believed that the feasibility ratio represents the difficulty of finding feasible solutions [22], [23]. For details on each CMOP in Table I, see the corresponding paper.

TABLE I: Properties of typical constrained multi-objective optimization benchmark problems (the number of objectives $M$, the number of constraint functions $N$, the dimensionality $D$, and the feasibility ratio). The feasibility ratio of the search space (i.e., $|\mathbb{F}|/|\mathbb{S}|$) was experimentally estimated by calculating the percentage of feasible solutions in $10^5$ uniformly randomly generated solutions, as in [22], [23].

| Problem | $M$ | $N$ | $D$ | Feasibility ratio |
|---|---|---|---|---|
| SRN | 2 | 2 | 2 | 0.08 |
| TNK | 2 | 2 | 2 | 0.03 |
| OSY | 2 | 6 | 6 | 0.02 |
| C1-DTLZ1 | $\geq 2$ | 1 | $M+k-1$ | 0.0 |
| C1-DTLZ3 | $\geq 2$ | 1 | $M+k-1$ | 0.0 |
| C2-DTLZ2 | $\geq 2$ | 1 | $M+k-1$ | 0.02 |
| C3-DTLZ1 | $\geq 2$ | $M$ | $M+k-1$ | 0.5 |
| C3-DTLZ4 | $\geq 2$ | $M$ | $M+k-1$ | 0.12 |
| $f_{\mathrm{TBTD}}$ | 2 | 3 | 3 | 0.0 |
| $f_{\mathrm{SRD}}$ | 2 | 11 | 7 | 0.043 |
| $f_{\mathrm{DBD}}$ | 2 | 5 | 4 | 0.319 |
| $f_{\mathrm{WB}}$ | 2 | 4 | 4 | 0.058 |
| $f_{\mathrm{CSI}}$ | 3 | 10 | 7 | 0.181 |
| $f_{\mathrm{SPD}}$ | 3 | 9 | 6 | 0.026 |
| $f_{\mathrm{W}}$ | 5 | 7 | 3 | 0.920 |

The SRN function [13], the TNK function [14], and the OSY function [15] are among the most widely used benchmark functions. These three functions were proposed in $1994 \sim 1995$ and have been used in numerous studies [24], [25], [23], [26]. However, Deb et al. pointed out that these functions have the following three issues: the dimensionality $D$ is too small, finding good solutions is not so difficult, and the difficulty and complexity of optimization cannot be tuned [16].

In order to address the above issues, Deb et al. proposed the CTP functions [16]. The CTP can be considered to be a general framework for constructing novel constrained multi-objective optimization benchmark problems. Various benchmark functions can be generated using the CTP framework. In [16], seven CTP functions (CTP1, ..., CTP7) were designed. The CTP framework was also used for constructing the CF functions [17] for the IEEE CEC2009 MOEA Competition[1]. Very recently, Li et al. proposed more difficult variants of the CTP functions [21]. Unfortunately, most of the CTP functions described above are two-objective CMOPs and are not suitable

[1] http://dces.essex.ac.uk/staff/zhang/moeacompetition09.htm

for evaluating the scalability of MOEAs with respect to $M$[2].

In [7], Jain and Deb propose five C-DTLZ functions. The C-DTLZ functions are extended variants of the DTLZ functions [18] for benchmarking MOEAs for CMOPs. Unlike the classical benchmark functions described above, the number of objectives of the C-DTLZ functions can be set to an arbitrary number, and thus they have frequently been used in recent comparative studies [7], [10], [11], [12]. According to the characteristics of the feasible region in the objective function space, the C-DTLZ functions are classified into the following three categories: Type-1, Type-2, and Type-3 problems.

Figure 1 shows the feasible or infeasible region of each C-DTLZ function in the objective space ($M = 2$). The C2-DTLZ2 and C3-DTLZ4 functions are unimodal, and the C1-DTLZ1, C1-DTLZ3, and C3-DTLZ1 functions are multimodal. In Type-1 constraint C-DTLZ functions (C1-DTLZ$*$), the shape and position of the PF are the same as in the original DTLZ functions, but there is "an infeasible barrier" that prevents MOEAs from approaching the PF. Thus, the population must overcome this infeasible barrier to converge to the true PF, which can be considered a difficult task. In Type-2 constraint C-DTLZ functions (C2-DTLZ$*$), part of the PF becomes the infeasible region by introducing a constraint. In other words, the shape of the PF of C2-DTLZ$*$ is discontinuous. In general, handling the discontinuously of the PF is difficult for MOEAs. Note that the constraint is nevertheless introduced, and the position of the PF of C1-DTLZ$*$ and C2-DTLZ$*$ remains unchanged from the original DTLZ functions. On the other hand, in Type-3 constraint C-DTLZ functions (C3-DTLZ$*$), the region of the original PF is made infeasible by introducing $M$ linear constraints, and the PF of C3-DTLZ$*$ is a boundary line between the feasible and infeasible region. Note that most constraint multi-objective benchmark functions such as the SRN, TNK, OSY, and the CTP functions (except for the CTP7 function) can be classified as Type-3 constraint functions.

RWLPs have also frequently been used for comparative studies [19], [27], [23], [26], [7]. Typical RWLPs include the two-bar truss design problem ($f_{\mathrm{TBTD}}$) [19], [26], the speed-reducer design problem ($f_{\mathrm{SRD}}$) [19], [26], the disc brake design problem ($f_{\mathrm{DBD}}$) [19], [26], the welded beam problem ($f_{\mathrm{WB}}$) [28], [26], the car side-impact problem ($f_{\mathrm{CSI}}$) [7], the ship parametric design problem ($f_{\mathrm{SPD}}$) [29], [27], and the water problem ($f_{\mathrm{W}}$) [20], [7]. It is generally believed that finding feasible solutions for RWLPs is a very difficult task because most RWLPs have many complex constraints [7]. In fact, as shown in Table I, for most RWLPs, $N$ is much larger than that for the artificially designed functions (e.g., $f_{\mathrm{CSI}}$ has 11 constraints), and the feasibility ratios of some RWLPs are low. Therefore, RWLPs have been considered to be challenging, difficult problems for MOEAs.

[2]The idea of constructing CTP functions with $M \geq 3$ is briefly introduced in [16], but to the best of our knowledge, they have never been realized.

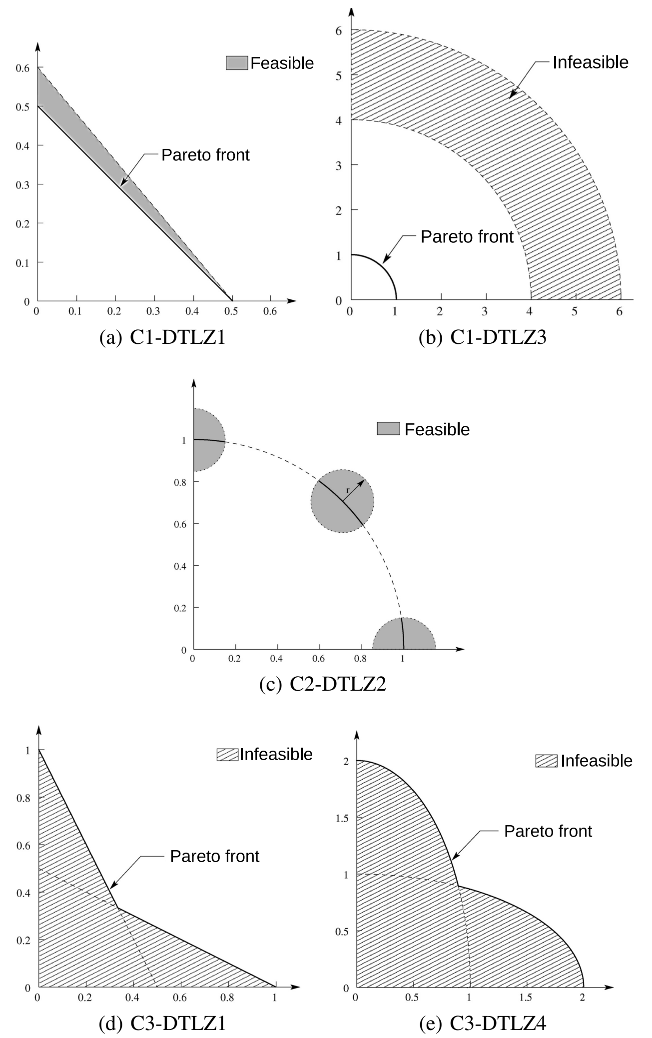



Fig. 1: Feasible or infeasible region of each C-DTLZ function in the objective space ($M = 2$). All figures were derived from [7].

## III. Experimental settings

In this paper, we investigate the properties of the two groups of constrained multi-objective optimization benchmark problems (C-DTLZ functions and RWLPs) described in Section II. Here, we describe the experimental settings. The results will be discussed in Section IV.

### A. Problems and performance evaluation methods

We used the two- and three-objective C-DTLZ functions[3]. As suggested in [7], the position parameter $k$ was set to $k = 10$ for the C1-DTLZ3 and C2-DTLZ2 functions and to $k = 5$ for the remaining functions. We also used the seven RWLPs in Table I.

We used the hypervolume (HV) indicator [30] for evaluating the quality of a set of obtained nondominated solutions $\boldsymbol{A}$. Before calculating the HV value, the objective function vector

[3]We used the source code of the C-DTLZ functions implemented by Li. The code was downloaded from http://www.cs.bham.ac.uk/~likw/.

$\boldsymbol{f}(\boldsymbol{x})$ of each $\boldsymbol{x} \in \boldsymbol{A}$ was normalized using the ideal point and the nadir point. The ideal point and the nadir point for all problems were experimentally estimated using all solutions obtained by all methods for all 101 runs. The reference point for calculating HV was set to $(1.1, ..., 1.1)^{\mathrm{T}}$. Note that we used only the feasible solutions dominating the reference point for the HV calculation.

Almost all previous studies (e.g., [7], [8], [10], [9], [11], [12]) used nondominated solutions in the population at the end of the search for calculating the HV value. In general, an MOEA maintains (nondominated) solutions obtained during the search process in the population, but the population size is limited. When using the nondominated solutions in the population for the HV calculation, a monotonic increase in HV over time (= the number of function evaluations) cannot be ensured [31], [32]. Thus, we cannot exactly evaluate the performance of MOEAs using such traditional evaluation methodology.

In order to address this issue, we used an *unbounded* external archive as suggested in [32], [33]. The unbounded external archive stores *all* nondominated feasible solutions found during the search process and can be introduced into any MOEAs without any changes to their original algorithms [32], [33]. Solutions in the unbounded external archive are not used in the search of an MOEA. When using the unbounded external archive, the above-described issue can be addressed [31], [32]. Since, in practice, users of MOEAs want to obtain nondominated solutions whenever possible, we believe that the benchmark methodology using the unbounded external archive is more practical. The unbounded external archive is also used in the recently proposed BBOB-biobj benchmark suite [34] in the COCO framework[4]. When the number of obtained nondominated solutions in the unbounded external archive is too large, a method that selects only representative solutions (e.g., [35]) should be used.

### B. MOEAs

We analyzed the performance of five variants of (C)NSGA-II [24] because it is one of the most widely used MOEAs for CMOPs. CNSGA-II is the NSGA-II algorithm using the constraint-domination $\prec_{\mathrm{const}}$, instead of the Pareto-domination $\prec$ described in Section I. For $\boldsymbol{x}^1$, $\boldsymbol{x}^2 \in \mathbb{S}$, we say that $\boldsymbol{x}^1$ constraint-dominates $\boldsymbol{x}^2$ and denote $\boldsymbol{x}^1 \prec_{\mathrm{const}} \boldsymbol{x}^2$ if the two vectors satisfy one of the following three conditions: (1) $\boldsymbol{x}^1 \in \mathbb{F}$ and $\boldsymbol{x}^2 \notin \mathbb{F}$, (2) $\boldsymbol{x}^1, \boldsymbol{x}^2 \notin \mathbb{F}$ and $C(\boldsymbol{x}^1) < C(\boldsymbol{x}^2)$, and (3) $\boldsymbol{x}^1, \boldsymbol{x}^2 \in \mathbb{F}$ and $\boldsymbol{x}^1 \prec \boldsymbol{x}^2$, where $C$ is a constraint violation function summarizing how the solution $\boldsymbol{x}$ violates $N$ constraints in Equation (1).

We investigate the performance of the following five (C)NSGA-II algorithms using different types of $C$:

**1. NSGA-II**: The NSGA-II using $\prec$, rather than $\prec_{\mathrm{const}}$.
**2. CNSGA-II-S**: The CNSGA-II using the sum of the constraint violation values $\{c_1(\boldsymbol{x}), ..., c_N(\boldsymbol{x})\}$ defined in Equation (2) as $C$, i.e., $C(\boldsymbol{x}) = \sum_{i=1}^{N} c_i(\boldsymbol{x})$.
**3. CNSGA-II-NS**: The CNSGA-II using the normalized sum of the constraint violation values as $C$, i.e., $C(\boldsymbol{x}) = \sum_{i=1}^{N} (c_i(\boldsymbol{x}) - c_i^{\min})/(c_i^{\max} - c_i^{\min})$, where $c_i^{\min} = \min_{\boldsymbol{y} \in \boldsymbol{P}}\{c_i(\boldsymbol{y})\}$, $c_i^{\max} = \max_{\boldsymbol{y} \in \boldsymbol{P}}\{c_i(\boldsymbol{y})\}$, and $\boldsymbol{P}$ is the population.
**4. CNSGA-II-CD**: The CNSGA-II using the Pareto-dominance relationship in the constraint violation value space $\mathbb{R}^N$ [36] as $C$.
**5. CNSGA-II-RR**: The CNSGA-II using the relative ranking (RR) [27] as $C$.

For details on CD and RR, see [36], [27], respectively. CNSGA-II-S is identical to the original "CNSGA-II" proposed in [24] and most typical variant in the five (C)NSGA-II algorithms. The aim of the normalization procedure of CNSGA-II-NS and the sophisticated techniques of CNSGA-II-CD and CNSGA-II-RR is to handle the different scale of the constraint violation values. For CNSGA-II-CD and CNSGA-II-RR, if the two compared individuals $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ have the same rank level, then the sum of the constraint violation values, as in CNSGA-II-S, is used as a tie-breaking method. Note that, as far as we know, no previous study analyzed the impact of $C$ on the performance of CNSGA-II, in isolation.

We used the jMetal[5] source code of NSGA-II. We used the SBX crossover and polynomial mutation as in the original CNSGA-II paper [24]. As suggested in [24], we set the control parameters of the variation operators as follows: $p_c = 1.0$, $\eta_c = 20$, $p_m = 1/D$, and $\eta_m = 20$. The population size of the CNSGA-II algorithms was set to 100. For handling the different scale of the objective function values, we introduced the normalization procedure of the objective function values to all five algorithms. The maximum number of function evaluations was set to $5 \times 10^4$, and 101 independent runs were performed. Owing to a sufficiently large sample size, statistical tests are not necessary.

## IV. Experimental results and discussion

Here, we report and discuss the experimental results for the five (C)NSGA-II algorithms for the five C-DTLZ functions and the seven RWLPs listed in Table I. First, we describe the results for the five C-DTLZ functions and seven RWLPs in Sections IV-A and IV-B, respectively. Finally, in Section IV-C, we generally discuss the experimental results for the C-DTLZ functions and RWLPs.

### A. Results for the five C-DTLZ functions

Figure 2 shows the performance comparison of the five (C)NSGA-II algorithms for the two-objective C-DTLZ functions. Due to space constraints, we do not show the results for three objective functions, but they are similar to those shown in Figure 2. Note that the behavior of the four CNSGA-II algorithms for the C1-DTLZ1, C1-DTLZ3, and C2-DTLZ2 functions is exactly the same, because $N = 1$ in these problems.

For the Type-1 functions, CNSGA-II performs well on the C1-DTLZ1 function for a number of function evaluations

[4]http://coco.gforge.inria.fr/

[5]The code was downloaded from http://jmetal.sourceforge.net/

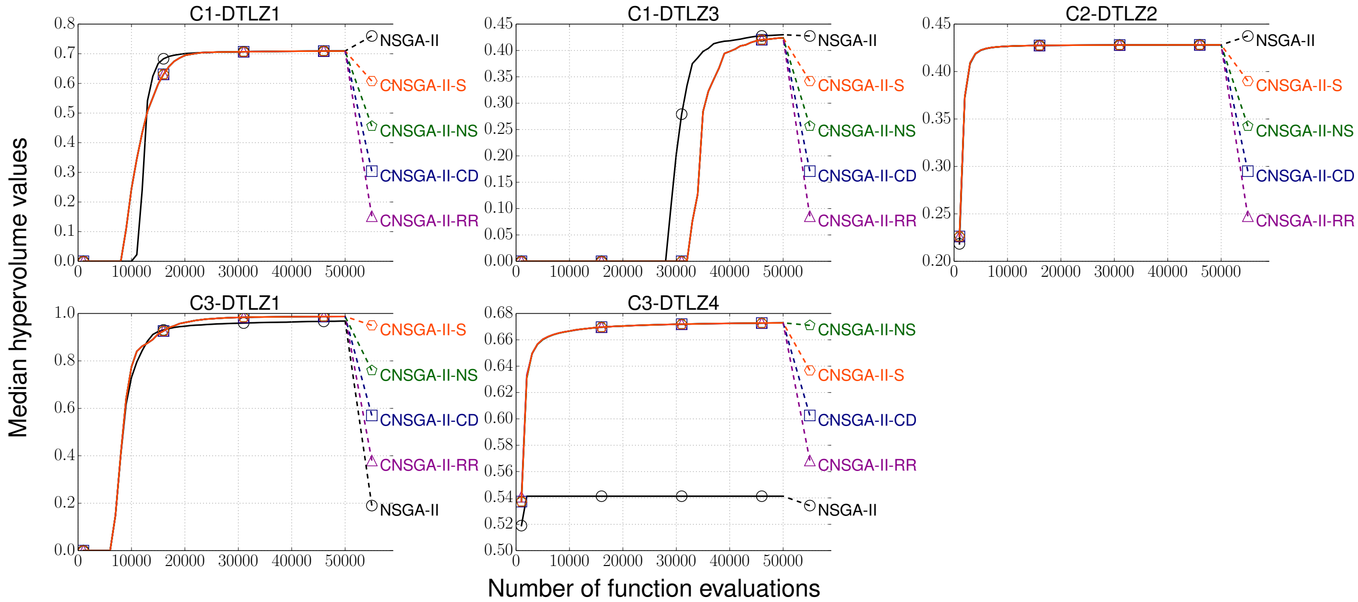



Fig. 2: Convergence behavior of the five (C)NSGA-II algorithms for the five C-DTLZ functions with $M = 2$. The median HV values across all the 101 runs are plotted.

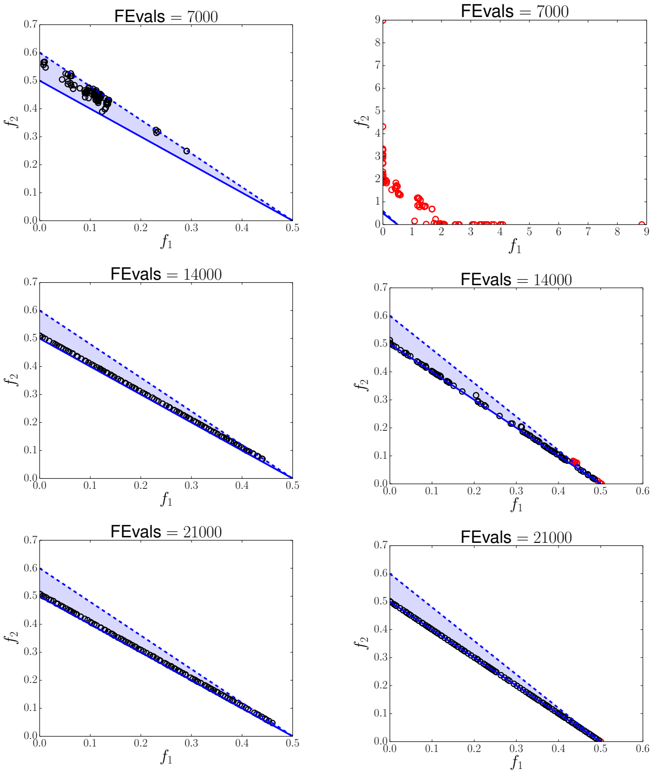
(FEvals) of approximately $1.2 \times 10^4$. However, for FEvals $= 1.2 \times 10^4 \sim 2.3 \times 10^4$, the four CNSGA-II algorithms are clearly outperformed by NSGA-II, which does not use any CHTs. For FEvals $> 2.3 \times 10^4$, there is no significant difference in performance among the five algorithms.

To explain the results for the C1-DTLZ1 function, we show the distributions of individuals in the populations of (a) CNSGA-II-S and (b) NSGA-II for FEvals $\in \{7\,000, 14\,000, 21\,000\}$ in Figure 3. As shown in Figure 3, for FEvals $= 7\,000$, CNSGA-II-S converges to the feasible region as well as the PF faster than NSGA-II. The reason for this result is that when all individuals in the population of CNSGA-II-S are infeasible, the environmental selection of CNSGA-II-S is performed based on only the amount of the constraint violation values (see Section III-B). In other words, when all individuals in the population are infeasible, the algorithmic behavior of CNSGA-II-S is identical to that of a single-objective GA minimizing the sum of the constraint violation values $C(\boldsymbol{x}) = \sum_{i=1}^{N} c_i(\boldsymbol{x})$. Due to this reason, we believe that CNSGA-II-S could find good feasible solutions faster than NSGA-II. However, the distribution of the individuals of CNSGA-II is clearly biased to $f_2$ for FEvals $= 7\,000$. After finding the feasible solutions, the population of CNSGA-II gradually creeps towards the $f_1$ along the PF. In summary, CNSGA-II-S finds the feasible solutions quickly, but requires many FEvals in order to obtain good solutions for the PF. On the other hand, the individuals in the population of NSGA-II are widely distributed when reaching the feasible region (FEvals $= 14\,000$). For FEvals $= 21\,000$, NSGA-II finds well-distributed feasible solutions while the population of CNSGA-II-S still moves toward $f_1$. These are the reasons why the HV values of NSGA-II are higher than those of the CNSGA algorithms for FEvals $= 1.2 \times 10^4 \sim 2.3 \times 10^4$ for the C1-DTLZ1 function.

For the C1-DTLZ3 function, surprisingly, NSGA-II clearly outperforms the four CNSGA-II algorithms at all times. The reason for this result is that the infeasible barrier of the C1-DTLZ3 function (Figure 1(b)) does not affect the search of an MOEA without any CHTs.

For the C2-DTLZ2 function, there is no performance difference between NSGA-II and CNSGA-II. In the C2-DTLZ2 function, part of the PF is the infeasible region, but the position of the PF remains unchanged. Recall that, in this experiment, we used the unbounded external archive that stores all nondominated feasible solutions found during the search process. The shape of the PF of the C2-DTLZ2 function is disconnected, and it is widely believed that handling the discontinuities in the PF is difficult [7]. However, when using the unbounded external archive, the distribution of the individuals in the population is not relevant to the HV value, so the effect of discontinuity might not be so significant. The results for the CTP7 function [16] are also shown in Figure 4. We selected the CTP7 function because part of its PF is the infeasible region, and it can be classified as a Type-2 constraint function. As expected, the results for the CTP7 function are similar to the results for the C2-DTLZ2 function.

Finally, although unsurprising, the four CNSGA-II algorithms perform significantly better than NSGA-II for the C3-DTLZ1 and C3-DTLZ4 functions, especially the C3-DTLZ4 function. Although not shown here, but the poor performance of NSGA-II was also observed for the SRN, TNK, and OSY functions. In explaining this result, we show the distributions of individuals in the populations of CNSGA-II-S and NSGA-II for the C3-DTLZ4 function in Figure 5. Note that all individ-




(a) CNSGA-II-S (b) NSGA-II

Fig. 3: Distributions of individuals in the populations of (a) CNSGA-II-S and (b) NSGA-II for the two-objective C1-DTLZ1 function in a single run (FEvals $\in \{7\,000, 14\,000, 21\,000\}$). The red and black circles represent the infeasible and feasible solutions respectively. The bold blue line indicates the PF, and the shaded region indicates the feasible region.

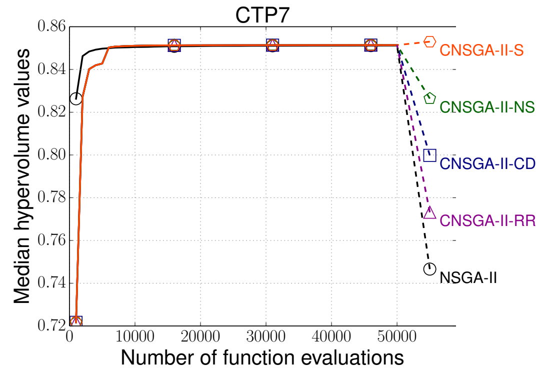



Fig. 4: Convergence behavior of the five (C)NSGA-II algorithms for the two-objective CTP7 function. The median HV values are plotted.

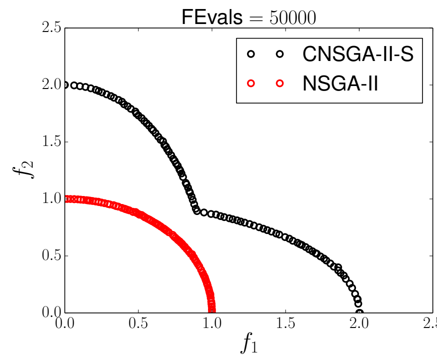



Fig. 5: Distributions of individuals in the populations of CNSGA-II-S and NSGA-II for the two-objective C3-DTLZ4 function in a single run (FEvals $= 50\,000$). All individuals of CNSGA-II-S and NSGA-II are feasible and infeasible, respectively.

uals of CNSGA-II-S and NSGA-II are feasible and infeasible, respectively. For the C3-DTLZ4 function, the original PF is infeasible, and its PF is a boundary line between the feasible and infeasible region. For this reason, in Figure 5, NSGA-II, which does not use any CHTs, passes the true PF and converges to the original PF. As a result, NSGA-II fails to find good feasible solutions.

### *B. Results for the seven RWLPs*

Figure 6 shows the performance comparisons of the five (C)NSGA-II algorithms on the six RWLPs ($f_{\mathrm{TBTD}}$, $f_{\mathrm{WB}}$, $f_{\mathrm{SRD}}$, $f_{\mathrm{CSI}}$, $f_{\mathrm{SPD}}$, and $f_{\mathrm{W}}$). We do not show the result for $f_{\mathrm{DBD}}$, which are similar to the results for $f_{\mathrm{SRD}}$.

As shown in Figure 6, the results for the RWLPs are similar to the results for the Type-3 constraint C-DTLZ functions. Namely, NSGA-II performs significantly worse than the four CNSGA-II algorithms, especially for $f_{\mathrm{DBD}}$ and $f_{\mathrm{SPD}}$. For $f_{\mathrm{WB}}$, NSGA-II achieves good feasible solutions only for FEvals $= 10^3$. We believe that the reason for the poor performance of NSGA-II for the RWLPs is the same as that for the C3-DTLZ4 function described in Section IV-A. Note that NSGA-II finds the feasible solutions for all seven RWLPs while many researchers consider these problems to be difficult for finding the feasible solutions. Moreover, it is widely believed that well-designed CHTs are needed to handle the different scale of each constraint violation value in $\{c_1, ..., c_N\}$ for the RWLPs [36], [27], [23], [7]. However, in this study, we compared the four CNSGA-II algorithms (CNSGA-II-S, CNSGA-II-NS, CNSGA-II-CD, and CNSGA-II-RR), but contrary to intuition, there is no significant difference between their performance. CNSGA-II-S uses the most simple $C$, and so it should perform worse than the remaining CNSGA-II algorithms using the more sophisticated $C$.

To analyze the results, we show the cumulative number of constraints that the infeasible solutions generated by CNSGA-II-S simultaneously violate for $f_{\mathrm{SRD}}$, $f_{\mathrm{CSI}}$, and $f_{\mathrm{W}}$ in Figure 7. As shown in Figure 7, the number of constraint functions that are simultaneously violated can be found out. For example, for $f_{\mathrm{SRD}}$, until FEvals $= 8\,000$, CNSGA-II-S generated about the $1\,700$ and $300$ infeasible solutions that simultaneously violate one and two constraint functions respectively. The infeasible solutions generated by CNSGA-II-S never simultaneously violated more than four constraints for $f_{\mathrm{SRD}}$. As shown in Table I, there are the 11, 10, and seven constraints in $f_{\mathrm{SPD}}$, $f_{\mathrm{CSI}}$, and $f_{\mathrm{W}}$, respectively. Handling

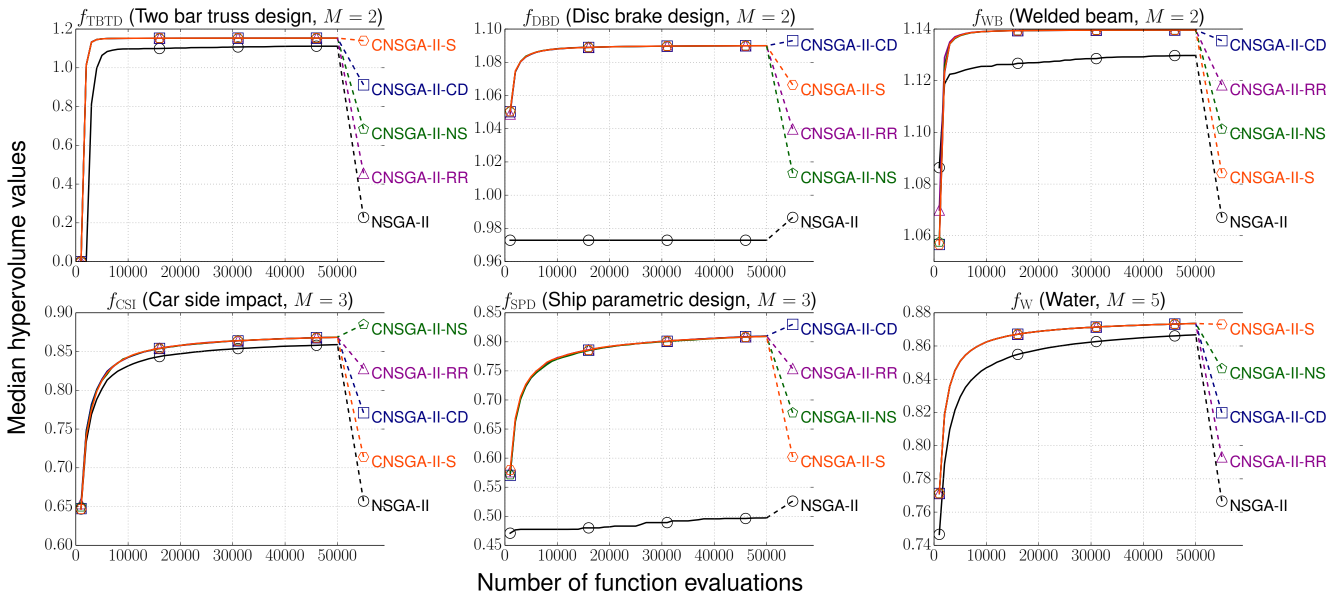



Fig. 6: Convergence behavior of the five (C)NSGA-II algorithms for the six RWLPs ($f_{\mathrm{TBTD}}$, $f_{\mathrm{WB}}$, $f_{\mathrm{SRD}}$, $f_{\mathrm{CSI}}$, $f_{\mathrm{SPD}}$, and $f_{\mathrm{W}}$). The median HV values across all 101 runs are plotted.

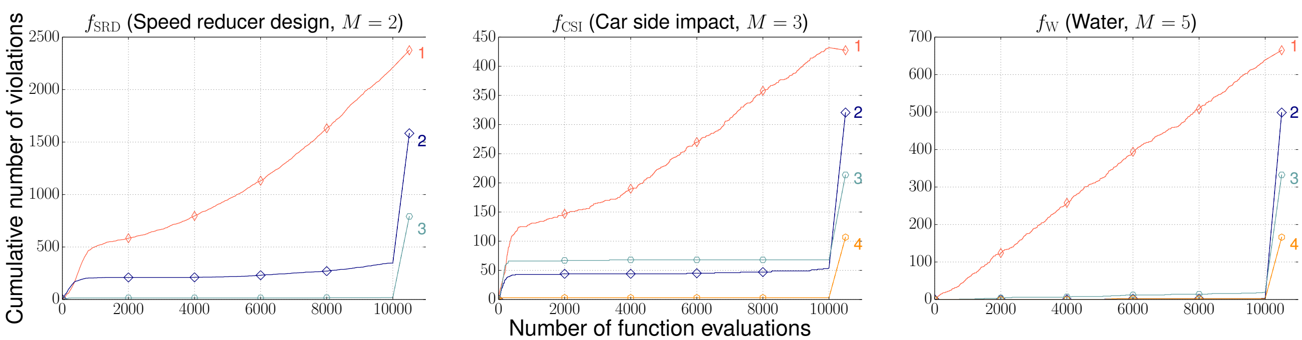



Fig. 7: Cumulative number of constraints that the infeasible solutions generated by CNSGA-II-S simultaneously violate.

many constraints and finding good feasible solutions for such problems appears to be difficult. However, as shown in Figure 7, most solutions obtained during the search process are feasible, and, in many cases, the infeasible solutions violate only one constraint. This is the reason why the four CNSGA-II algorithms perform similarly to each other on the seven RWLPs in Figure 6. Although there are many constraints in the seven RWLPs, finding the feasible solutions appears to be relatively easy.

### *C. Overall discussion of the experimental results for the C-DTLZ functions and RWLPs*

Based on the experimental results described in Section IV-A, we can say that an MOEA without any CHTs (e.g., NSGA-II) can find well-approximated feasible solutions for the Type-1 and Type-2 constraint functions. Since the position of the PF of the Type-1 and Type-2 functions is unchanged from the original functions, improving the solution according to the objective function values is identical to minimizing the constraint violation values (i.e., searching the feasible solutions). The Type-1 and Type-2 constraint functions should probably not be used for evaluating the performance of MOEAs for CMOPs in light of our experimental results.

RWLPs have been considered to be challenging, difficult problems [7]. However, in Section IV-B, we showed that even NSGA-II without any CHTs can easily find feasible solutions for the seven RWLPs. Moreover, the CNSGA-II algorithms with the simple $C$ (i.e., CNSGA-II-S) and the more sophisticated $C$ (i.e., CNSGA-II-NS, CNSGA-II-CD, and CNSGA-II-RR) perform similarly to each other for the RWLPs. Therefore, RWLPs might be not as difficult as researchers expected.

We believe that finding good feasible solutions for *actual* real-world problems is a difficult task due to their complex constraints [4]. In other words, finding good feasible solutions for real-world CMOPs should be more difficult than finding good feasible solutions for the C-DTLZ functions or widely used RWLPs. More complex, difficult benchmark problems are required to evaluate the performance of MOEAs for CMOPs.

## V. Conclusion

We have analyzed the properties of the C-DTLZ functions and the seven RWLPs listed in Table I. The experimental results revealed that the Type-1 and Type-2 C-DTLZ functions have some critical issues and may be inappropriate for benchmarking MOEAs for CMOPs. Researchers in the evolutionary computation community had considered RWLPs to be challenging problems due to their numerous constraints, but we showed that RWLPs might be not as difficult as they had believed. In light of our experimental results, it is possible that constrained multi-objective optimization benchmark problems require careful reconsideration.

There is a great deal of room for designing novel, appropriate problems for benchmarking MOEAs for CMOPs. Among the five C-DTLZ functions, Type-3 functions appear to be one of the most appropriate types of benchmark problems. Therefore, we believe that constructing novel benchmark problems based on Type-3 DTLZ functions is a promising future direction. The PF of the Type-1 and Type-2 C-DTLZ functions is identical to that of the original DTLZ functions, which causes the problem described in this paper. However, the region of the PF of the Type-1 and Type-2 functions can be made infeasible by adding the properties of the Type-3 functions to them. We will construct composite functions of the Type-3 functions and the Type-1 and Type-2 functions.

## Acknowledgment

This research is supported by the HPCI System Research Project "Research and development of multiobjective design exploration and high-performance computing technologies for design innovation" (Project ID:hp160203).

## References


[1] H. K. Singh, T. Ray, and R. A. Sarker, "Optimum Oil Production Planning Using Infeasibility Driven Evolutionary Algorithm," *Evol. Comput.*, vol. 21, no. 1, pp. 65–82, 2013.

[2] S. Z. Martínez, A. A. Montaño, and C. A. C. Coello, "Constrained multi-objective aerodynamic shape optimization via swarm intelligence," in *GECCO*, 2014, pp. 81–88.

[3] H. K. Singh, K. Alam, and T. Ray, "Use of Infeasible Solutions During Constrained Evolutionary Search: A Short Survey," in *ACALCI*, 2016, pp. 193–205.

[4] A. K. Sharma, R. Datta, M. Elarbi, B. Bhattacharya, and S. Bechikh, "Practical Applications in Constrained Evolutionary Multi-objective Optimization," in *Recent Advances in Evolutionary Multi-objective Optimization*, 2017, pp. 159–179.

[5] B. Li, J. Li, K. Tang, and X. Yao, "Many-Objective Evolutionary Algorithms: A Survey," *ACM Comput. Surv.*, vol. 48, no. 1, p. 13, 2015.

[6] E. Mezura-Montes and C. A. C. Coello, "Constraint-handling in nature-inspired numerical optimization: Past, present and future," *Swarm and Evol. Comput.*, vol. 1, no. 4, pp. 173–194, 2011.

[7] H. Jain and K. Deb, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach," *IEEE TEVC*, vol. 18, no. 4, pp. 602–622, 2014.

[8] H. Seada and K. Deb, "A Unified Evolutionary Optimization Procedure for Single, Multiple, and Many Objectives," *IEEE TEVC*, vol. 20, no. 3, pp. 358–369, 2016.

[9] M. Asafuddoula, T. Ray, and R. A. Sarker, "A Decomposition-Based Evolutionary Algorithm for Many Objective Optimization," *IEEE TEVC*, vol. 19, no. 3, pp. 445–460, 2015.

[10] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition," *IEEE TEVC*, vol. 19, no. 5, pp. 694–716, 2015.

[11] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization," *IEEE TEVC*, vol. 20, no. 5, pp. 773–791, 2016.

[12] T. Chugh, K. Sindhya, K. Miettinen, J. Hakanen, and Y. Jin, "On Constraint Handling in Surrogate-Assisted Evolutionary Many-Objective Optimization," in *PPSN*, 2016, pp. 214–224.

[13] N. Srinivas and K. Deb, "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, 1994.

[14] M. Tanaka, H. Watanabe, Y. Furukawa, and T. Tanino, "GA-based decision support system for multicriteria optimization," in *IEEE SMC*, 1995, pp. 1556–1561.

[15] A. Osyczka and S. Kundu, "A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm," *Stru. and Multi. Opt.*, vol. 10, no. 2, pp. 94–99, 1995.

[16] K. Deb, A. Pratap, and T. Meyarivan, "Constrained Test Problems for Multi-objective Evolutionary Optimization," in *EMO*, 2001, pp. 284–298.

[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," Univ. of Essex, Tech. Rep. CES-487, 2008.

[18] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *IEEE CEC*, 2002, pp. 825–830.

[19] C. A. C. Coello and G. T. Pulido, "Multiobjective structural optimization using a microgenetic algorithm," *Stru. and Multi. Opt.*, vol. 30, no. 5, pp. 388–403, 2005.

[20] T. Ray, K. Tai, and C. Seow, "An evolutionary algorithm for multiobjective optimization," *Eng. opt.*, vol. 33, no. 3, pp. 399–424, 2001.

[21] J. Li, Y. Wang, S. Yang, and Z. Cai, "A comparative study of constraint-handling techniques in evolutionary constrained multiobjective optimization," in *IEEE CEC*, 2016, pp. 4175–4182.

[22] S. Venkatraman and G. G. Yen, "A Generic Framework for Constrained Optimization Using Genetic Algorithms," *IEEE TEVC*, vol. 9, no. 4, pp. 424–435, 2005.

[23] Y. G. Woldesenbet, G. G. Yen, and B. G. Tessema, "Constraint Handling in Multiobjective Evolutionary Optimization," *IEEE TEVC*, vol. 13, no. 3, pp. 514–525, 2009.

[24] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE TEVC*, vol. 6, no. 2, pp. 182–197, 2002.

[25] P. Hingston, L. Barone, S. Huband, and R. L. While, "Multi-level Ranking for Constrained Multi-objective Evolutionary Optimisation," in *PPSN*, 2006, pp. 563–572.

[26] W. Gong, Z. Cai, and Z. Li, "An efficient multiobjective differential evolution algorithm for engineering design," *Stru. and Multi. Opt.*, vol. 38, no. 2, pp. 137–157, 2009.

[27] H. K. Singh, A. Isaacs, T. Ray, and W. Smith, "Infeasibility Driven Evolutionary Algorithm (IDEA) for Engineering Design Optimization," in *AI2008*, 2008, pp. 104–115.

[28] T. Ray and K. M. Liew, "A swarm metaphor for multiobjective design optimization," *Eng. opt.*, vol. 34, no. 2, pp. 141–153, 2002.

[29] M. G. Parsons and R. L. Scott, "Formulation of Multicriterion Design Optimization Problems for Solution With Scalar Numerical Optimization Methods," *J. Ship Research*, vol. 48, no. 1, pp. 61–76, 2004.

[30] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE TEVC*, vol. 7, no. 2, pp. 117–132, 2003.

[31] M. López-Ibáñez, J. D. Knowles, and M. Laumanns, "On Sequential Online Archiving of Objective Vectors," in *EMO*, 2011, pp. 46–60.

[32] A. Radulescu, M. López-Ibáñez, and T. Stützle, "Automatically Improving the Anytime Behaviour of Multiobjective Evolutionary Algorithms," in *EMO*, 2013, pp. 825–840.

[33] D. Brockhoff, T. Tran, and N. Hansen, "Benchmarking Numerical Multiobjective Optimizers Revisited," in *GECCO*, 2015, pp. 639–646.

[34] T. Tusar, D. Brockhoff, N. Hansen, and A. Auger, "COCO: The Bi-objective Black Box Optimization Benchmarking (bbob-biobj) Test Suite," *CoRR*, vol. abs/1604.00359, 2016.

[35] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Selecting a small number of representative non-dominated solutions by a hypervolume-based solution selection approach," in *FUZZ-IEEE*, 2009, pp. 1609–1614.

[36] A. Oyama, K. Shimoyama, and K. Fujii, "New Constraint-Handling Method for Multi-Objective and Multi-Constraint Evolutionary Optimization," *T. JSASS*, vol. 50, no. 167, pp. 56–62, 2007.