MOEA/D with Feasibility-based Weight Adjustment for Constrained Optimization

Yusuke Yasuda[1], Wataru Kumagai[2], Kenichi Tamura[1], and Keiichiro Yasuda[1]
[1]*Department of Electrical Engineering and Computer Science, Tokyo Metropolitan University*
1-1, Minamiosawa, Hachioji-shi, Tokyo 192-0397, Japan
[2]*Innovation Center, Market Headquarter, Yokogawa Electric Corporation, Ltd*
2-9-32, Naka-cho, Musashino-shi, Tokyo 180-0006, Japan
Email: yasuda-yusuke@ed.tmu.ac.jp

***Abstract*—Although metaheuristics are mainly focused on unconstrained optimization, real-world applications require its extension to constrained optimization by maintaining and utilizing direct search and multi-point search. Constraint handling techniques for metaheuristics are known to use constraint violation as an additional objective function, but in general, they utilize Pareto ranking, which leaves issues of feasibility and convergence of the obtained solution. To address this issue, this paper extends MOEA/D, a multi-objective optimization method using a scalarization framework, to constrained optimization and proposes an adaptive weight adjustment method suitable for constrained optimization. Finally, we verify the usefulness of the proposed method through numerical experiments.**

***Index Terms*—constrained optimization, metaheuristics, constraint handling technique, MOEA/D, adaptation**

I. Introduction

In recent years, as systems become larger and more complex, metaheuristics have attracted attention as a framework for practical optimization methods [1]. Metaheuristics are classified as direct search methods that do not require analytical information such as the gradient of the objective function, and they are highly versatile in that they can be optimized even when the objective function is not explicitly formulated. In addition, most metaheuristics are heuristic search strategies based on the interaction of multiple individuals, which makes it difficult to fall into local optimal solutions even for multimodal functions. On the other hand, metaheuristics are designed primarily for unconstrained optimization and thus require Constraint Handling Techniques (CHTs) to be applied to constrained optimization problems [2]. Therefore, there is a need to develop CHTs to extend metaheuristics to constrained optimization while retaining and utilizing features of metaheuristics such as direct search and multi-point search.

Many CHTs for metaheuristics have been proposed so far [2], [3]. The penalty-based approaches, which are typical CHT, add the constraint violation to the objective function and transform it into an unconstrained optimization problem. While the penalty-based approaches are easy to implement because it only requires replacing the objective function, they are known to be difficult to set and adjust the appropriate penalty coefficients to obtain a good feasible solution because they are highly problem-dependent. Other methods deal with constraints by transforming the problem into a multi-objective optimization problem, where constraint violation is an additional objective function [6]–[9]. Since the multi-objective-based approaches can simultaneously optimize the objective function and constraint violation, they are expected to have high global optimization performance even when the feasible region is a non-convex set and there are many local optimal solutions due to constraints [3]. However, since Pareto ranking [10] is generally used [6], [7], there are some issues such as excessive diversity in the search for the infeasible region and low feasibility and convergence of the obtained solution [9].

In this paper, we focus on the framework of the multi-objective-based approach, which can maintain robustness even in the feasible region of a non-convex set, and extend a Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D) [11] to constrained optimization with adaptive weight adjustment to improve the feasibility and convergence of the obtained solution. Since MOEA/D obtains various Pareto solutions by dividing a multi-objective optimization problem into several single-objective optimization problems using a scalarization function with different weights, it is expected to improve convergence to a good feasible solution by appropriately adjusting the weights according to the problem.

The contribution of this paper is to verify the usefulness of the proposed method in more detail, based on the authors' previous work [9]. To this end, numerical experiments with an increased number of problems and dimensions are conducted to verify the search performance and characteristics of the proposed method.

II. Constrained Optimization

A. Constrained Optimization Problems

In this paper, we focus on the constrained optimization problem defined by (1).

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^N} \quad & f(\boldsymbol{x}) \\ \text{subj. to} \quad & g_k(\boldsymbol{x}) \leq 0,\ k=1,2,\cdots,K \\ & a_j \leq x_j \leq b_j,\ j=1,\cdots,N \end{aligned} \tag{1}$$

Let $f(\boldsymbol{x})$ be the objective function, $g_k(\boldsymbol{x}),\ k=1,\cdots,K$ be the K inequality constraint function, and $f, g_k : \mathbb{R}^N \rightarrow \mathbb{R}$ be the mapping. Let $[a_j, b_j],\ j=1,\cdots,N$ be the upper and lower constraints imposed on each element x_j of the decision variable The region $\mathcal{F}$ satisfying all constraints is called the feasible region $\mathcal{F}$, and a solution contained in the feasible region $\mathcal{F}$ is called a feasible solution, otherwise it is called an infeasible solution.

When $g_k(\boldsymbol{x}^o) = 0$ holds for some solution $\boldsymbol{x}^o$, $g_k(\boldsymbol{x}^o)$ is called an active constraint and $\boldsymbol{x}^o$ is on the boundary of that constraint. In constraint optimization, it is reported that it is common for the global optimal solution $\boldsymbol{x}^*$ to lie on the boundary of the feasible region [2], in which case at least one constraint in $\boldsymbol{x}^*$ is an active constraint.

B. Constraint Handling Techniques

Many CHTs for metaheuristics have been proposed in the past [2], and in recent years, attempts have been made to improve search performance by utilizing infeasible solutions in the search [3].

The penalty-based approaches transform a constrained optimization problem into an unconstrained optimization problem by adding a penalty term to the objective function [2], [3]. The penalty-based approaches perform a search while indirectly taking advantage of infeasible solutions by creating artificial landscapes in the infeasible region. However, if the penalty coefficients are not set and adjusted appropriately, feasible solutions may not be obtained or local optimal solutions may result due to constraints. Although the dynamic and adaptive methods [4], [5] for adjusting penalty coefficients have been proposed, there is currently no general-purpose strategy because the appropriate penalty coefficients are problem-dependent.

The multi-objective-based approaches transform a constrained optimization problem into a multi-objective optimization problem by considering constraint violation as additional objective functions [3]. The multi-objective-based approaches can efficiently find the global optimal solution that exists on the boundary of the feasible region because the search can be performed while directly utilizing infeasible solutions, and it can achieve high global optimization performance even when there are many local optimal solutions caused by constraints [3], [7]. However, Pareto ranking [10] is commonly used in multi-objective-based approaches [6], [7], which leads to excessive search diversity in infeasible regions. There is a possibility that a feasible solution may not be obtained.

Venkatraman et al. [6] proposed a method in which the search is divided into a first and second stage, in the first stage only the constraint violation is optimized to obtain a feasible solution, and once a feasible solution is obtained, the search moves to the second stage to simultaneously optimize the objective function value and constraint violation through Pareto ranking. Ray et al. [7] proposed an Infeasibility Driven Evolutionary Algorithm (IDEA) that retains both feasible and infeasible solutions by setting the number of infeasible solutions retained in the population as a parameter and performs survival selection by Pareto ranking of objective function values for the feasible solution set and objective function values and constraint violation for the infeasible solution set. These methods can be harmful if the feasible region is small compared to the search space, as survival selection by Pareto ranking rarely works, or no feasible solution can be obtained in the first place.

On the other hand, Wang et al. [8] proposed a new multi-objective-based approach using MOEA/D, which attempts to converge to good feasible solutions from various searches including infeasible regions by dynamically adjusting weights within relaxed constraints. The authors have also proposed a method based on MOEA/D [9], and have confirmed better search performance than conventional methods using Pareto ranking for problems where the feasible region is a convex set.

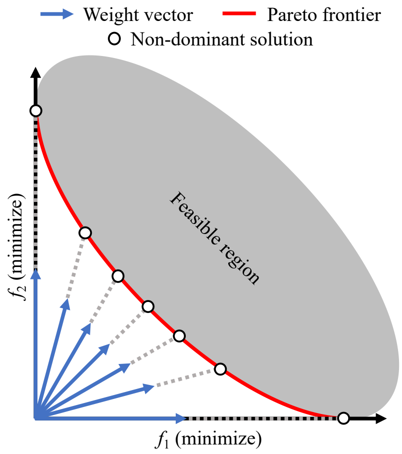


Fig. 1. MOEA/D search mechanism.

III. MOEA/D

A. Overview of MOEA/D

MOEA/D [11] was proposed by Zhang et al. and is one of the leading multi-objective optimization methods in recent years. The blue arrows represent the weight vectors and the red curve represents the Pareto frontier in Fig. 1. The scalarization function $S(\boldsymbol{x}^i|\boldsymbol{w}^i)$, which uses different weight vectors $\boldsymbol{w}^i,\ i = 1, \cdots, m$, divides the multi-objective optimization problem into several single-objective optimization problems. Furthermore, T neighborhood $B(i) = \{i_1, \cdots, i_T\}$ based on the Euclidean distance between the weight vectors for the i-th individual is defined, and multiple single-objective optimization problems are solved simultaneously by performing evolution operations within the neighborhood. The pseudo-code of MOEA/D is shown in **Algorithm 1**.

The weight vector is given by (3) to satisfy (2).

$$\sum_{r=1}^{R} w_r = 1, \quad w_r \geq 0 \tag{2}$$

$$w_r \in \left\{0, \frac{1}{H}, \frac{2}{H}, \cdots, \frac{H}{H}\right\} \tag{3}$$

Note that $H \in \mathbb{N}$ and R denote the objective number. The weight vector is generated equal to the number of individuals m and is given by $m = {}_{H+R-1}C_{R-1}$. For example, if $R = 2$, then $m = {}_{H+1}C_1$ and $H + 1$ weight vectors are generated.

In MOEA/D, scalarization functions play an important role in approximating the Pareto frontier. Typical scalarization functions include weighted sum, weighted Chebyshev, and Penalty-based Boundary Intersection (PBI), but the best scalarization function depends on the shape of the Pareto frontier [11]. This paper employs weighted sums as a basic

Algorithm 1 MOEA/D

procedure MOEA/D(m,T,$g_{\max}$)
  Initialize $\boldsymbol{x}^{i,(1)}$ $(i=1,\cdots,m)$.
  Set $\boldsymbol{w}^{i,(1)}$ $(i=1,\cdots,m)$ by (3).
  $g := 1$
  while $g \leq g_{\max}$ **do**
    for $i=1$ **to** m **do**
      Calculate the Euclidean distance between $\boldsymbol{w}^{i,(1)}$ and $\boldsymbol{w}^{j,(1)}$ $(j=1,\cdots,m)$.
      Set the neighborhood index $B(i)=\{i_1,\cdots,i_T\}$.
    end for
    for $i=1$ **to** m **do**
      Randomly select $a,b \in B(i)$.
      Crossover and mutation on $\boldsymbol{x}^{a,(g)}$ and $\boldsymbol{x}^{b,(g)}$ to produce $\boldsymbol{y}^{i,(g)}$.
      for $j \in B(i)$ **do**
        if $S(\boldsymbol{y}^{i,(g)}|\boldsymbol{w}^{j,(g)}) \leq S(\boldsymbol{x}^{j,(g)}|\boldsymbol{w}^{j,(g)})$ **then**
          $\boldsymbol{x}^{j,(g)} := \boldsymbol{y}^{i,(g)}$
        end if
      end for
    end for
    $g := g+1$
  end while
  return $\boldsymbol{x}^{i,(g_{\max})}$ $(i=1,\cdots,m)$
end procedure

study to extend to constrained optimization. The weighted sum scalarization function $S^{\mathrm{WS}}(\boldsymbol{x}|\boldsymbol{w})$ is given by (4).

$$S^{\mathrm{WS}}(\boldsymbol{x}|\boldsymbol{w}) = \sum_{r=1}^{R} w_r f_r(\boldsymbol{x}) \tag{4}$$

B. Extension to Constrained Optimization

MOEA/D is designed to find Pareto solutions with excellent convergence and diversity for multi-objective optimization problems and cannot be applied directly to constrained optimization problems. However, it can be applied in the same way as in a multi-objective optimization problem by the constraint violation as an additional objective function [6]–[9].

The total amount by which a solution $\boldsymbol{x}$ deviates from the constraints is defined as the constraint violation $v(\boldsymbol{x})$ and is transformed into a bi-objective optimization problem defined by the objective function $f(\boldsymbol{x})$ $(= f_1(\boldsymbol{x}))$ and the constraint violation $v(\boldsymbol{x})$ $(= f_2(\boldsymbol{x}))$. By applying MOEA/D to this problem, we expect to obtain a Pareto solution set (Pareto frontier) in the f-v space in constrained optimization, represented by the red curve in Fig. 2.

The constraint violation $v(\boldsymbol{x})$ can be flexibly defined, but (5) that simply makes it the sum of the deviations or (6) that normalizes by the individual $\boldsymbol{x}^i \in X;\ i=1,\cdots,m$ is possible.

$$v(\boldsymbol{x}) = \sum_{k=1}^{K} \Omega_k(\boldsymbol{x}) \tag{5}$$

$$v^{\mathrm{nor}}(\boldsymbol{x}) = \sum_{k=1}^{K} \frac{\Omega_k(\boldsymbol{x}) - \min_{\boldsymbol{x}' \in X} \Omega_k(\boldsymbol{x}')}{\max_{\boldsymbol{x}' \in X} \Omega_k(\boldsymbol{x}') - \min_{\boldsymbol{x}' \in X} \Omega_k(\boldsymbol{x}')} \tag{6}$$

Note that $\Omega_k(\boldsymbol{x}) = \max\{0, g_k(\boldsymbol{x})\}$.

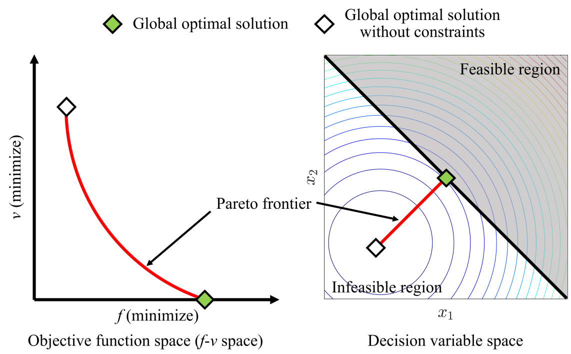


Fig. 2. Pareto frontier in constrained optimization.

IV. Proposed Method

This chapter builds on MOEA/D, which has been extended to constrained optimization and proposes MOEA/D with adaptive weight adjustment for constrained optimization to improve convergence to a good feasible solution. The pseudo-code of the proposed method is shown in **Algorithm 2** and described in detail in the following sections.

A. Variable Weights by the Parameter α

In most constrained optimization, the global optimal solution is located on the constraint boundary (the edge of the Pareto frontier in Fig. 2), so the region with good objective function values near the constraint boundary is a promising region. Therefore, it is appropriate to give the weight vector corresponding to the solution biased toward the feasible region, rather than the entire Pareto frontier. Therefore, the parameter $\alpha \in [0,1]$ is used to define the weight vector $\boldsymbol{w}^i,\ i=1,\cdots,m$ in (7).

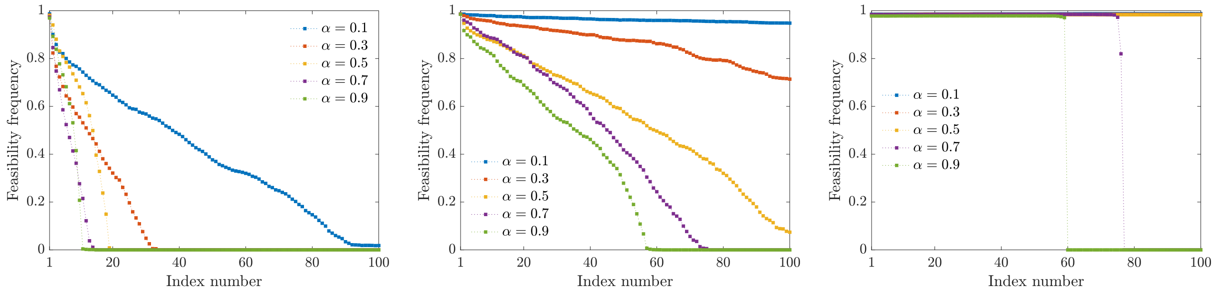
$$\boldsymbol{w}^i = \left[\alpha\frac{i-1}{m-1}, 1-\alpha\frac{i-1}{m-1}\right]^{\mathrm{T}} \tag{7}$$

Note that when $w_1^i = 0$ or $w_2^i = 0$, the objective function value/constraint violation is not ignored and is replaced by the minute value $\delta > 0$. When $\alpha = 1$, the objective function value, and constraint violation are treated equally, and the contribution of the objective function value decreases as α approaches 0. Thus, it is expected to promote search near constraint boundaries.

B. Verification of the Effect of the Parameter α on the Search

To test the effect of the parameter α on the search, we applied MOEA/D to the Test problem (see Appendix) using 5 different α with $\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$ and ran each individual through all generations. Observe the relative frequency "Feasibility frequency". This allows us to observe the probability of each individual becoming the feasible solution. The conditions for MOEA/D are Simulated Binary Crossover [10] (crossover rate $p_c = 1$, distribution constant $\eta_c = 20$) for crossover, Polynomial Mutation [10] (mutation rate $p_m = 1/N$, distribution constant $\eta_m = 20$) for mutation, number of individuals $m = 100$, the maximum number of generations $g_{\max} = 5000$, and the number of neighbors $T = m/10$. The constraint violation is given by (5).

The verification results are shown in Fig. 3. The results confirm the following:



(a) Prob. 1 $(d = 10^{-2})$. (b) Prob. 2 $(d = 10^{-2})$. (c) Prob. 3 $(d = 10^{-2})$.

Fig. 3. "Feasibility frequency" when $\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$ is set in Test problems ($N = 100$).

- The smaller the index number, the higher the "Feasibility frequency", and the larger the index number, the lower the "Feasibility frequency".
- The smaller the α is, the more individuals with high "Feasibility frequency" increase, and the larger the α is the fewer individuals with high "Feasibility frequency" decrease.

Although omitted for reasons of space, similar trends were observed in various problems, and it was confirmed that the "Feasibility frequency" of each individual varies greatly depending on the severity of the constraints and the landscape, even when the same α is used. The reason why the difference between individuals with high and low "Feasibility frequency" is so pronounced in Prob. 3 is that the Pareto frontier in the f-v space becomes concave near the constraint boundary, and the weighted sum cannot properly approximate a concave Pareto frontier.

The above suggests that although the introduction of the parameter α makes it possible to adjust the balance between feasible and infeasible solutions in the population, it is necessary to adjust α appropriately depending on the problem in order to efficiently search for the promising region.

C. Adaptive Adjustment of the Parameter α

To efficiently search for the promising region in the previous section, a mechanism that maintains an appropriate balance between feasible and infeasible solutions in the population is considered effective. On the other hand, since the severity of the constraints and the landscape differ depending on the problem, it is difficult to give an a priori α that achieves an appropriate balance. Therefore, we determine the population bias based on the viability of the individual $\boldsymbol{x}^t$ corresponding to the specified weight vector $\boldsymbol{w}^t$, $t \in \{1, \cdots, m\}$, and adjust α in the direction to eliminate it. In the proposed method, the parameter $\alpha^{(g)}$ of the number of generations g is updated by (8).

$$\alpha^{(g+1)} = \begin{cases} \min\{\gamma_u \alpha^{(g)}, 1\} & \text{if } v(\boldsymbol{x}^{t,(g)}) = 0 \\ \gamma_d \alpha^{(g)} & \text{otherwise} \end{cases} \tag{8}$$

Note that $\gamma_u > 1$ is the increasing factor and $\gamma_d \in (0, 1)$ is the decreasing factor. If $\boldsymbol{x}^t$ is feasible, increase α because the population is biased toward the feasible region, otherwise decrease α because the population is biased toward the infeasible region. This can be regarded as adjusting the Feasibility frequency of $\boldsymbol{x}^t$ to be 0.5.

Algorithm 2 Proposed Method

procedure PROPOSED METHOD(m,T,$g_{\max}$)
  Initialize $\boldsymbol{x}^{i,(1)}$ $(i = 1, \cdots, m)$.
  Initialize $\alpha^{(1)}$.
  Set $\boldsymbol{w}^{i,(1)}$ $(i = 1, \cdots, m)$ by (7).
  $g := 1$
  while $g \leq g_{\max}$ **do**
    for $i = 1$ **to** m **do**
      Calculate the Euclidean distance between $\boldsymbol{w}^{i,(1)}$ and $\boldsymbol{w}^{j,(1)}$ $(j = 1, \cdots, m)$.
      Set the neighborhood index $B(i) = \{i_1, \cdots, i_T\}$.
    end for
    for $i = 1$ **to** m **do**
      Randomly select $a, b \in B(i)$.
      Crossover and mutation on $\boldsymbol{x}^{a,(g)}$ and $\boldsymbol{x}^{b,(g)}$ to produce $\boldsymbol{y}^{i,(g)}$.
      for $j \in B(i)$ **do**
        if $S(\boldsymbol{y}^{i,(g)}|\boldsymbol{w}^{j,(g)}) \leq S(\boldsymbol{x}^{j,(g)}|\boldsymbol{w}^{j,(g)})$ **then**
          $\boldsymbol{x}^{j,(g)} := \boldsymbol{y}^{i,(g)}$
        end if
      end for
    end for
    Update $\alpha^{(g)}$ by (8).
    Update $\boldsymbol{w}^{i,(g)}$ $(i = 1, \cdots, m)$ by (7).
    $g := g + 1$
  end while
  return $\boldsymbol{x}^{i,(g_{\max})}$ $(i = 1, \cdots, m)$
end procedure

The above is expected to maintain an appropriate balance between feasible and infeasible solutions in the population and to efficiently search for the promising region. In the proposed method, the parameters t, γ_u, γ_d are added, and $t = \lfloor 0.8m \rfloor, \gamma_u = 1.001, \gamma_d = 0.999$ are recommended values based on preliminary experiments.

V. NUMERICAL EXPERIMENT

A. Experimental Conditions

Through numerical experiments, we evaluate the search performance of the proposed method. Proposal1 is a model in which the constraint violation is defined by (5) and no normalization is performed on the objective function and the constraint violation, and Proposal2 is a model in which the constraint violation is defined by (6) and normalized to the objective function as well. For the benchmark problem, we use the Test problem (see Appendix), where the number of dimensions $N \in \{50, 100, 300, 500\}$ and the severity of the constraints $d \in \{10^{-2}, 10^{-4}\}$. The initialization region of the

TABLE I

EXPERIMENTAL RESULTS IN THE TEST PROBLEMS, WHERE THE BEST MEAN ARE SHOWN IN BOLD

		Proposal1			Proposal2			IDEA		
Prob.	Dim.	Mean	Std	Feas. runs	Mean	Std	Feas. runs	Mean	Std	Feas. runs
Prob. 1	10	**3.85E-05**	2.56E-05	50	2.42E-04	1.27E-04	50	2.28E-04	9.98E-05	50
($d = 10^{-2}$)	50	**3.02E-04**	1.10E-04	50	1.47E-03	3.62E-04	50	1.98E-02	0	1
	100	**6.58E-04**	1.05E-04	50	3.00E-03	4.27E-04	50	-	-	0
	300	**2.46E-03**	2.22E-04	50	7.33E-03	6.82E-04	50	-	-	0
	500	**4.71E-03**	3.59E-04	50	1.08E-02	6.29E-04	50	-	-	0
Prob. 2	10	**6.38E-05**	4.40E-05	50	2.27E-04	1.11E-04	50	2.47E-04	1.08E-04	50
($d = 10^{-2}$)	50	**5.45E-04**	2.16E-04	50	1.38E-03	3.13E-04	50	-	-	0
	100	**1.62E-03**	4.38E-04	50	2.75E-03	3.68E-04	50	-	-	0
	300	9.11E-03	1.04E-03	50	**7.58E-03**	5.61E-04	50	-	-	0
	500	1.51E-02	1.16E-03	50	**1.11E-02**	7.41E-04	50	-	-	0
Prob. 3	10	2.01E-03	8.93E-04	50	1.50E-03	6.66E-04	50	**2.48E-04**	1.10E-04	50
($d = 10^{-2}$)	50	**6.37E-03**	1.21E-03	50	6.91E-03	1.23E-03	50	-	-	0
	100	1.02E-02	1.40E-03	50	**9.89E-03**	1.20E-03	50	-	-	0
	300	1.93E-02	1.29E-03	50	**1.92E-02**	1.53E-03	50	-	-	0
	500	**2.38E-02**	1.32E-03	50	2.42E-02	1.54E-03	50	-	-	0
Prob. 1	10	**4.30E-05**	2.02E-05	50	4.08E-04	1.80E-04	50	3.58E-04	3.06E-04	50
($d = 10^{-4}$)	50	**2.02E-04**	7.37E-05	50	1.60E-03	3.65E-04	50	-	-	0
	100	**3.76E-04**	7.46E-05	50	2.37E-03	3.42E-04	50	-	-	0
	300	**1.39E-03**	1.54E-04	50	3.81E-03	2.80E-04	50	-	-	0
	500	**3.61E-03**	3.33E-04	36	5.16E-03	3.15E-04	50	-	-	0
Prob. 2	10	**3.96E-05**	2.78E-05	50	3.52E-04	1.69E-04	50	4.64E-04	3.83E-04	50
($d = 10^{-4}$)	50	**2.15E-04**	5.84E-05	50	1.62E-03	4.19E-04	50	-	-	0
	100	**4.22E-04**	9.44E-05	50	2.39E-03	3.16E-04	50	-	-	0
	300	**1.76E-03**	2.13E-04	50	3.80E-03	3.29E-04	50	-	-	0
	500	**3.92E-03**	2.69E-04	50	5.26E-03	2.71E-04	50	-	-	0
Prob. 3	10	2.46E-03	9.94E-04	50	2.51E-03	1.13E-03	50	**4.43E-04**	2.29E-04	50
($d = 10^{-4}$)	50	3.95E-03	6.87E-04	50	**3.89E-03**	6.00E-04	50	-	-	0
	100	4.26E-03	5.14E-04	50	**4.23E-03**	6.10E-04	50	-	-	0
	300	4.73E-03	2.91E-04	50	**4.67E-03**	3.17E-04	50	-	-	0
	500	5.57E-03	3.28E-04	50	**5.49E-03**	3.22E-04	50	-	-	0

population is $[-5, 5]^N$ for the Test problem. The comparison method used is IDEA [7], which utilizes Pareto ranking.

The common conditions of the method are the same as in section IV-B. The parameters of the proposed method are: neighborhood $T = m/10$, $\alpha^{(1)} = 1$, and microvalue $\delta = 10^{-6}$. Using 50 initial values, the mean (Mean) and standard deviation (Std.) of $|f(\boldsymbol{x}^*) - f(\boldsymbol{x}^{\text{best}})|$ and the number of trials that obtained feasible solutions (Feasibility runs) are calculated and compared. Note that $\boldsymbol{x}^{\text{best}}$ is the feasible solution that minimizes the objective function value.

B. Experimental Results

The experimental results for the Test problem are shown in TABLE I. The results of the experiment confirm the following:

- The proposed method may have poorer search performance than IDEA for $N = 50$, but is superior for $N \geq 100$.
- While IDEA rarely obtains feasible solutions as the dimensionality increases or the constraints become tighter, Proposal2 obtains feasible solutions in all results.
- Proposal1 often has better search performance than Proposal2 but in Prob. 1 ($d = 10^{-4}$), Proposal1 may not obtain a feasible solution.

The proposed method shows better convergence to a feasible solution than IDEA for high-dimensional and severely constrained problems. In particular, Proposal2, which normalizes for the objective function and constraint violation, consistently produces good feasible solutions for all problems.

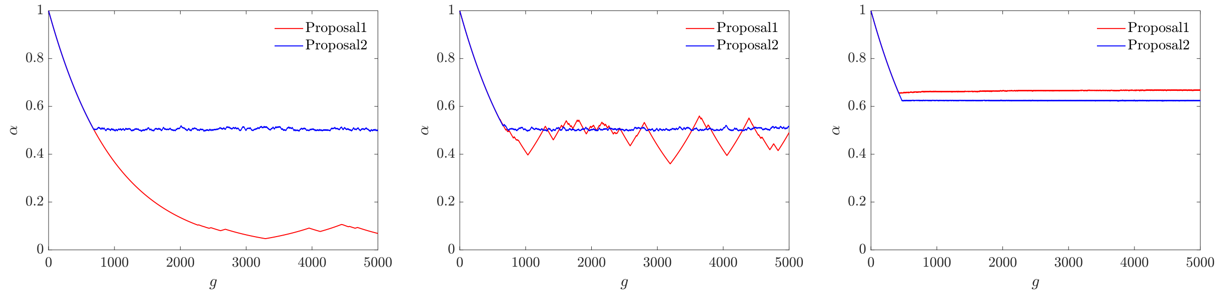
Furthermore, the generation transitions of α in Proposal1 and Proposal2 are shown in Fig. 4, and the "Feasibility frequency" are shown in Fig. 5. The results for generations 4500 to 5000 were added in addition to all generations. The results confirm the following:

- The "Feasibility frequency" in Proposal2 is generally consistent for all generations and for generations 4500 to 5000, and is about 0.5 for $\boldsymbol{x}^t$ $(t = 80)$.
- The "Feasibility frequency" of Proposal1 is similar to that of Proposal2 in Fig. 5(b),(c), but the trend is different in Fig. reffig:result3(a).
- Proposal2 takes relatively fewer generations to adapt α than Proposal1, and α fluctuates less.

As shown above, Proposal2 achieves its goal of adjusting α, but Proposal1 does not adjust α properly in some cases. On the other hand, Proposal2 may have poorer search performance than Proposal1, and since the superiority of Proposal2 cannot be discussed in general, analysis of the effects of normalization is a subject for future work.

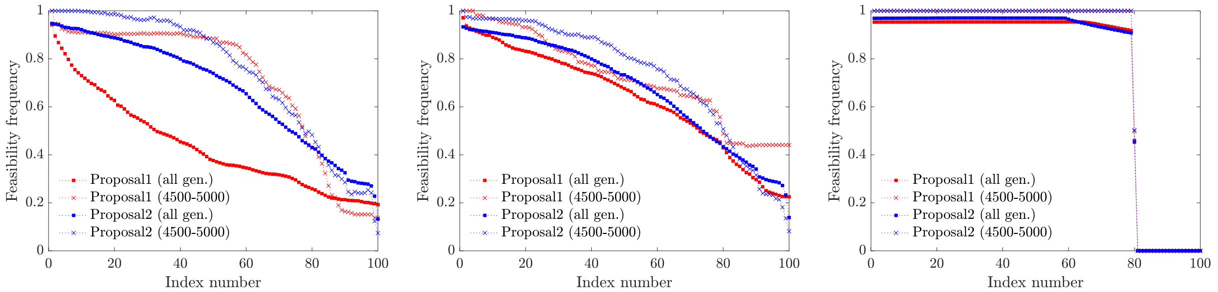
VI. CONCLUSION

In this paper, MOEA/D with adaptive weight adjustment for constraint optimization is proposed to improve the feasibility and convergence of the obtained solution for CHTs with constraint violation as an additional objective function. Proposal2, a model in which the objective function and constraint violation quantities are normalized by the population,



(a) Prob. 1 $(d = 10^{-2})$.   (b) Prob. 2 $(d = 10^{-2})$.   (c) Prob. 3 $(d = 10^{-2})$.

Fig. 4. Transitions of α for Proposal1 and Proposal2 in the Test problems ($N = 100$).



(a) Prob. 1 $(d = 10^{-2})$.   (b) Prob. 2 $(d = 10^{-2})$.   (c) Prob. 3 $(d = 10^{-2})$.

Fig. 5. "Feasibility frequency" of Proposal1 and Proposal2 in the Test problem ($N = 100$).

obtained good feasible solutions for all problems tested in this study. Future works include the following:

- Detailed analysis of the impact of the normalization of the proposed method.
- Validation and improvement of the proposed method when the feasible region is a non-convex set or a large number of constraints.
- Extension of the proposed concept to constrained multi-objective optimization.

REFERENCES

[1] E. Aiyoshi and K. Yasuda, *Metaheuristics and Their Applications*. Tokyo: The Institute of Electrical Engineers of Japan, Ohmsha, 2007 (in Japanese).

[2] C. A. C. Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11-12, pp. 1245–1287, 2002.

[3] H. K. Singh, K. Alam, and T. Ray, "Use of infeasible solutions during constrained evolutionary search: A short survey," *Proc. of Australasian Conf. on Artificial Life and Computational Intelligence*, pp. 193–205, 2016.

[4] J. Joines and C. Houck, "On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with gas," *Proc. of the First IEEE Conf. on Evolutionary Computation*, pp. 579–584, 1994.

[5] A. Lemonge and H. Barbosa, "An adaptive penalty scheme for genetic algorithms in structural optimization," *Int. Journal for Numerical Methods in Engineering*, vol. 59, no. 5, pp. 703–736, 2004.

[6] S. Venkatraman and G. G. Yen, "A generic framework for constrained optimization using genetic algorithms," *IEEE Trans. on Evolutionary Computation*, vol. 9, no. 4, pp. 424–435, 2005.

[7] 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, 2009.

[8] B. Wang, H. Li, Q. Zhang, and Y. Wang, "Decomposition-based multiobjective optimization for constrained evolutionary optimization," *IEEE Trans. on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 574–587, 2021.

[9] Y. Yasuda, W. Kumagai, K. Tamura, and K. Yasuda, "An extension of MOEA/D to constrained optimization and adaptive weight adjustment," *IEEJ Trans. on Electronics, Information and Systems*, vol. 142, no. 1, pp. 108–109, 2022 (in Japanese).

[10] 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.

[11] 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.

APPENDIX

The definition of the Test problem used in this paper is given in TABLE II. $\mathbf{1} = [1, \cdots, 1]^{\mathrm{T}} \in \mathbb{R}^N$, $\mathrm{sgn} : \mathbb{R} \to \{-1, 1\}$ is the sign function, $d \in (0, 1)$ is a parameter that sets the constraint severity. The objective function, global optimal solution, and feasible region are common as they are in a hypersphere of radius $\sqrt{d}$, but the constraint functions are of different scales and landscapes. $g^1(\boldsymbol{x})$ (Prob. 1) is a quadratic function, $g^2(\boldsymbol{x})$ (Prob. 2) an exponential function, and $g^3(\boldsymbol{x})$ (Prob. 3) an irrational function with power roots.

TABLE II
TEST PROBLEMS

Problem	Definition		Optimal solution
Prob. 1	$\min_{\boldsymbol{x}\in\mathbb{R}^N}$	$f^1(\boldsymbol{x}) = f(\boldsymbol{x})$	$[1-\sqrt{d}, \cdots, 1-\sqrt{d}]^{\mathrm{T}}$
	subj. to	$g^1(\boldsymbol{x}) = g(\boldsymbol{x}) \leq 0$	
Prob. 2	$\min_{\boldsymbol{x}\in\mathbb{R}^N}$	$f^2(\boldsymbol{x}) = f(\boldsymbol{x})$	$[1-\sqrt{d}, \cdots, 1-\sqrt{d}]^{\mathrm{T}}$
	subj. to	$g^2(\boldsymbol{x}) = \exp(10 g(\boldsymbol{x})) - 1 \leq 0$	
Prob. 3	$\min_{\boldsymbol{x}\in\mathbb{R}^N}$	$f^3(\boldsymbol{x}) = f(\boldsymbol{x})$	$[1-\sqrt{d}, \cdots, 1-\sqrt{d}]^{\mathrm{T}}$
	subj. to	$g^3(\boldsymbol{x}) = \mathrm{sgn}(g(\boldsymbol{x}))\lvert g(\boldsymbol{x})\rvert^{1/4} \leq 0$	
	where	$f(\boldsymbol{x}) = \lVert\boldsymbol{x}\rVert^2/N,\ \ g(\boldsymbol{x}) = \lVert\boldsymbol{x}-\mathbf{1}\rVert^2/N - d,\ \ d \in (0,1)$	