

Effects of the Number of Constraints on the Performance of Multi-Objective Evolutionary Algorithms

Yuki Tanigaki, Naoki Masuyama, and Yusuke Nojima

Osaka Prefecture University, Sakai, Osaka, JAPAN

Summary

In recent years, there is a great demand for algorithms solving constrained multi-objective optimization problems (CMOPs) in real-world engineering applications. Unlike the box-constrained problems, an additional constraint handling technique (CHT) is required to solve CMOPs. In real-world engineering applications, there are cases in which 50 or more constraints are considered simultaneously. Thus, it is important to examine the behavior of each CHT on a large number of constraints. However, well-known test problems such as CF functions and C-DTLZ functions have only a small number of constraints. One or two constraints are considered among CF functions and all problems included in C1-DTLZ and C2-DTLZ functions have a single constraint. In this context, we propose test problems that can freely change the number of constraints. In designing such test problems, we extend the WFG toolkit, which is proposed for creating box-constrained multi-objective test problems. The performance of a number of popular CHTs is compared on the proposed test problems with up to 100 constraints. We can observe severe performance deterioration by the increase in the number of constraints.

Key words:

Benchmark Problem Framework, Constraint Handling, Evolutionary Multi-Objective Optimization, WFG Toolkit.

1. Introduction

In real-world optimization problems, it is often required to consider multiple objectives simultaneously. A number of evolutionary multi-objective optimization (EMO) algorithms such as MOEA/D [1] and NSGA-II [2] have been proposed for solving multi-objective optimization problems. Among them, solving constrained multi-objective optimization problems (CMOPs) is a challenging task. On the box-constrained multi-objective optimization problems, multi-objective optimization algorithms only consider the upper and lower limits of the decision variable. When these algorithms apply to CMOPs, an additional constraint handling technique (CHT) is required. Various approaches for the hybridization of EMO algorithms and CHTs have been proposed for CMOPs. Constraint dominance principle (CDP) [2] is one of the most popular CHTs proposed by Deb et al. In CDP, a feasible solution always dominates an infeasible one. When infeasible solutions are compared, the solution with

a smaller constraint violation is preferred. CDP is originally introduced with NSGA-II and its successful applications to the other EMO algorithms have been reported [3, 4]. The concept underlying CDP is the separation of constraints and objectives. The objectives and constraints are separately handled. The stochastic ranking [5] and epsilon constraint handling [6] are also classified into this category.

An integration of the constraints into objectives or vice versa is another promising approach. The objective function values of each solution are modified based on its constraint violations in [7]. Angle-based constrained dominance principle (ACDP) utilizes the information of the angle between two solutions compared in the objective space to adjust the dominance relation in CDP [8]. In [9], constraint functions are regarded as objectives. A CMOP with m objectives and p constraints is redefined to an unconstrained problem with $(m + p)$ objectives. Similar mechanisms have been adopted in the constrained optimization by multi-objective genetic algorithms (COMOGA) [10].

When a CMOP has two or more constraints, algorithms need to handle multiple constraints simultaneously. As a simple extension from a single constraint problem, multiple constraint violations are integrated into a single violation measure. For example, CDP prefers the solution having the smaller total violation. In the in-feasibility driven evolutionary algorithm (IDEA) [11], Cai and Wang's method [12], and the adaptive trade-off model [13], an integrated violation measure is treated as an additional objective function instead of p objectives in COMOGA. The critical issue here is how to integrate multiple constraint violations into a single violation measure. One straightforward approach is the sum of its constraint violations such as CDP. The number of the violated constraints and the violation of the most violated constraint are also used.

To evaluate the performance of EMO algorithms, a number of test CMOPs have been proposed. Table 1 summarizes the properties of representative test CMOPs. These CMOPs have a fixed and a small number of constraints. It is, however, often required to deal with a large number of constraints in real-world optimization problems [14–18]. For example, a vehicle structure

design problem includes 54 constraints such as crashworthiness, body torsional stiffness, and robustness in the environment with a low-frequency vibration [18]. Assuming such real-world optimization problems, existing test CMOPs are insufficient for the algorithm development. In this paper, we examine the behavior of the EMO algorithms on CMOPs with a large number of constraints. As test problems, we extend the walking fish group (WFG) toolkit [19] to design a test problem framework that can freely change the number of constraints. The WFG toolkit is a framework for creating unconstrained multi-objective minimization problems with an arbitrary number of objectives. In the proposed framework, both the number of objectives and the number of constraints are scalable. For designing various relationships among the constraints as seen in real-world optimization problems, we adopt three types of constraint functions: (i) a correlation function, (ii) a conflict function, and (iii) a separation function. These functions specify the relationship among an arbitrary number of constraints. The constraint violations are highly correlated and conflicted in the correlation functions and the conflict function, respectively. In the separation function, the constraint violations are highly conflicted and separately violated.

This paper is organized as follows. In Section 2, we explain how our test problems with various characteristics are constructed from the WFG toolkit. Three types of constraint functions are also introduced in Section 2. In Section 3, we discuss the features of CMOPs created using the proposed framework. Experimental results on our test problems are reported in Section 4 where the performance of a number of popular EMO algorithms is examined on the proposed test problem instances with up to 100 constraints. We also show that the effect of the number of constraints on the algorithm performance depends on the relationship among the constraints. Section 5 concludes this paper.

Table 1: Common properties of existing benchmark problems for constrained multi-objective optimization.

Problem	Number of objectives	Number of constraints
CF1-3 [20]	2	1
CF4-7 [20]	2	1 or 2
CF8-10 [20]	3	1
C1-DTLZ [4]	Scalable	1
C2-DTLZ [4]	Scalable	1
C3-DTLZ [4]	Scalable	Same as the number of objectives
BNH [21]	2	1
CTP [22]	2	1
SRN [23]	2	2

2. Proposed Test Problem Framework

Let us consider the following CMOP with m objectives and p constraints:

$$\begin{cases} \text{Minimize } f_i(\mathbf{x}) & (i=1, 2, \dots, m), \\ \text{Subject to } g_j(\mathbf{x}) \geq 0 & (j=1, 2, \dots, p), \end{cases} \quad (1)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ is an n -dimensional vector of decision variables, $f_i(\mathbf{x})$ is the i th objective to be minimized and $g_j(\mathbf{x})$ is the j th constraint. When any constraint is not satisfied, the solution is an infeasible solution. On the other hand, solutions satisfying all constraints are called feasible solutions. If $g_j(\mathbf{x})$ has a negative value, $|g_j(\mathbf{x})|$ can be regarded as the violation of the j th constraint. The constraint violation of all the feasible solutions is zero.

In this section, we describe the configuration of the proposed framework where the number of constraints can be specified in an arbitrary number. First, we explain the detail of the WFG toolkit in subsection 2.1. Subsection 2.2 describes how to implement constraints into the WFG toolkit, and subsection 2.3 shows an example of constraint functions.

2.1 WFG Toolkit

The WFG toolkit [19] is a framework for creating unconstrained multi-objective minimization problems proposed by Huband et al. It consists of transformation functions and shape functions. The transformation functions change the characteristics of each problem instance such as multi-modality and non-separability. The shape functions determine the shape of the Pareto front. Each objective function of the m -objective minimization problem is expressed by the following equations:

$$f_i(\mathbf{x}) = y_m + s_i h_i(y_1, \dots, y_{m-1}), \quad (i = 1, 2, \dots, m), \quad (2)$$

$$y = \begin{cases} (\max(t_m^q, A_i)(t_i^q - 0.5) + 0.5 & (i=1, 2, \dots, m-1), \\ t_m^q & (i=m), \end{cases} \quad (3)$$

$$t^q = (t_1^q, \dots, t_m^q)^T \leftarrow t^{q-1} \leftarrow \dots \leftarrow t^1 \leftarrow \mathbf{x} \quad (4)$$

In (2), $h_i(y)$ is a shape function, s is a set of scale parameters of the objective functions and the vector y is referred to as underlying variables. The range of y is set as $[0, 1]^m$. Among them, y_m is called an underlying distance variable. That is, if y_m is close to zero, the solution is also close to the Pareto front. The underlying variables are calculated by using transformation functions from decision variables as shown in (3) and (4). In (4), t is a transition vector during the transformation from x to y and a symbol

\leftarrow represents a calculation of the transition vector via transformation functions. The underlying variables are calculated from decision variables through a q -step transformation. In (3), A_i controls the dimensionality of the Pareto front. When each A_i is specified as $A_i = 1$, an m -objective problem has an $(m - 1)$ -dimensional Pareto front.

Fig. 1 shows the Pareto front and the search space of WFG1. Since the distance between a solution and the Pareto front is determined by y_m , the search space of all problems created by the WFG toolkit spreads to the upper right.

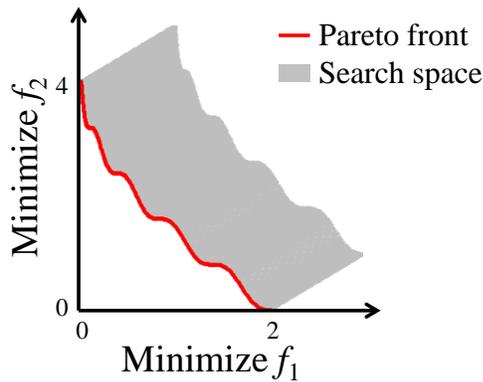


Fig. 1 Search space in WFG1 with two objectives. The gray area shows all feasible solutions of the box-constraint specified in the WFG toolkit.

2.2 Constraint Implementation

All the multi-objective minimization problems created by the WFG toolkit are unconstrained. In the proposed framework, additional constraint functions are employed. The constraint functions are defined as follows:

$$\text{Subject to } g_j(\mathbf{x}) = b_j(y_m) - y_{m+j} \geq 0, \quad (j = 1, 2, \dots, p), \quad (5)$$

$$\mathbf{y} = \begin{cases} (\max(t_m^q, A_i)(t_i^q - 0.5) + 0.5) & (i = 1, 2, \dots, m-1), \\ t_i^q & (i = m, \dots, m+p), \end{cases} \quad (6)$$

$$\mathbf{t}^q = (t_1^q, \dots, t_m^q, \dots, t_{m+p}^q)^T \leftarrow \mathbf{t}^{q-1} \leftarrow \dots \leftarrow \mathbf{t}^1 \leftarrow (\mathbf{x}, \mathbf{x}') \quad (7)$$

where $b_j(y_m)$ is a base function based on the underlying distance variable, $(y_{m+1}, y_{m+2}, \dots, y_{m+p})^T$ are additional underlying variables for constraint functions and $\mathbf{x}' = (x'_1, x'_2, \dots, x'_c)^T$ is a c -dimensional vector of additional decision variables utilized to calculate the underlying constraint variables. For convenience, $(y_{m+1}, y_{m+2}, \dots, y_{m+p})^T$ are labeled as underlying constraint variables. When the

values of the base functions are larger than those of the underlying constraint variables, the constraints are satisfied.

The base functions determine the geometry of the constraint functions in the range of $[-1, 1]$. Fig. 2 shows an example of the landscape of a base function. The feasible and infeasible regions in the objective space of a two-objective WFG1 are shown in Fig. 3. To clarify the feasible and infeasible regions, the underlying constraint variable is specified as zero in the figure in this section. Since there are the band-shaped infeasible regions in the search space, EMO algorithms are required to handle the constraint to converge toward the Pareto front.

The underlying constraint variables control the difficulty to satisfy the constraint. When the j th underlying constraint variable is close to 1, the j th constraint is difficult to be satisfied. Note that the range of the underlying constraint variables is $[0, 1]$. The characteristics of the underlying constraint variables are determined by the transformation functions. If all underlying constraint variables are created by the same transformation function, all constraints have the same difficulty.

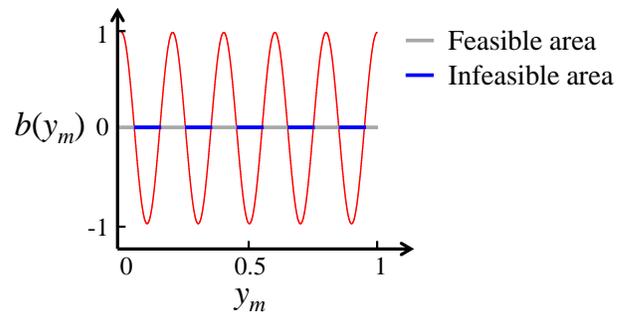


Fig. 2 Landscape of a constraint function: $g(y_m) = \cos(10y_m\pi)$. When the underlying constraint variable is zero, the feasible area and the infeasible area are shown by the gray line and the blue line, respectively.

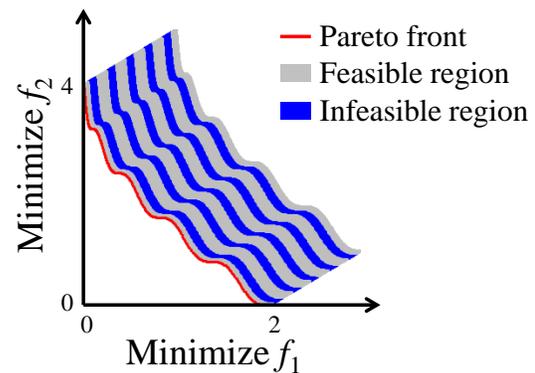


Fig. 3 Search space in WFG1 with a single constraint. Red line shows the Pareto front. Solutions in the gray regions are feasible while those in the blue regions are infeasible.

2.3 Examples of the Base Functions for Creating Constraints

In this subsection, we introduce three types of the base functions having different relations among the constraints where the number of constraints can be arbitrarily specified.

Correlation function:

$$b_j(y_m) = \begin{cases} \cos(wy_m\pi) \geq 0, & \text{if } j=1 \\ \cos\left(wy_m\pi - \frac{1}{(j-1)}\sin(wy_m\pi)\right) \geq 0, & \text{otherwise} \end{cases} \quad (8)$$

where w is a rotation rate that determines the width of the infeasible region. An example of the correlation function when $w = 4$ is shown in Fig. 4. Since the same underlying distance variable y_m maximizes all constraint violations (e.g., 0.25 and 0.75 in the case of Fig. 4), the increase and decrease in the constraint violation are highly correlated.

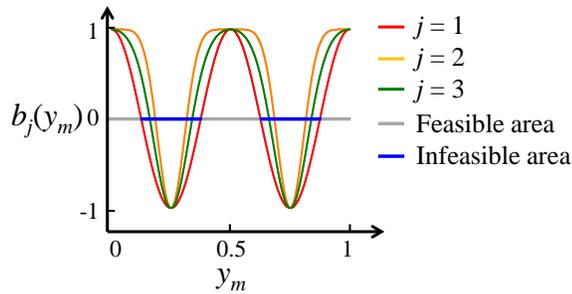


Fig. 4 Landscape of the correlation function with $w = 4$.

Conflict function:

$$b_j(y_m) = \cos\left(wjy_m\pi - \frac{(j-1)}{2}F(wy_m)\pi\right) \geq 0 \quad (9)$$

where $F(wy_m)$ is a triangular function shown in Fig. 5. In conflict function, each constraint violation is maximized by the different y_m value. That is, the increase and decrease of the constraint violation partially conflict each other. For example, the constraint violation of $b_1(y_m)$ increases while one of $b_3(y_m)$ decreases on $0.6 < y_m < 0.7$ in Fig. 6. It should be noted that the first conflict function $b_1(y_m)$ is exactly the same as one of the correlation functions.

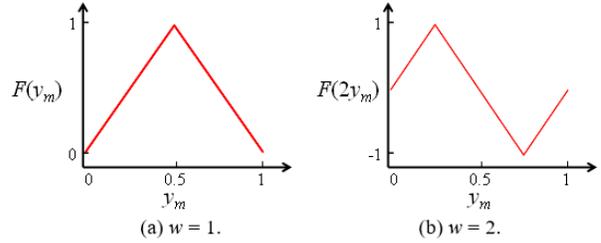


Fig. 5 Triangular function $F(wy_m)$ for the conflict function.

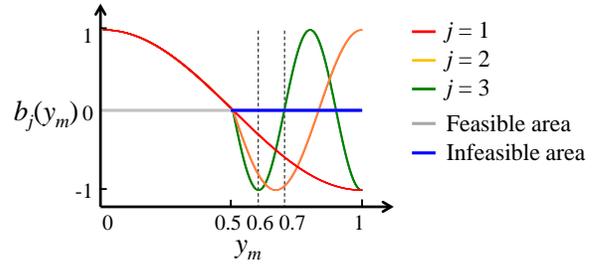


Fig. 6 Landscape of the conflict function with $w = 1$.

Separation function:

$$b_j^p(y_m) = \begin{cases} Conf_p(y_m) \geq 0, & \text{if } j=1 \\ \text{sign}(b_{j-1}^p(y_m))Conf_{p-j+1}(y_m) \geq 0, & \text{otherwise} \end{cases} \quad (10)$$

where $Conf_j(y_m)$ is the j th conflict function and $\text{sign}(\cdot)$ represents the function that extracts the sign of a real number. Fig. 7 shows the separation function when $p = 3$ and $w = 2$. In the correlate function and the conflict function, $b_1(y_m)$ is violated in the entire infeasible region. On the other hand, in the separation function, any constraint function is not always violated. Therefore, EMO algorithms need to handle all constraints for finding feasible solutions.

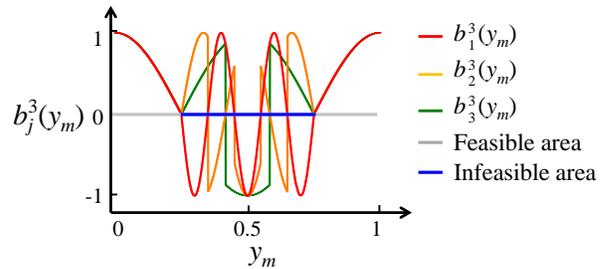


Fig. 7 Landscape of the separation function with $p = 3$ and $w = 2$.

3. The Features of the Proposed Constraint Functions

3.1 The Infeasible Solutions Dominate the Pareto Optimal Solutions

As pointed out in [24], it is possible to obtain the Pareto optimal solutions without considering the constraints when there is no infeasible solution that dominates the Pareto optimal solutions. In the base functions shown in subsection 2.3, all constraint functions have positive values when y_m is 0. That is, the Pareto optimal solutions of the original WFG problems are feasible. In computational experiments, we make the following modification to y_m within the constraint function to avoid these solutions. As shown in Fig. 8, all the original Pareto optimal solutions become infeasible.

$$y'_m = y_m + 0.55/w \tag{11}$$

3.2 The Similarity of the Constraint Functions

The proposed constraint functions are constructed by the base function and the underlying constraint variables. To satisfy the j th constraint, the j th underlying constraint variable should be minimized. The underlying constraint variables are calculated as follows:

$$y_{m+j} = t_{m+j}^q = \left\{ y_m - \left(\sum_{k=1}^c v_k^j \cdot t_{n+k}^{q-1} \right) / \sum_{k=1}^c v_k^j \right\}^2, \tag{12}$$

$(j = 1, \dots, p),$

$$v^j = \{2j, 2(j + 1), \dots, 2(j + c)\}, \tag{13}$$

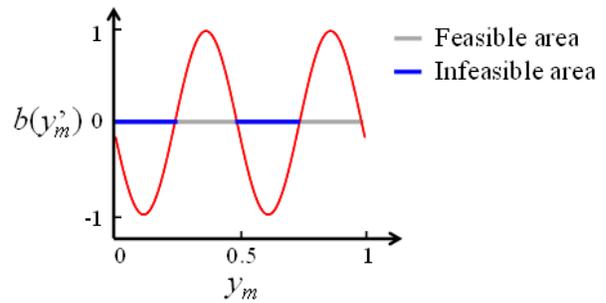
where v^j is the weight vector, n is the number of decision variables utilized to calculate $(y_1, y_2, \dots, y_m)^T$ in the original WFG toolkit, and c is the number of additional decision variables utilized to calculate the underlying constraint variables. The number of elements in v is the same as c . By varying the constants of the weight vector v , the underlying constraint variables are independently calculated.

In this subsection, we examine the similarity of the constraints based on the Spearman's rank correlation coefficient [25]. This metric can be calculated as the rank correlation between the two constraints. Let us assume that a large number of well-distributed solutions are sampled in the decision space. These solutions are sorted in ascending order of the constraint violation. That is, each constraint gives the ranking which starts with rank 1 for the solution with the worst constraint violation. The Spearman

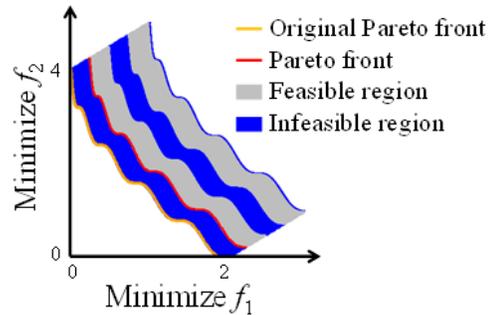
correlation r_s between constraint g_i and g_j is stated as follows:

$$r_s(g_i, g_j) = \frac{cov(rank(g_i), rank(g_j))}{std(rank(g_i))std(rank(g_j))} \tag{14}$$

where $rank(g_i)$ represents the ranking of the solutions for the i th constraint violation and $cov(\cdot)$ is the covariance of the rankings, $std(\cdot)$ represents the standard deviation. In our experiments, we randomly generate 1,000,000 infeasible solutions. The range of the Spearman correlation is $[-1, 1]$. When the Spearman correlation between two constraints will be high, they give a similar ranking. When the Spearman correlation between two constraints will be low, they give a dissimilar ranking. The Spearman correlation among 50 constraints is summarized in Fig. 9. We can observe that the correlation function has similar constraint functions while the other functions have the combinations with dissimilar constraint violations.



(a) Landscape of a constraint function: $g(y'_m) = \cos(4y'_m)$.



(b) Search space in a two-objective WFG1 with $g(y'_m)$.

Fig. 8 Search space in WFG1 with a modification. The Pareto optimal solutions shown by a red line are dominated by the original Pareto front.

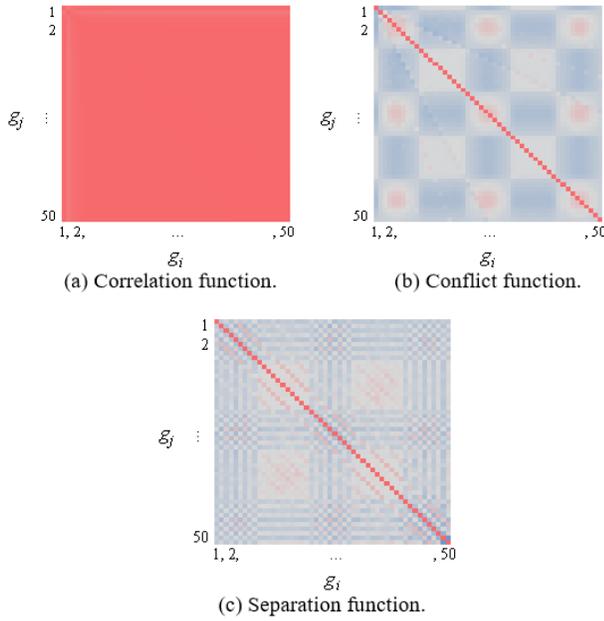


Fig. 9 The Spearman correlation among 50 constraints. Each square represents the correlation between g_i and g_j . For example, the upper right square shows the correlation between g_1 and g_{50} . The square on the diagonal is always one because they show the correlation among the same constraints. Red square and blue square show similar and dissimilar results, respectively.

4. Computational Experiments

In this section, we compare the search performance of the following three representative EMO algorithms on the proposed framework with up to 100 constraints. Each test problem instance is represented by the name of the constraint function and the number of constraints. For example, a constraint problem with two correlation functions is referred to as “Correlation_2”.

NSGA-II-CDP [2]: This algorithm is an extended version of NSGA-II to adopt CDP. In CDP, a feasible solution is always preferred to an infeasible one. In the population update, the feasible solutions are selected first. Then, when the number of feasible solutions is smaller than the population size, infeasible solutions are selected. For the feasible solutions, the fitness evaluation is the same as NSGA-II. For the infeasible solutions, the total violation is utilized as the fitness evaluation.

IDEA [11]: IDEA introduces a ranking-based constraint violation measure where solutions are separately sorted using each constraint violation. Each solution has p ranks. The i th rank represents the ranking position of the solution based on its i th constraint violation value. Then, the final rank is calculated as the sum of the p violation ranks.

In IDEA, this rank is regarded as the additional objective function. More precisely, solutions are compared with the

same evaluation criteria as NSGA-II for the $(m + 1)$ -objective problem. Besides, IDEA explicitly holds an infeasible solution in the current population. The number of infeasible solutions in the current population is determined by a user-defined parameter. That is, all solutions in the population are divided into two subpopulations for the generation update. The first subpopulation only includes feasible solutions and the second subpopulation only includes infeasible ones. In both subpopulations, IDEA optimizes the objective functions and the constraint violation measures simultaneously.

MOEA/D-IEpsilon [26]: MOEA/D-IEpsilon is an extended version of MOEA/D [1] to be able to deal with constraints by embedding an epsilon constraint handling approach [6]. In MOEA/D-IEpsilon, an infeasible solution having the total violation less than the epsilon level is regarded as a feasible solution. The epsilon level is determined based on the maximum total violation in the initial population and the number of feasible solutions in the population. The epsilon level basically decreases at every generation update. Finally, the epsilon level equals to zero at T_c th generation. That is, MOEA/D-IEpsilon always prefers feasible solutions after T_c th generation.

4.1 Parameter Setting

We use the following parameter specifications in all the three EMO algorithms.

Population size: 105,

Crossover: SBX crossover [27],

Mutation: Polynomial mutation [28],

Termination condition: 250 generations,

Ratio of infeasible solutions in IDEA: 20%,

Scalarizing function in MOEA/D-IEpsilon: Tchebycheff,

T_c in MOEA/D-IEpsilon: 200 generations.

The detailed parameters of the framework are listed as follows:

Number of objectives m : 3,

Number of decision variables n : 24,

Number of additional decision variables utilized to calculate the underlying constraint variables c : 5,

Rotation rate w for constraint functions: 10,

Number of constraints p : 2, 10, 50, 100.

In [19], nine problems called WFG1-WFG9 have been proposed. Here, we extend WFG1 with a small modification. In WFG1, randomly generated underlying variables y are biased with the following transformation function.

$$b_poly(y, \alpha) = y^\alpha, \quad (15)$$

where α is a parameter controlling the distribution of underlying variables. If α is close to zero, a randomly generated solution tends to be far from the Pareto front. The original WFG1 uses 0.02. We specify α to 0.25 for the clear performance comparison. Fig. 10 shows randomly generated 105 solutions on WFG1. The randomly generated solutions with $\alpha = 0.25$ are widely spread.

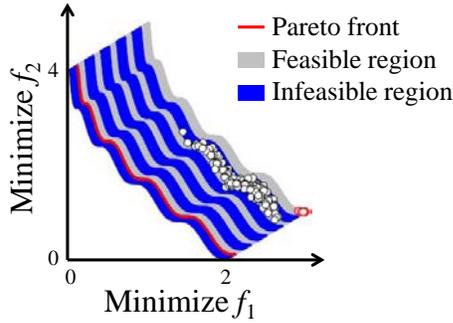


Fig. 10 Randomly generated 105 solutions. Black circles show the solutions with $\alpha = 0.25$ and red circles show the solutions with $\alpha = 0.02$.

4.2 Experimental Results and Discussions

The average inverse generational distance (IGD) values over 100 runs are shown in Fig. 11. The IGD metric measures the convergence and diversity of the obtained solutions [29]. The detailed definition is as follows:

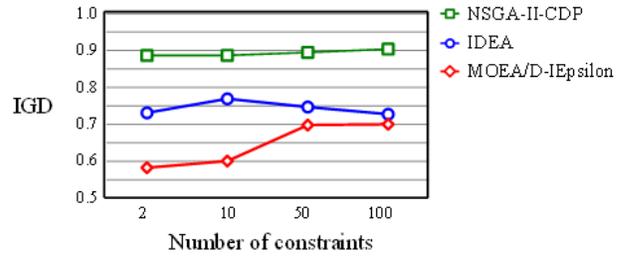
$$IGD(Z, A) = \frac{1}{|Z|} \sum_{i=1}^{|Z|} \min_{j=1}^{|A|} d(z_i, a_j), \tag{16}$$

where $d(z_i, a_j)$ is the Euclidean distance between z_i and a_j . z_i is a reference point on the Pareto front and a_j is a non-dominated solution in the final population. We uniformly sampled 10,000 points for the reference points.

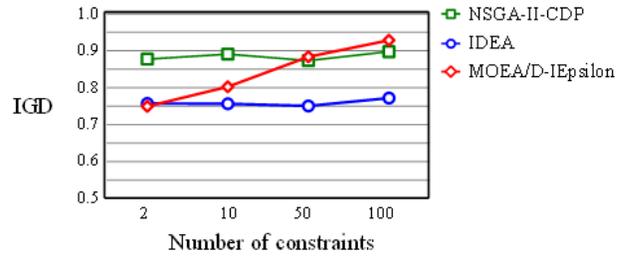
From Fig. 11, the performance of MOEA/D-IEpsilon is good at the instances with two constraints. However, the performance of MOEA/D-IEpsilon is severely degraded by the increase in the number of constraints. Among them, MOEA/D-IEpsilon does not work well on the conflict function and the separation function with a large number of constraints. The dissimilar constraint violations may have a negative effect to control the epsilon level. The worst performance is obtained by NSGA-II-CDP for almost all instances except Conflict_50 and Conflict_100. A feasible solution is always preferred to an infeasible solution in CDP. In the proposed framework, EMO algorithms are required to pass through the infeasible region to converge toward the Pareto front. Thus, no selection pressure is given toward the Pareto front when all solutions in the current population are feasible. We can also observe that

performance deterioration of IDEA by the increase in the number of constraints is relatively small.

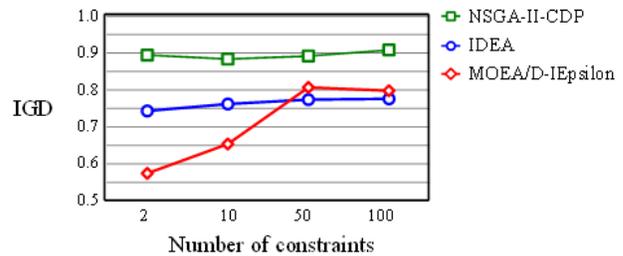
In Fig. 12, we show the history of the underlying distance variable of feasible solutions included in the population for the instances with 100 constraints. MOEA/D-IEpsilon well converge toward the Pareto front in the initial generations. This observation suggests the strong convergence property of the epsilon constraint handling approach.



(a) Average IGD on the correlation function.



(b) Average IGD on the conflict function.



(c) Average IGD on the separation function.

Fig. 11 Average IGD values on the proposed framework with up to 100 constraints.

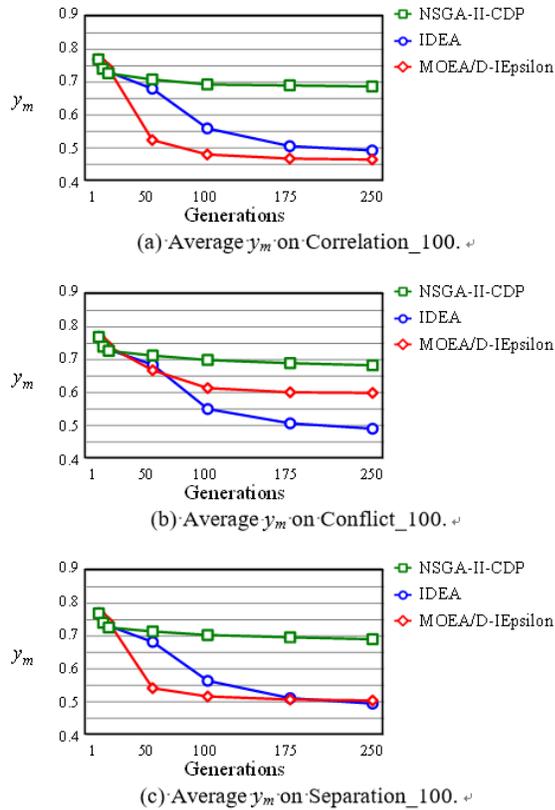


Fig. 12 The history of the underlying distance variable of the feasible solution in the current population on the instances with 100 constraints.

Next, we consider the integration techniques of multiple constraints into a single violation measure. In NSGA-II-CDP and MOEA/D-Iepsilon, the summation of constraint violation is utilized as a violation measure. In IDEA, the rank-based measure is adopted. Since many existing EMO algorithms employ a single violation measure, the behavior of such algorithms is affected by how to integrate multiple constraints. Here, we examine the effect of the following integration techniques with IDEA.

- (i) The summation of constraint violations (Sum).
- (ii) The violation of the most violated constraint (Max).
- (iii) The ranking position in the rank-based measure of IDEA (Rank).

These measures are regarded as an additional objective function of IDEA. We also examine the performance of IDEA when the constraints are considered as the objectives. That is, IDEA solves an unconstrained problem with $(m + p)$ objectives in the following way.

- (iv) The conversion of the constraints to the objectives (CtoO).

Table 2 shows the average IGD values over 100 runs together with the standard deviation on the instances with 2 and 100 constraints. The worst result for each instance is

shown by bold face. All the IGD values of IDEA with CtoO deteriorate due to the increase in the number of constraints. This is because 102 objectives are optimized when CtoO applied to a CMOP with 100 constraints. As reported in [30], the performance of EMO algorithms is often severely degraded by the increase in the number of objectives. There is no clear performance deterioration of IDEA with the other integration techniques. An interesting observation is that there is no clear performance deterioration of IDEA with Sum while the performance of MOEA/D-Iepsilon, where Sum is utilized as a violation measure, is severely degraded by the increase in the number of constraints in Fig. 11. This observation suggests that the epsilon constraint handling approach is not suitable to handle a large number of constraints.

Table 2: The IGD by IDEA with the different integration techniques. Standard deviation is shown in parentheses.

Instance	Sum	Max	Rank	CtoO
Correlation_2	0.731 (0.111)	0.759 (0.111)	0.731 (0.088)	0.712 (0.113)
Correlation_100	0.736 (0.103)	0.755 (0.157)	0.728 (0.089)	0.798 (0.155)
Conflict_2	0.748 (0.133)	0.770 (0.089)	0.758 (0.109)	0.775 (0.127)
Conflict_100	0.752 (0.109)	0.756 (0.105)	0.772 (0.127)	1.002 (0.222)
Separation_2	0.750 (0.125)	0.783 (0.099)	0.743 (0.130)	0.746 (0.113)
Separation_100	0.756 (0.117)	0.729 (0.126)	0.776 (0.123)	1.050 (0.191)

4.3 Comparison of the Proposed Constraint Functions to the Real-World Engineering Design Optimization Problem

In this subsection, we verify the validity of the framework by comparing the difficulty of the three constraint functions to the vehicle structure design problem (VSDP) [18]. VSDP has 222 design variables, two objective functions, and 54 constraints. The design variables represent the thickness of the parts of the vehicle. These parts are used in three vehicle models. The two objective functions are the minimization of the total weight of the three models and the maximization of the number of common thickness parts. The constraints include the crashworthiness, body torsional stiffness, and robustness in the low-frequency vibration modes.

First, we show the Spearman correlation among the constraints of VSDP in Fig. 13. The Spearman correlation among 54 constraints of VSDP is calculated in the same way in Section 3. The Spearman correlation of some constraints is not shown. This is because they are not violated in all 1,000,000 infeasible solutions. The Spearman correlation cannot be calculated for these constraints. We can observe that most of the combinations of the constraints in VSDP are dissimilar. From the viewpoint of the Spearman correlation, the conflict

function and the separation function may be suitable to investigate the performance against VSDP.

We summarize the average hypervolume over 35 runs of each population in Table 3. Following the experimental settings in [31], the population size is 100 and the stopping criterion is set by 30,000 solution evaluations. Other experimental settings are the same in subsection 4.1. Hypervolume is the size of the dominated region by a set of solutions in the objective space [32]. For the hypervolume calculation, the objective functions are normalized as recommended in [31]. The best result is shown by bold face. In Fig. 14, we also show all solutions during the search iterations in a single run of each EMO algorithm on VSDP. The gray circles and the blue circles represent feasible solutions and infeasible solutions, respectively.

In Table 3, the best hypervolume is obtained by IDEA. The performance of MOEA/D-IEpsilon is not good. This observation is consistent with the results on Conflict_50, Conflict_100, Separation_50, and Separation_100. From Fig. 14, we can see that the infeasible solutions generated by IDEA are widely spread. This observation may suggest the potential usefulness of maintaining the infeasible solutions in the population. On the other hand, well-converged solutions are obtained by NSGA-II-CDP while the low convergence ability of NSGA-II-CDP is shown on our framework. This is because the existence of feasible solutions keeps the selection pressure toward the Pareto front in VSDP.

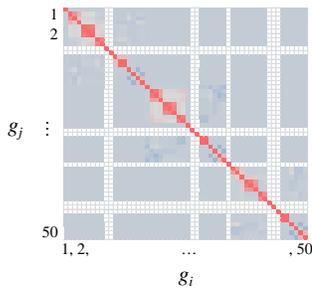


Fig. 13 The Spearman correlation among 54 constraints of VSDP. Red square and blue square show similar and dissimilar results, respectively.

Table 3: The average hypervolume over 35 runs on VSDP. Standard deviation is shown in parentheses.

NSGA-II-CDP	IDEA	MOEA/D-IEpsilon
0.081	0.105	0.069
(0.014)	(0.031)	(0.015)

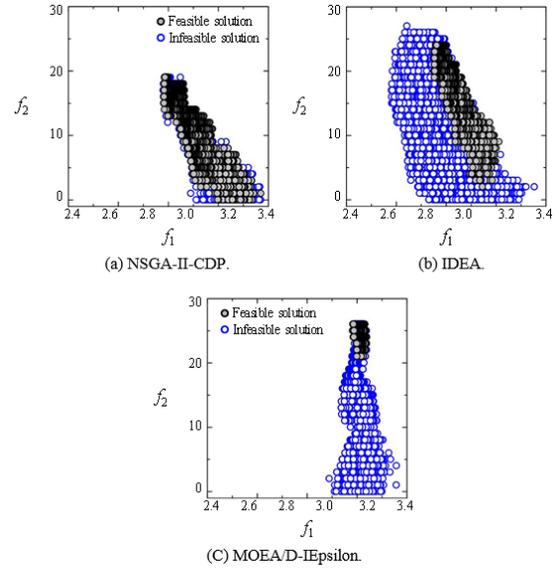


Fig. 14 All individuals during the search iterations on VSDP. The first objective is minimized and the second objective is maximized.

5. Conclusion

In this paper, we proposed a framework that creates CMOPs. The framework is aimed to generate test problems where the number of constraints can be arbitrarily set. In existing benchmark CMOPs, the number of constraints is fixed. Three types of relationships among the constraints are also specified in the framework. Through computational experiments with three representative constrained EMO algorithms, we examined the effect of the number of constraints on the performance of EMO algorithms.

From the experimental results on the framework with 2 to 100 constraints, we obtained the following interesting observations: (i) the search ability of MOEA/D-IEpsilon was degraded by the increase in the number of constraints, (ii) the performance of MOEA/D-IEpsilon severely deteriorated on the problems where the fitness landscapes of the constraint violations are dissimilar, (iii) IDEA showed the stable performance regardless of the violation measure to be utilized as an additional objective, and (iv) NSGA-II-CDP did not pass through the infeasible region to converge toward the Pareto front. We also demonstrated that the performance of MOEA/D-IEpsilon and IDEA on the real-world engineering design optimization problem was consistent with the results on the framework when the fitness landscapes of the constraint violations for the framework were relatively similar to the real-world optimization problem.

Whereas we can obtain these observations about the search behavior of each EMO algorithm from the proposed framework, we need further research on the implementation of the constraints. For example, the shape of the infeasible region and the dependency of the constraint to the decision variables are important issues for discussing the characteristics of EMO algorithms.

Acknowledgments

This work was supported by MEXT Development of Innovative Design and Production Processes that Lead the Way for the Manufacturing Industry in the Near Future through Priority Issue on Post-K computer.

References

- [1] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, 2007.
- [2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.
- [3] M.A. Jan and R.A. Khanum, "A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D," *Applied Soft Computing*, vol. 13, no. 1, pp. 128-148, 2013.
- [4] 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 Trans. on Evolutionary Computation*, vol.18, no. 4, pp. 602-622, 2014.
- [5] T.P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Trans. on Evolutionary Computation*, vol. 4, no. 3, pp. 284-294, 2000.
- [6] M. Asafuddoula, T. Ray, R. Sarker, and K. Alam, "An adaptive constraint handling approach embedded MOEA/D," *Proc. of 2012 IEEE Congress on Evolutionary Computation (CEC)*, 8 pages, 2012.
- [7] Y.G. Woldesenbet, G.G. Yen, and B.G. Tessema, "Constraint handling in multiobjective evolutionary optimization," *IEEE Trans on Evolutionary Computation*, vol. 13, no. 3, pp. 514-525, 2009.
- [8] Z. Fan, W. Li, X. Cai, K. Hu, H. Lin, and H. Li, "Angle-based constrained dominance principle in MOEA/D for constrained multi-objective optimization problems," *Proc. of 2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 460-467, 2016.
- [9] D.A.G. Vieira, R.L.S. Adriano, L. Krahenbuhl, and J.A. Vasconcelos, "Handling constraints as objectives in a multiobjective genetic based algorithm," *Journal of Microwaves and Optoelectronics*, vol. 2, no. 6, pp. 50-58, 2002.
- [10] P.D. Surry and N.J. Radcliffe "The COMOGA method: constrained optimisation by multi-objective genetic algorithms," *Control and Cybernetics*, vol. 26, pp. 391-412, 1997.
- [11] T. Ray, H. K. Singh, A. Isaacs, and W. Smith, "Infeasibility driven evolutionary algorithm for constrained optimization," *Constraint-handling in Evolutionary Optimization*, pp. 145-165, Springer, 2009.
- [12] Z. Cai and Y. Wang, "A multiobjective optimization-based evolutionary algorithm for constrained optimization," *IEEE Trans on Evolutionary Computation*, vol. 10, no. 6, pp. 658-675, 2006.
- [13] Y. Wang, Z. Cai, Y. Zhou, and W. Zeng, "An adaptive tradeoff model for constrained evolutionary optimization," *IEEE Trans on Evolutionary Computation*, vol. 12, no. 1, pp. 80-92, 2008.
- [14] R.R. de Lucena, B.S.L.P. de Lima, B.P. Jacob, and D.M. Rocha, "Optimization of pipeline routes using an AIS/adaptive penalty method," *Proc. of Eighth International Conference on Engineering Computational Technology*, Paper 66, 7 pages, 2012.
- [15] M. Liao, Y. Zhou, Y. Su, Z. Lian, and H. Jiang, "Dynamic analysis and multi-objective optimization of an offshore drilling tube system with pipe-in-pipe structure," *Applied Ocean Research*, vol. 75, pp. 85-99, 2018.
- [16] K.S. Zhang, Z.H. Han, Z.J. Gao, and Y. Wang, "Constraint aggregation for large number of constraints in wing surrogate-based optimization," *Structural and Multidisciplinary Optimization*, 18 pages, 2018.
- [17] G.K.W. Kenway and J.R.R.A. Martins, "Multipoint high-fidelity aerostuctural optimization of a transport aircraft configuration," *Journal of Aircraft*, vol. 51, no. 1, pp. 144-160, 2014.
- [18] T. Kohira, H. Kemmotsu, A. Oyama, and T. Tatsukawa, "Proposal of benchmark problem based on real-world car structure design optimization," *Companion of the Genetic and Evolutionary Computation Conference (GECCO) 2018*, pp. 183-184, 2018.
- [19] S. Huband, L. Barone, L. While, and P. Hington, "A scalable multi-objective test problem toolkit," *Lecture Notes in Computer Science: Evolutionary Multi-Criterion Optimization*, vol. 3410, pp. 280-295, 2005.
- [20] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," *University of Essex, Colchester, UK and Nanyang technological University, Singapore, Tech. Rep.*, vol. 264, 2008.
- [21] T.T. Binh and U. Korn, "MOBES: A multiobjective evolution strategy for constrained optimization problems," *Proc. of the Third International Conference on Genetic Algorithms*, vol. 25, pp. 176-182, 1997.
- [22] K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, 2001.
- [23] N. Srinivas and K. Deb, "Multi-objective function optimization using non-dominated sorting genetic algorithms," *IEEE Trans. on Evolutionary Computation*, vol. 2, no. 3, pp. 221-248, 1994.
- [24] R. Tanabe and A. Oyama, "A note on constrained multi-objective optimization benchmark problems," *Proc. of 2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1127-1134, 2017.

- [25] J.H. Zar, "Significance testing of the spearman rank correlation coefficient," *Journal of the American Statistical Association*, vol. 67, no. 339, pp. 578-580, 1972.
- [26] Z. Fan, W. Li, X. Cai, H. Li, H. Huang, and Z. Cai, "An improved epsilon constraint handling method embedded in MOEA/D for constrained multiobjective optimization problems," *Proc. of 2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, 8 pages, 2016.
- [27] K. Deb and A. Kumar, "Real-coded genetic algorithms with simulated binary crossover: studies on multimodal and multiobjective problems," *Complex Systems*, vol. 9, no. 6, pp. 431-454, 1995.
- [28] M. Hamdan, "On the disruption-level of polynomial mutation for evolutionary multi-objective optimization algorithms," *Computing and Informatics*, vol. 29, pp. 783-800, 2010.
- [29] P.A. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. on Evolutionary Computation*, vol. 7, no. 2, pp. 174-188, 2003.
- [30] R.C. Purshouse and P.J. Fleming, "Evolutionary many-objective optimization: An exploratory analysis," *Proc. of 2003 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2066-2073, 2003.
- [31] <https://ladse.eng.isas.jaxa.jp/benchmark/index.html>, Accessed on: Dec. 2018.
- [32] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms – A comparative case study," *Lecture Notes in Computer Science: Parallel Problem Solving from Nature*, vol. 1498, pp. 292-301, 1998.



Yusuke Nojima received the B.S. and M.S. degrees in mechanical engineering from Osaka Institute of Technology, Osaka, Japan, in 1999 and 2001, respectively, and the Ph.D. degree in system function science from Kobe University, Hyogo, Japan, in 2004. Since 2004, he has been with Osaka Prefecture University, Osaka, Japan, where he was a Research Associate and is currently an Associate Professor in Department of Computer Science and Intelligent Systems. His research interests include evolutionary machine learning and evolutionary multiobjective optimization.



Yuki Tanigaki received the B.S. and M.S. degree in computer science and intelligent systems from Osaka Prefecture University, Sakai, Osaka, Japan, in 2014 and 2016, respectively. He is currently a Ph.D course student with the Department of Computer Science and Intelligent Systems, Osaka Prefecture University. His research interests include evolutionary multi-objective optimization and many-objective optimization.



Naoki Masuyama graduated from Nihon University in 2010, received the M.E. degree from Tokyo Metropolitan University in 2012, and he obtained the Ph.D. degree from Faculty of Computer Science and Information Technology, University of Malaya, Malaysia in 2016. He is currently an Assistant Professor in Graduate School of Engineering, Osaka Prefecture University, Japan. His research interests include data mining and machine learning.