# Constrained Optimization by Applying the $\alpha$ Constrained Method to the Nonlinear Simplex Method With Mutations

Tetsuyuki Takahama, *Member, IEEE,* and Setsuko Sakai, *Member, IEEE*

***Abstract*—Constrained optimization problems are very important and frequently appear in the real world. The $\alpha$ constrained method is a new transformation method for constrained optimization. In this method, a satisfaction level for the constraints is introduced, which indicates how well a search point satisfies the constraints. The $\alpha$ level comparison, which compares search points based on their level of satisfaction of the constraints, is also introduced. The $\alpha$ constrained method can convert an algorithm for unconstrained problems into an algorithm for constrained problems by replacing ordinary comparisons with the $\alpha$ level comparisons. In this paper, we introduce some improvements including mutations to the nonlinear simplex method to search around the boundary of the feasible region and to control the convergence speed of the method, we apply the $\alpha$ constrained method and we propose the improved $\alpha$ constrained simplex method for constrained optimization problems. The effectiveness of the $\alpha$ constrained simplex method is shown by comparing its performance with that of the stochastic ranking method on various constrained problems.**

***Index Terms*—$\alpha$ constrained method, constrained optimization, evolutionary algorithms, nonlinear optimization, nonlinear simplex method.**

## I. Introduction

CONSTRAINED optimization problems, where the objective functions are optimized under given constraints are very important and frequently appear in the real world. The general constrained optimization problem (P) with inequality, equality, upper bound, and lower bound constraints is defined as follows:

$$\begin{array}{llll} \text{(P) minimize} & f(\boldsymbol{x}) & & \\ \text{subject to} & g_j(\boldsymbol{x}) \le 0, & j = 1, \ldots, q & \\ & h_j(\boldsymbol{x}) = 0, & j = q+1, \ldots, m & \\ & l_i \le x_i \le u_i, & i = 1, \ldots, n & (1) \end{array}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ is an $n$ dimensional vector of decision variables, $f(\boldsymbol{x})$ is an objective function, $g_j(\boldsymbol{x}) \le 0$ are $q$ inequality constraints, and $h_j(\boldsymbol{x}) = 0$ are $m - q$ equality constraints. The functions $f$, $g_j$, and $h_j$ are linear or nonlinear real-valued functions. The values $u_i$ and $l_i$ are the upper bound and the lower bound of $x_i$, respectively. The upper and lower bounds define the *search space* $\mathcal{S}$. The inequality and equality constraints define the *feasible region* $\mathcal{F}$. The feasible solutions exist in $\mathcal{F} \subseteq \mathcal{S}$.

There exist many studies on solving constrained optimization problems using evolutionary algorithms [1]–[4]. These studies can be classified into several categories according to the way the constraints are treated as follows.

1) Constraints are only used to see whether a search point is feasible or not. In this category, the search process begins with a feasible point or multiple feasible points and continues to search for new points within the feasible region. When a new search point is generated and the point is not feasible, the point is repaired or discarded. Approaches in this category are called death penalty. Michalewicz and Nazhiyath proposed a method in which an infeasible search point is repaired and changed to a feasible point by referring to one of the feasible points [5]. El-Gallad *et al.* proposed a method in which an infeasible search point is discarded and replaced by the best visited point [6]. In this category, generating initial feasible points is difficult and computationally demanding when the feasible region is very small. Especially, if the problem has equality constraints, it is almost impossible to find initial feasible points.
2) The constraint violation, which is the sum of the violation of all constraint functions, is combined with the objective function. The penalty function method is in this category. In the penalty function method, an extended objective function is defined by adding the constraint violation to the objective function as a penalty. The optimization of the objective function and the constraint violation is realized by the optimization of the extended objective function. The main difficulty of the penalty function method is the difficulty of selecting an appropriate value for the penalty coefficient that adjusts the strength of the penalty. If the penalty coefficient is large, feasible solutions can be obtained. However, in this case, the optimization of the objective function will be insufficient and the quality of the solution will not be high, because search around the boundary of the feasible region tends to be avoided. On the contrary, if the penalty coefficient is

Manuscript received July 23, 2004; revised February 22, 2005. This paper was supported in part by the Japan Society of the Promotion of Science under Grant-in-Aid for Scientific Research (c) 17510139 and Grant-in-Aid for Scientific Research 16500083.

T. Takahama is with the Department of Intelligent Systems, Hiroshima City University, Asaminami-ku, Hiroshima 731-3194, Japan (e-mail: takahama@its.hiroshima-cu.ac.jp).

S. Sakai is with the Faculty of Commercial Sciences, Hiroshima Shudo University, Asaminami-ku, Hiroshima 731-3195, Japan (e-mail: setuko@shudo-u.ac.jp).

Digital Object Identifier 10.1109/TEVC.2005.850256

small, high-quality (but infeasible) solutions can be obtained as it is difficult to decrease the constraint violation. Several methods that control the penalty coefficient dynamically have been proposed [7]–[9], however, ideal control of the coefficient is problem dependent [10] and it is difficult to determine a general control scheme. Farmani and Wright proposed self-adaptive fitness formulation [11]. The approach does not require parameter tuning because the value of penalty for each search point is decided based on three bounding points: the best point, the worst of the infeasible points, and the point with the worst objective function value in the current search points. It is shown that the method can find good solutions. But the solutions for some problems are not sufficient when they are compared to solutions found by Runarsson and Yao [12].

3) The constraint violation and the objective function are used separately. In this category, both the constraint violation and the objective function are optimized by a lexicographic order in which the constraint violation precedes the objective function. Takahama and Sakai [13]–[15] proposed the $\alpha$ constrained method, which adopts a lexicographic ordering with relaxation of the constraints. This method can optimize problems with equality constraints effectively through the relaxation of the constraints. Deb [16] proposed a method in which the extended objective function that realizes the lexicographic ordering is used. Runarsson and Yao [12] proposed the stochastic ranking method in which the stochastic lexicographic order, which ignores the constraint violation with some probability, is used. These methods were successfully applied to various problems.
4) The constraints and the objective function are optimized by multiobjective optimization methods. In this category, the constrained optimization problems are solved as the multiobjective optimization problems in which the objective function and the constraint functions are objectives to be optimized [17]–[21]. In many cases, solving multiobjective optimization problems is a more difficult and expensive task than solving single objective optimization problems.

In this paper, we investigate the $\alpha$ constrained method in the promising category 3). In the $\alpha$ constrained method, a *satisfaction level* for the constraints is introduced to indicate how well a search point satisfies the constraints and the $\alpha$ *level comparison* is defined as an order relation that gives priority to the satisfaction level over the value of the objective function. The $\alpha$ constrained method is a new transformation method that converts an algorithm for unconstrained optimization into an algorithm for constrained optimization by replacing the ordinal comparisons with the $\alpha$ level comparisons in direct search methods such as Powell's method, the nonlinear simplex method and genetic algorithms. In [13]–[15], we proposed the $\alpha$ constrained Powell's method which is a combination of the $\alpha$ constrained method and Powell's direct search method [22]. In [23] and [24], we proposed the $\alpha$ constrained simplex method which is a combination of the $\alpha$ constrained method and the nonlinear simplex method by Nelder and Mead [25]. In [26] and [27], we proposed the $\alpha$ constrained genetic algorithm which is a combination of the $\alpha$ constrained method and the genetic algorithm using linear ranking selection. We showed that constrained optimization problems were effectively solved by these methods.

In the nonlinear simplex method, a simplex is spanned by multiple search points and the simplex shows the region in which an optimal solution will exist. The simplex is gradually reduced while better points are found primarily inside the simplex. When the simplex has sufficiently converged, the optimal solution with a high precision will be obtained. However, when the simplex method is applied to constrained optimization problems, points around the boundary of the feasible region are sometimes skipped when the simplex is reduced. In this paper, to avoid such skipping, we modify the nonlinear simplex method by adding mutations for searching the boundary and by using multiple simplexes instead of a single simplex. We propose the improved $\alpha$ constrained simplex method, which solves constrained optimization problems, by applying the $\alpha$ constrained method to the modified nonlinear simplex method. The simplex method can be seen as an evolutionary algorithm with high convergence speed, in which a particular selection, special variation operators, and replacement strategy are used. There exist several studies on hybridizing the simplex method with a genetic algorithm to improve the convergence speed of the genetic algorithm [28]–[30]. But they are for solving unconstrained problems. We show that the $\alpha$ constrained simplex method is a fast and stable algorithm for constrained optimization problems. We also show the effectiveness of the $\alpha$ simplex method by comparing its performance with that of the stochastic ranking method on various constrained problems.

The rest of this paper is organized as follows. Section II describes the $\alpha$ constrained method briefly. Section III describes the improved $\alpha$ constrained simplex method by introducing mutations and multiple simplexes. Section IV presents experimental results on various benchmark problems discussed in [12]. Comparisons with the results in [12] are included in this section. Section V describes discussion on parameter settings. Finally, Section VI concludes with a brief summary of this paper and a few remarks.

## II. The $\alpha$ Constrained Method

In this section, we survey briefly the $\alpha$ constrained method [13], [14].

### A. Satisfaction Level of Constraints

We introduce the *satisfaction level* of constraints $\mu(\boldsymbol{x})$ to indicate how well a search point $\boldsymbol{x}$ satisfies the constraints. The satisfaction level $\mu(\boldsymbol{x})$ is the following function:

$$\begin{cases} \mu(\boldsymbol{x}) = 1, & \text{if } g_i(\boldsymbol{x}) \leq 0,\ h_j(\boldsymbol{x}) = 0 \\ & \quad \text{for all } i, j \\ 0 \leq \mu(\boldsymbol{x}) < 1, & \text{otherwise} \end{cases}. \tag{2}$$

In order to define the satisfaction level $\mu(\boldsymbol{x})$ of (P), the satisfaction level of each constraint in (P) is defined and all individual satisfaction levels are combined. For example, each constraint in (P) can be transformed into one of the following satisfaction levels that are defined by piecewise linear functions on $g_i$ and $h_j$

$$\mu_{g_i}(\boldsymbol{x}) = \begin{cases} 1, & \text{if } g_i(\boldsymbol{x}) \le 0 \\ 1 - \frac{g_i(\boldsymbol{x})}{b_i}, & \text{if } 0 \le g_i(\boldsymbol{x}) \le b_i \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

$$\mu_{h_j}(\boldsymbol{x}) = \begin{cases} 1 - \frac{|h_j(\boldsymbol{x})|}{b_j}, & \text{if } |h_j(\boldsymbol{x})| \le b_j \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where $b_i$ and $b_j$ are proper positive fixed numbers. In order to obtain the satisfaction level $\mu(\boldsymbol{x})$ of (P), the satisfaction levels $\mu_{g_i}(\boldsymbol{x})$ and $\mu_{h_j}(\boldsymbol{x})$ need to be combined. In this paper, minimization is used for the combination operator

$$\mu(\boldsymbol{x}) = \min_{i,j} \left\{ \mu_{g_i}(\boldsymbol{x}), \mu_{h_j}(\boldsymbol{x}) \right\}. \quad (5)$$

Also, it is possible to define a penalty $\phi(\cdot)$ and define the satisfaction level from the penalty

$$\phi(\boldsymbol{x}) = \sum_i \|\max\{0, g_i(\boldsymbol{x})\}\|^p + \sum_j \|h_j(\boldsymbol{x})\|^p \quad (6)$$

$$\mu(\boldsymbol{x}) = \max\left\{0, 1 - \frac{\phi(\boldsymbol{x})}{B}\right\} \quad (7)$$

where $p$ and $B$ are proper positive fixed numbers.

### B. The $\alpha$ Level Comparison

The *$\alpha$ level comparison* is defined as an order relation on the set of $(f(\boldsymbol{x}), \mu(\boldsymbol{x}))$. If the satisfaction level of a point is less than 1, the point is not feasible and its worth is low. The $\alpha$ level comparisons are defined by a lexicographic order in which $\mu(\boldsymbol{x})$ precedes $f(\boldsymbol{x})$, because the feasibility of $\boldsymbol{x}$ is more important than the minimization of $f(\boldsymbol{x})$.

Let $f_1(f_2)$ and $\mu_1(\mu_2)$ be the function value and the satisfaction level respectively, at a point $\boldsymbol{x}_1(\boldsymbol{x}_2)$. Then, for any $\alpha$ satisfying $0 \le \alpha \le 1$, the $\alpha$ level comparisons $<_\alpha$ and $\le_\alpha$ between $(f_1, \mu_1)$ and $(f_2, \mu_2)$ are defined as follows:

$$(f_1, \mu_1) <_\alpha (f_2, \mu_2) \Leftrightarrow \begin{cases} f_1 < f_2, & \text{if } \mu_1, \mu_2 \ge \alpha \\ f_1 < f_2, & \text{if } \mu_1 = \mu_2 \\ \mu_1 > \mu_2, & \text{otherwise} \end{cases} \quad (8)$$

$$(f_1, \mu_1) \le_\alpha (f_2, \mu_2) \Leftrightarrow \begin{cases} f_1 \le f_2, & \text{if } \mu_1, \mu_2 \ge \alpha \\ f_1 \le f_2, & \text{if } \mu_1 = \mu_2 \\ \mu_1 > \mu_2, & \text{otherwise} \end{cases}. \quad (9)$$

In the case of $\alpha = 0$, the $\alpha$ level comparisons $<_0$ and $\le_0$ are equivalent to the ordinal comparisons $<$ and $\le$ between function values. Also, in the case of $\alpha = 1$, $<_1$ and $\le_1$ are equivalent to the lexicographic order in which the satisfaction level $\mu(\boldsymbol{x})$ precedes the function value $f(\boldsymbol{x})$.

### C. Properties of the $\alpha$ Constrained Method

An optimization problem solved by the $\alpha$ constrained method, that is, a problem in which ordinary comparisons are replaced with the $\alpha$ level comparisons, $(\mathrm{P}_{\le_\alpha})$, is defined as follows:

$$(\mathrm{P}_{\le_\alpha}) \quad \text{minimize}_{\le_\alpha} \quad f(\boldsymbol{x}) \quad (10)$$

where $\text{minimize}_{\le_\alpha}$ denotes the minimization based on the $\alpha$ level comparison $\le_\alpha$. Also, a problem $(\mathrm{P}^\alpha)$ is defined such that the constraint of (P), that is, $\mu(\boldsymbol{x}) = 1$, is relaxed and replaced with $\mu(\boldsymbol{x}) \ge \alpha$

$$(\mathrm{P}^\alpha) \quad \begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) \\ \text{subject to} & \mu(\boldsymbol{x}) \ge \alpha. \end{array} \quad (11)$$

The problem $(\mathrm{P}_{\le_\alpha})$ means searching the minimum point ordered by the order relation $\alpha$ level comparison $\le_\alpha$. The problem $(\mathrm{P}^\alpha)$ means searching the minimum point ordered by the ordinary comparison $\le$ under the relaxed condition $\mu(\boldsymbol{x}) \ge \alpha$.

For the three types of problems, $(\mathrm{P}^\alpha)$, $(\mathrm{P}_{\le_\alpha})$, and (P), the following theorems are given [13], [14].

*Theorem 1:* If an optimal solution of $(\mathrm{P}^1)$ exists, any optimal solution of $(\mathrm{P}_{\le_\alpha})$ is an optimal solution of $(\mathrm{P}^\alpha)$.

*Theorem 2:* If an optimal solution of (P) exists, any optimal solution of $(\mathrm{P}_{\le_1})$ is an optimal solution of (P).

*Theorem 3:* Let $\{\alpha_n\}$ be a strictly increasing nonnegative sequence converging to 1. Let $f(\boldsymbol{x})$ and $\mu(\boldsymbol{x})$ be continuous functions of $\boldsymbol{x}$. Assume that an optimal solution $\boldsymbol{x}^*$ of $(\mathrm{P}^1)$ exists and an optimal solution $\hat{\boldsymbol{x}}_n$ of $(\mathrm{P}_{\le_{\alpha_n}})$ exists for any $\alpha_n$. Then, any accumulation point of the sequence $\{\hat{\boldsymbol{x}}_n\}$ is an optimal solution of $(\mathrm{P}^1)$.

The proof of theorem is given in the Appendix.

Theorems 1 and 2 show that a constrained optimization problem can be transformed into an equivalent unconstrained optimization problem by using the $\alpha$ level comparisons. So, if the $\alpha$ level comparisons are incorporated into an existing unconstrained optimization method, constrained optimization problems can be solved. It is thought that the $\alpha$ constrained method converts an algorithm for unconstrained optimization into an algorithm for constrained optimization by replacing the ordinary comparisons with the $\alpha$ level comparisons. Theorem 3 shows that, in the $\alpha$ constrained method, an optimal solution of $(\mathrm{P}^1)$ can be obtained by converging $\alpha$ to 1, in a similar fashion to increasing the penalty coefficient to infinity in the penalty function method.

## III. The $\alpha$ Constrained Simplex Method

In this section, we first describe the nonlinear simplex method proposed by Nelder and Mead [25]. Then, we describe the improved $\alpha$ constrained simplex method, which is the integration of the $\alpha$ constrained method and the nonlinear simplex method with some modifications including mutations.

### A. Nonlinear Simplex Method

A set $S \subset \boldsymbol{R}^n$ is called a *simplex* if it is the convex hull spanned by $n+1$ points which are affine independent. Let the set of all vertices of $S$ be denoted by $U$ and each vertex be denoted by $\boldsymbol{x}^k$, $k = 1, \cdots, n+1$ where $n$ is the dimension of $\boldsymbol{x}^k$.

In the search process, a new search point, which has a desirable value with respect to the objective function, is determined based on the points in $U$. This new search point then replaces the least desirable point in $U$. This process is iterated until the simplex $S$ has sufficiently converged. For the sake of this process, the following three points are determined:

$$\boldsymbol{x}^l = \arg\min_k f(\boldsymbol{x}^k) \quad (12)$$

$$\boldsymbol{x}^h = \arg\max_k f(\boldsymbol{x}^k) \quad (13)$$

$$\boldsymbol{x}^s = \arg\max_{k \neq h} f(\boldsymbol{x}^k) \quad (14)$$

where $\boldsymbol{x}^l$, $\boldsymbol{x}^h$, and $\boldsymbol{x}^s$ are the points in $U$ which currently have the best, the worst, and the second worst function values, respectively. Thus, $\boldsymbol{x}^h$ is the point to be replaced in each iteration. To determine a new search point, the centroid $\boldsymbol{x}^0$ of all other points except $\boldsymbol{x}^h$ is calculated

$$\boldsymbol{x}^0 = \frac{1}{n}\sum_{k \neq h} \boldsymbol{x}^k. \quad (15)$$

Using $\boldsymbol{x}^l$, $\boldsymbol{x}^h$, $\boldsymbol{x}^s$, and $\boldsymbol{x}^0$ defined in (12)–(15), the following operations are defined.

- *Reflection* generates the reflection point $\boldsymbol{x}^r$ of $\boldsymbol{x}^h$ about $\boldsymbol{x}^0$

$$\boldsymbol{x}^r = (1+a)\boldsymbol{x}^0 - a\boldsymbol{x}^h \quad (a > 0). \quad (16)$$

- *Contraction* generates the contraction point $\boldsymbol{x}^c$ dividing between $\boldsymbol{x}^h$ and $\boldsymbol{x}^0$

$$\boldsymbol{x}^c = b\boldsymbol{x}^h + (1-b)\boldsymbol{x}^0 \quad (0 < b < 1). \quad (17)$$

- *Expansion* generates the expansion point $\boldsymbol{x}^e$ in the direction from $\boldsymbol{x}^0$ to $\boldsymbol{x}^r$

$$\boldsymbol{x}^e = c\boldsymbol{x}^r + (1-c)\boldsymbol{x}^0 \quad (c > 1). \quad (18)$$

- *Reduction* replaces all points $\boldsymbol{x}^k \in U$ by the middle of $\boldsymbol{x}^k$ and $\boldsymbol{x}^l$

$$\boldsymbol{x}^k = \frac{1}{2}(\boldsymbol{x}^k + \boldsymbol{x}^l). \quad (19)$$

where $a$, $b$, and $c$ are the algorithm parameters with recommended values of $a = 1$, $b = 0.5$, and $c = 2$.

The nonlinear simplex method is described as follows.

Step 0) An initial simplex is created by generating $n+1$ vertices $U = \{\boldsymbol{x}^k\}$.

Step 1) The points $\boldsymbol{x}^l$, $\boldsymbol{x}^h$, $\boldsymbol{x}^s$, and $\boldsymbol{x}^0$ are determined.

Step 2) The reflection point $\boldsymbol{x}^r$ is calculated. If $\boldsymbol{x}^r$ is better than the best point $\boldsymbol{x}^l$ $(f(\boldsymbol{x}^r) < f(\boldsymbol{x}^l))$, then go to Step 3). Otherwise, go to Step 4).

Step 3) The expansion point $\boldsymbol{x}^e$ is calculated. If $\boldsymbol{x}^e$ is better than $\boldsymbol{x}^l$, $\boldsymbol{x}^h$ is replaced by $\boldsymbol{x}^e$. Otherwise, replace $\boldsymbol{x}^h$ by $\boldsymbol{x}^r$ and return to Step 1).

Step 4) If $\boldsymbol{x}^r$ is better than or equal to $\boldsymbol{x}^s$, replace $\boldsymbol{x}^h$ by $\boldsymbol{x}^r$ and return to Step 1). Otherwise, go to Step 5).

Step 5) If $\boldsymbol{x}^r$ is better than $\boldsymbol{x}^h$, $\boldsymbol{x}^h$ is replaced by $\boldsymbol{x}^r$.

Step 6) The contraction point $\boldsymbol{x}^c$ is calculated. If $\boldsymbol{x}^c$ is better than $\boldsymbol{x}^h$, $\boldsymbol{x}^h$ is replaced by $\boldsymbol{x}^c$. Otherwise, all points except for $\boldsymbol{x}^l$ are replaced by their reduction points. Go back to Step 1).

Usually, the simplex in which the distances between the vertices are the same is used as an initial simplex. The $n+1$ points for an initial simplex in which the distances are $d$ can be defined as follows:

$$s_1 = \frac{1}{\sqrt{2}n}(\sqrt{n+1} + n - 1)d \quad (20)$$

$$s_2 = \frac{1}{\sqrt{2}n}(\sqrt{n+1} - 1)d \quad (21)$$

$$[x^0, x^1, x^2, \cdots, x^n] = \begin{bmatrix} 0 & s_1 & s_2 & s_2 & \cdots & s_2 \\ 0 & s_2 & s_1 & s_2 & \cdots & \vdots \\ \vdots & s_2 & s_2 & s_1 & \cdots & \vdots \\ \vdots & \vdots & \vdots & s_2 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & & s_2 \\ 0 & s_2 & s_2 & \cdots & s_2 & s_1 \end{bmatrix}. \quad (22)$$

The iterations will stop when the simplex $S$ is converged enough, such as when the standard deviation of the function values of all vertices $\delta$ becomes sufficiently small

$$\delta = \sqrt{\frac{1}{n+1}\sum_i \left(f(\boldsymbol{x}^k) - \bar{f}\right)^2} \quad (23)$$

$$\bar{f} = \frac{1}{n+1}\sum_i f(\boldsymbol{x}^k). \quad (24)$$

The simplex method can be seen as an evolutionary algorithm using population of $n+1$ individuals, a particular selection of parents, special variation operators (such as reflection, contraction and expansion) and replacement strategy.

### *B. Modifications to the Nonlinear Simplex Method*

To solve constrained optimization problems, several modifications are introduced to the nonlinear simplex method.

The first and most important modification is the application of the $\alpha$ constrained method. The ordinary comparisons in the nonlinear simplex method are replaced with the $\alpha$ level comparisons. By this replacement, the $\alpha$ constrained simplex method is defined and the method can solve constrained problems.

The second modification is the introduction of mutations. In this study, we adopt mutations which are similar to the boundary mutation proposed by Michalewicz *et al.* [5]. In the boundary mutation, an individual is changed into a point on the boundary of the feasible region. A gene $x_i$ of an individual $\boldsymbol{x}$ is randomly selected, and is mutated and changed to a new gene $x_i'$, which is selected from either $l(x_i)$ or $r(x_i)$ with equal probability, where $l(x_i)$ and $r(x_i)$ are determined as follows:

$$l(x_i) = \min_{(\ldots, x_{i-1}, x_i', x_{i+1}, \ldots) \in \mathcal{F}} x_i' \quad (25)$$

$$r(x_i) = \max_{(\ldots, x_{i-1}, x_i', x_{i+1}, \ldots) \in \mathcal{F}} x_i'. \quad (26)$$

Here, $l(x_i)$ and $r(x_i)$ are the lower bound and the upper bound, respectively, of the gene $x_i'$ such that the new individual $\boldsymbol{x}' = (\cdots, x_{i-1}, x_i', x_{i+1}, \cdots)$ exists in the feasible region.

In the $\alpha$ constrained simplex method, some infeasible points might be included as search points and $l(x_i)$ or $r(x_i)$ might not be defined for these infeasible points. In this case, we adopt another mutation, in which the satisfaction level $\mu(\boldsymbol{x})$ is maximized to approach the feasible region [26], [27]. That is, if a point is feasible, it is mutated by the boundary mutation and if not, the satisfaction level of the point is maximized by the other mutation. In the $\alpha$ constrained simplex method, the worst point is changed by the mutations. A variable $x_i^h$ of the worst point $\boldsymbol{x}^h$ is randomly selected and mutated, producing a new point $\boldsymbol{x}' = (\cdots, x_{i-1}^h, x_i', x_{i+1}^h, \cdots)$. In other words, the mutations replace $x_i^h$ by $x_i'$ with a probability equal to the mutation rate $P_m$ as follows:

$$x_i' = \begin{cases} l(x_i) \text{ or } r(x_i), & \text{if } \boldsymbol{x} \in \mathcal{F} \\ \arg \max_{l_i \le x_i' \le u_i} \mu(\ldots, x_i', \ldots), & \text{if } \boldsymbol{x} \notin \mathcal{F} \end{cases} \quad (27)$$

where $l_i$ and $u_i$ are the upper and lower bound of the $i$th variable $x_i$, respectively. The decision of $l(x_i)$ and $r(x_i)$ and the maximization of $\mu(\ldots, x_i, \ldots)$ is realized by the combination of bracketing and binary search. These mutations are effective in order to search for optimal solutions in constrained optimization problems, because optimal solutions of such problems very often exist near the boundary of the feasible region.

The third modification is the introduction of multiple simplexes. It is known that the simplex $S$ sometimes loses affine independence and the nonlinear simplex method cannot find optimal solutions well. To avoid this situation, multiple simplexes are used in the $\alpha$ constrained simplex method. In the initialization step, the $N$ $(> n+1)$ points are generated as initial search points. When a centroid $\boldsymbol{x}^0$ is determined, a simplex is composed by selecting $n+1$ points from all points except for the worst point $\boldsymbol{x}^h$ and the centroid of the simplex is used as the centroid $\boldsymbol{x}^0$. The centroids are defined by the different simplexes. So, even if some simplexes lose affine independence, the other affine independent simplexes can lead the search process. In this operation, at least $n+2$ search points is used, because $n+1$ points are used for a simplex and one point is used for the worst point.

The nonlinear simplex method realizes very fast convergence, but it also sometimes skips important regions and falls into a local minimum. To control the convergence speed, the following modifications are introduced: The reduction operation is replaced with the contraction operation between the worst point and the best point to avoid rapid convergence of the search points. A higher value is selected as the coefficient $b$ such as $b = 0.75$ to control the convergence speed. The introduction of the mutations and the multiple simplexes also help to control the convergence speed.

### C. Algorithm of the $\alpha$ Constrained Simplex Method

The algorithm of the $\alpha$ constrained simplex method is described as follows.

Step 0) An initial $N$ search points are generated, where $U = \{\boldsymbol{x}^k, k = 1, 2, \cdots, N\}$.

Step 1) The points $\boldsymbol{x}^l$, $\boldsymbol{x}^h$, and $\boldsymbol{x}^s$ are determined.

Step 2) With a probability equal to the mutation rate $P_m$, the mutations are applied to $\boldsymbol{x}^h$ and Step 1) is returned to. Otherwise, with a probability of $1-P_m$, a simplex is composed by selecting $n+1$ points from $U - \{\boldsymbol{x}^h\}$ and $\boldsymbol{x}^0$ is determined as the centroid of the simplex. In this latter case, go to Step 3). In this paper, the $n+1$ points are selected in a simple way where the points are taken sequentially based on the vertex number $k$ with skipping the worst vertex $\boldsymbol{x}^h$.

Step 3) The reflection point $\boldsymbol{x}^r$ is calculated. If $\boldsymbol{x}^r$ is better than the best point $\boldsymbol{x}^l$, that is, $(f(\boldsymbol{x}^r), \mu(\boldsymbol{x}^r)) <_\alpha (f(\boldsymbol{x}^l), \mu(\boldsymbol{x}^l))$, then go to Step 4). Otherwise, go to Step 5).

Step 4) The expansion point $\boldsymbol{x}^e$ is calculated. If $\boldsymbol{x}^e$ is better than $\boldsymbol{x}^l$, $\boldsymbol{x}^h$ is replaced by $\boldsymbol{x}^e$. Otherwise, $\boldsymbol{x}^h$ is replaced by $\boldsymbol{x}^r$. Go back to Step 1).

Step 5) If $\boldsymbol{x}^r$ is better than or equal to $\boldsymbol{x}^s$, replace $\boldsymbol{x}^h$ by $\boldsymbol{x}^r$ and return to Step 1). Otherwise, go to Step 6).

Step 6) If $\boldsymbol{x}^r$ is better than $\boldsymbol{x}^h$, $\boldsymbol{x}^h$ is replaced by $\boldsymbol{x}^r$.

Step 7) The contraction point $\boldsymbol{x}^c$ is calculated. If $\boldsymbol{x}^c$ is better than $\boldsymbol{x}^h$, $\boldsymbol{x}^h$ is replaced by $\boldsymbol{x}^c$. Otherwise, $\boldsymbol{x}^h$ is replaced by the contraction point between $\boldsymbol{x}^h$ and $\boldsymbol{x}^l$. Go back to Step 1).

The initial search points are randomly selected in the boundary of the search space $\mathcal{S}$ as follows: The $i$th dimension is randomly selected and either the lower bound $l_i$ or the upper bound $u_i$ is assigned to $x_i$. The other variables are generated randomly between the lower bound and the upper bound of each variable. The iterations are executed until the maximum iteration $T_{max}$.

Usually, the $\alpha$ level does not need to be controlled. Many constrained problems can be solved based on the lexicographic order where the $\alpha$ level is constantly 1. However, for some problems in which the feasible region is very small, such as problems with equality constraints, the $\alpha$ level should be controlled properly to obtain high quality solutions. For example, a simple control strategy $\alpha(t) = (1-\beta)\alpha(t-1)+\beta$ $(0 < \beta < 1)$ increases the $\alpha$ level from the initial value $\alpha(0)$ to 1. The algorithm in which the $\alpha$ level is controlled is shown in Fig. 1. Sample source files can be downloaded from http://www.chi.its.hiroshima-cu.ac.jp/~takahama/eng/research.html.

## IV. Constrained Nonlinear Programming Problems

In this paper, the 13 benchmark problems mentioned in Runarsson and Yao [12] are optimized, and the results by the $\alpha$ constrained simplex method are compared with their results.

### A. Test Problems and the Experimental Conditions

In the 13 benchmark problems, `g02`, `g03`, `g08`, and `g12` are maximization problems, and the others are minimization problems. Problems `g03`, `g05`, `g11`, and `g13` contain equality constraints. For the problems, the equality constraints are relaxed, that is, all equality constraints $h_j(\boldsymbol{x}) = 0$ are replaced by inequalities: $|h_j(\boldsymbol{x})| \le \delta$, $\delta > 0$, where $\delta = 10^{-4}$. Problem `g12` is the harder version studied in [31], where the feasible region consists of $9^3$ disjointed spheres, each with a radius of 0.25.

```
simplex_alpha()
{
   α=α(0);
   U=generate N initial search points;
   for(t=1; t≤T_max; t++) {
      x^l=best vertex in U;              ········· eq.(12)
      x^h=worst vertex in U;             ········· eq.(13)
      x^s=second worst vertex in U;      ······ eq.(14)
      sample r ∈ u(0,1);
      if(r<P_m)
         mutation(x^h);
      else
         operation(U, x^l, x^h, x^s, α);
      α=α(t);
   }
}

operation(U, x^l, x^h, x^s, α)
{
   S=select n+1 vertices from U-{x^h};
   x^0=1/(n+1) Σ_{x∈S} x;
   x^r=(1+a)x^0 − a x^h;
   if((f(x^r),μ(x^r)) <_α (f(x^l),μ(x^l))) {
      x^e=c x^r + (1−c)x^0;
      if((f(x^e),μ(x^e)) <_α (f(x^l),μ(x^l)))  x^h=x^e;
      else x^h=x^r;
   }
   else if((f(x^r),μ(x^r)) ≤_α (f(x^s),μ(x^s)))  x^h=x^r;
   else {
      if((f(x^r),μ(x^r)) <_α (f(x^h),μ(x^h)))  x^h=x^r;
      x^c=b x^h + (1−b)x^0;
      if((f(x^c),μ(x^c)) <_α (f(x^h),μ(x^h)))  x^h=x^c;
      else x^h=b x^h + (1−b)x^l;
   }
}

mutation(x^h) {
   i=select a dimension randomly from [1,n];
   if(x^h is feasible) {
      sample r ∈ u(0,1);
      if(r ≤ 0.5)
        x_i'=search boundary from x_i^h toward l_i;  ······ eqs.(25) and (27)
      else
        x_i'=search boundary from x_i^h toward u_i;  ······ eqs.(26) and (27)
   }
   else
     x_i'=search maximum point of μ from x_i^h;  ········· eq.(27)
   x_i^h=x_i';
}
```

Fig. 1. Algorithm of the $\alpha$ constrained simplex method with control of the $\alpha$ level, where $\alpha(t)$ is the function for controlling the $\alpha$ level and $u(0,1)$ is a uniform random number generator in [0,1].

The parameters for the $\alpha$ constrained method are as follows. Every satisfaction level is defined as a piecewise linear function in (3) and (4), and the parameters for the satisfaction level are $b_i = b_j = 1000$. The satisfaction levels are combined by minimization in (5). The $\alpha$ level is controlled according to the (28). The initial $\alpha$ level $\alpha_0$ is the mean value of the best satisfaction level and the average of all satisfaction levels in the initial search points. The $\alpha$ level is updated when the number of itera-

TABLE I
EXPERIMENTAL RESULTS ON 13 BENCHMARK PROBLEMS USING STANDARD SETTINGS; 30 INDEPENDENT RUNS WERE PERFORMED

| $f$ | optimal | best | median | mean | worst | st. dev. | #func | #const | time(s) |
|---|---|---|---|---|---|---|---|---|---|
| g01 | -15.000 | -14.9999998 | -14.9999990 | -14.9999951 | -14.9999765 | 6.4e-06 | 83422.0 | 303161.5 | 0.43 |
| ↑g02 | 0.803619 | 0.8036191 | 0.7851627 | 0.7841868 | 0.7542585 | 1.3e-02 | 102677.1 | 328930.6 | 0.99 |
| ↑g03 | 1.000 | 1.0005001 | 1.0005001 | 1.0005001 | 1.0005001 | 8.5e-14 | 85426.2 | 310968.5 | 0.37 |
| g04 | -30665.539 | -30665.5386718 | -30665.5386718 | -30665.5386718 | -30665.5386718 | 4.2e-11 | 74081.6 | 305343.4 | 0.28 |
| g05 | 5126.498 | 5126.4967140 | 5126.4967140 | 5126.4967140 | 5126.4967140 | 3.5e-11 | 64982.6 | 308389.3 | 0.38 |
| g06 | -6961.814 | -6961.8138756 | -6961.8138756 | -6961.8138756 | -6961.8138756 | 1.3e-10 | 37492.8 | 293366.6 | 0.20 |
| g07 | 24.306 | 24.3062096 | 24.3062158 | 24.3062625 | 24.3068044 | 1.3e-04 | 86121.0 | 317587.3 | 0.41 |
| ↑g08 | 0.095825 | 0.0958250 | 0.0958250 | 0.0958250 | 0.0958250 | 3.8e-13 | 128315.2 | 306248.0 | 0.28 |
| g09 | 680.630 | 680.6300574 | 680.6300574 | 680.6300574 | 680.6300574 | 2.9e-10 | 85073.6 | 323426.5 | 0.33 |
| g10 | 7049.248 | 7049.2480207 | 7049.2480208 | 7049.2480217 | 7049.2480468 | 4.7e-06 | 80052.9 | 316036.5 | 0.49 |
| g11 | 0.750 | 0.7499000 | 0.7499000 | 0.7499000 | 0.7499000 | 4.9e-16 | 84891.1 | 308124.7 | 0.32 |
| ↑g12 | 1.000000 | 1.0000000 | 1.0000000 | 1.0000000 | 1.0000000 | 3.9e-10 | 13481.8 | 30283.4 | 0.70 |
| g13 | 0.053950 | 0.0539415 | 0.0539415 | 0.0667702 | 0.4388026 | 6.9e-02 | 73527.6 | 311144.2 | 0.43 |

tion $t$ becomes the multiple of $T_\alpha$. After the number of iterations exceeds $T_{\max}/2$, the $\alpha$ level is set to 1 to obtain solutions with minimum constraint violation

$$\alpha(t)=\begin{cases}\frac{1}{2}\left(\max_i \mu(\boldsymbol{x}^i)+\frac{\sum_i \mu(\boldsymbol{x}^i)}{N}\right), & \text{if } t=0\\ (1-\beta)\alpha(t-1)+\beta, & \text{if } 0<t\le\frac{T_{\max}}{2} \text{ and } (t \bmod T_\alpha)=0\\ \alpha(t-1), & \text{if } 0<t\le\frac{T_{\max}}{2} \text{ and } (t \bmod T_\alpha)\neq 0\\ 1, & \text{if } t>\frac{T_{\max}}{2}.\end{cases} \tag{28}$$

The $\alpha$ level is controlled using $\beta = 0.03$ and $T_\alpha = 50$ for all problems. The parameters for the simplex method are as follows: The number of search points $N = 90$, the maximum iterations $T_{\max} = 85\,000$, the mutation rate $P_m = 0.06$ and the operation parameters $a = 1$, $b = 0.75$, and $c = 2$ in (16), (17) and (18) are common settings. In g12, $T_{\max} = 8500$ is used. In this paper, 30 independent runs are performed. The effect of parameters will be discussed later.

### B. Experimental Results

Table I summarizes the experimental results. The table shows the known "optimal" solution for each problem and statistics for the 30 independent runs. These include the best, median, mean, worst, and standard deviation of the objective values found. Also, the average number of evaluations of the objective function and the constraints in each run are shown in the columns labeled #func and #const, respectively. The average execution time (seconds) in each run using a 1.3 GHz Mobile Pentium III notebook PC is shown in the column labeled time(s). The problems marked by an up arrow are maximization problems.

For every problem, the best solution is almost equivalent to the optimal solution. For problems g04, g06, g08, g09, and g12, the optimal solutions are found consistently in all 30 runs. For problems g01, g07, and g10, the near-optimal solutions are found in all 30 runs. For problems g03, g05, g11, and g13 which contain equality constraints, objective values that were better than the optimal values were found, because the equality constraints are relaxed. The constant violation, which is given by the maximum of the constraint functions, or $\max_{i,j}\{0, g_i(\boldsymbol{x}), |h_j(\boldsymbol{x})|\}$, for each problem was just $10^{-4}$, which was same as the value of $\delta$. For problem g13, the solutions equivalent to the best solution were found in 29 runs and the exception was only once. For problem g02, the optimal solutions were not consistently found. This problem has many local optima (maxima) with high peak near the global optimum. Once a local optimum, which is much better than other search points, has found, the $\alpha$ constrained simplex method tends to converge to it and the global optimum is sometimes skipped. To avoid this situation, it is thought that adjusting algorithm parameters to decrease convergence speed is useful, such as increasing the mutation rate $P_m$ and increasing the contraction parameter $b$.

The $\alpha$ constrained simplex method is a very fast algorithm. The execution times ranged from 0.2 to 0.99 s using a notebook PC. In all problems, the execution time is less than 1 s and in 11 problems the execution time is less than 0.5 s. The number of evaluations of the constraints ranged from 290 000 to 330 000 with the exception of g12. The number of evaluations of the objective function ranged between 13 000–130 000. In the $\alpha$ constrained simplex method, the objective function and the constraints are treated separately. So, when the order relation of the search points can be decided only by the satisfaction level of the constraints, the objective function is not evaluated. The number of evaluations of the objective function is less than the number of evaluations of the constraints even when the mutation is not applied, or the mutation rate $P_m$ is zero. This nature contributes to the efficiency of the algorithm especially when the objective function is computationally demanding.

### C. Comparison With the Stochastic Ranking Method

To show the effectiveness of the $\alpha$ constrained simplex method, the solutions found by this method are compared with those found by Runarsson and Yao's stochastic

TABLE II
COMPARISON BETWEEN OUR (INDICATED BY $\alpha$SIMPLEX) AND RUNARSSON AND YAO'S (INDICATED BY RY [12]) ALGORITHMS

| $f$ | optimal | Best Result | | Median Result | | Mean Result | | Worst Result | |
|---|---|---|---|---|---|---|---|---|---|
| | | $\alpha$Simplex | RY | $\alpha$Simplex | RY | $\alpha$Simplex | RY | $\alpha$Simplex | RY |
| g01 | -15.000 | -15.000 | -15.000 | -15.000 | -15.000 | -15.000 | -15.000 | -15.000 | -15.000 |
| ↑g02 | 0.803619 | **0.803619** | 0.803515 | 0.785163 | **0.785800** | **0.784187** | 0.781975 | **0.754259** | 0.726288 |
| ↑g03 | 1.000 | **1.001** | 1.000 | **1.001** | 1.000 | **1.001** | 1.000 | **1.001** | 1.000 |
| g04 | -30665.539 | -30665.539 | -30665.539 | -30665.539 | -30665.539 | -30665.539 | -30665.539 | -30665.539 | -30665.539 |
| g05 | 5126.498 | 5126.497 | 5126.497 | **5126.497** | 5127.372 | **5126.497** | 5128.881 | **5126.497** | 5142.472 |
| g06 | -6961.814 | -6961.814 | -6961.814 | -6961.814 | -6961.814 | **-6961.814** | -6875.940 | **-6961.814** | -6350.262 |
| g07 | 24.306 | **24.306** | 24.307 | **24.306** | 24.357 | **24.306** | 24.374 | **24.306** | 24.642 |
| ↑g08 | 0.095825 | 0.095825 | 0.095825 | 0.095825 | 0.095825 | 0.095825 | 0.095825 | 0.095825 | 0.095825 |
| g09 | 680.630 | 680.630 | 680.630 | **680.630** | 680.641 | **680.630** | 680.656 | **680.630** | 680.763 |
| g10 | 7049.248 | **7049.248** | 7054.316 | **7049.248** | 7372.613 | **7049.248** | 7559.192 | **7049.248** | 8835.655 |
| g11 | 0.750 | 0.750 | 0.750 | 0.750 | 0.750 | 0.750 | 0.750 | 0.750 | 0.750 |
| ↑g12 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 |
| g13 | 0.053950 | **0.053942** | 0.053957 | **0.053942** | 0.057006 | **0.066770** | 0.067543 | 0.438803 | **0.216915** |

ranking method [12]. In [12], 30 independent runs were performed, the maximum number of evaluations in each run was $1750 \times 200 = 350\,000$ except for g12 ($175 \times 200 = 35\,000$) and all equality constraints were relaxed using $\delta = 10^{-4}$. Table II shows the comparison of the two methods. The better cases are highlighted using boldface. The results of the $\alpha$ constrained simplex method were taken from Table I, where all equality constraints were relaxed using $\delta = 10^{-4}$ and the maximum number of evaluations was about 30 000 for problem g12 and about from 290 0000 to 330 000 for the other problems. The solutions found by the stochastic ranking method were very high-quality solutions that were equivalent to the known optimal solutions. Nevertheless, the $\alpha$ constrained simplex method found better solutions for problems g02, g03, g07, g10, and g13. Also, the stability of the $\alpha$ constrained simplex method was better than that of the stochastic ranking method for problems g05, g06, g07, g09, and g10. Therefore, the performance of the $\alpha$ constrained simplex method is at least as good as the performance of the stochastic ranking method.

In particular, for problem g10, better solutions than the best solution in [12] was found in all 30 runs. The value of the objective function $f$, all variables $x_1, x_2, \ldots, x_8$ and all constraints $g_1, g_2, \ldots, g_6$ of the best solution found by the $\alpha$ constrained simplex method and the best solution in [12] are shown in Table III.

TABLE III
BEST SOLUTION OF PROBLEM g10

| | $\alpha$Simplex | best in [12] |
|---|---|---|
| $f$ | 7049.248020689827 | 7049.3307 |
| $x_1$ | 579.3072292519911 | 579.3167 |
| $x_2$ | 1359.969167520341 | 1359.943 |
| $x_3$ | 5109.971623917496 | 5110.071 |
| $x_4$ | 182.0177450823991 | 182.0174 |
| $x_5$ | 295.6011350455768 | 295.5985 |
| $x_6$ | 217.9822549134181 | 217.9799 |
| $x_7$ | 286.4166100330206 | 286.4162 |
| $x_8$ | 395.6011350451201 | 395.5979 |
| $g_1$ | -1.04570e-11 | -6.75e-06 |
| $g_2$ | -9.50413e-12 | -6.75e-06 |
| $g_3$ | -4.56735e-12 | -6.00e-06 |
| $g_4$ | -5.61092e-07 | -0.040709 |
| $g_5$ | -2.99309e-06 | -0.042268 |
| $g_6$ | -3.35774e-06 | -0.283957 |

*D. Solving Problems With Equality Constraints*

It is very difficult to solve problems with equality constraints directly without relaxing the constraints. In this case, it is difficult to find a feasible point. Even if a feasible point is found, almost all points in the neighborhood of the point are infeasible and there is little chance to find a better point, which is feasible and has better objective value than the point. Thus, all search points tends to converge to the feasible point, which is not optimal. To avoid this situation, problems with equality constraints are often solved by relaxing the equality constraints and expanding the feasible region. However, $\alpha$ constrained simplex method can solve problems with equality constraints directly by controlling the $\alpha$ level and reducing expanded feasible region to the original feasible region internally. Even in $\alpha$ constrained simplex method, the maximization of the satisfaction level is not easy and search points often converge to a point which has higher satisfaction level. Thus, it is necessary to use many search points and adopt slower control of the $\alpha$ level. In this experiment, the number of search points $N = 180$ and the parameter for controlling the $\alpha$ level $\beta = 0.015$ (and $\delta = 0$) are used and other settings are same as the standard settings.

Table IV summarizes the statistics of experimental results for the 30 independent runs. The rows labeled violation show the best, median, mean, worst, and standard deviation of the constraint violation of each problem. In the case of solving problems with relaxing constraints ($\delta = 10^{-4}$), constraint violation

TABLE IV
EXPERIMENTAL RESULTS ON PROBLEMS WITH EQUALITY CONSTRAINTS USING $N = 180$, $\beta = 0.015$, AND $\delta = 0$

| $f$ | optimal | best | median | mean | worst | st. dev. | #func | #const | time(s) |
|---|---|---|---|---|---|---|---|---|---|
| ↑g03 | 1.000 | 1.0000000 | 0.9999994 | 0.9999993 | 0.9999968 | 6.6e-07 | 46174.7 | 320882.4 | 0.45 |
| violation | | 0.00e+00 | 0.00e+00 | 1.19e-14 | 1.73e-13 | 3.6e-14 | | | |
| g05 | 5126.498 | 5126.4981104 | 5126.4985042 | 5126.4989231 | 5126.5033258 | 1.1e-03 | 40804.3 | 297482.1 | 0.45 |
| violation | | 2.33e-12 | 5.31e-12 | 6.09e-12 | 1.16e-11 | 2.1e-12 | | | |
| g11 | 0.750 | 0.7500000 | 0.7500001 | 0.7500002 | 0.7500013 | 2.9e-07 | 43538.9 | 314637.5 | 0.30 |
| violation | | 1.07e-17 | 9.49e-17 | 1.41e-16 | 6.08e-16 | 1.3e-16 | | | |
| g13 | 0.053950 | 0.0539498 | 0.0539499 | 0.0539499 | 0.0539500 | 2.4e-08 | 42628.8 | 300291.7 | 0.36 |
| violation | | 4.17e-14 | 1.34e-13 | 1.32e-13 | 2.18e-13 | 4.6e-14 | | | |

TABLE V
EXPERIMENTAL RESULTS ON 13 BENCHMARK PROBLEMS WITH VARYING $P_m$; 30 INDEPENDENT RUNS WERE PERFORMED

| $P_m$ | 0.0 | 0.02 | 0.04 | 0.06 | 0.08 | 0.1 |
|---|---|---|---|---|---|---|
| g01 | -13.806016 | -14.933307 | **-14.999999** | -14.999995 | -14.999964 | -14.999917 |
| ↑g02 | 0.346874 | 0.743445 | 0.783298 | 0.784187 | 0.788250 | **0.791348** |
| ↑g03 | 0.566772 | 0.976787 | 0.992290 | **1.000500** | **1.000500** | **1.000500** |
| g04 | -30665.538672 | -30665.538672 | -30665.538672 | -30665.538672 | -30665.538672 | -30665.538672 |
| g05 | 5126.496714 | 5126.496714 | 5126.496714 | 5126.496714 | 5126.496714 | 5126.496714 |
| g06 | -6961.813876 | -6961.813876 | -6961.813876 | -6961.813876 | -6961.813876 | -6961.813876 |
| g07 | **24.306209** | **24.306209** | **24.306209** | 24.306262 | 24.310081 | 24.322305 |
| ↑g08 | 0.095825 | 0.095825 | 0.095825 | 0.095825 | 0.095825 | 0.095825 |
| g09 | **680.630057** | **680.630057** | **680.630057** | **680.630057** | **680.630057** | 680.630058 |
| g10 | 7049.248027 | **7049.248021** | **7049.248021** | 7049.248022 | 7049.248025 | 7049.248089 |
| g11 | 0.749900 | 0.749900 | 0.749900 | 0.749900 | 0.749900 | 0.749900 |
| ↑g12 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 |
| g13 | 0.092428 | 0.079599 | 0.143742 | **0.066770** | 0.118085 | 0.092428 |

was $10^{-4}$. In this experiment, $\alpha$ constrained simplex method found the very good solutions with constraint violation ranged from 0 to $10^{-11}$. The objective value of each problem is almost equal to the optimal value. These results show that the $\alpha$ constrained simplex method can solve problems with equality constraints directly by controlling the $\alpha$ level. This feature is an important advantage of the $\alpha$ constrained simplex method.

## V. DISCUSSION

In this section, the effect of algorithm parameters such as the mutation rate and the number of search points will be discussed.

### A. Effect of the Mutations

The mutations are effective for finding solutions around the boundary of the feasible region and controlling the convergence speed of the search points. However, mutations with a high mutation rate tend to increase the number of evaluations of the constraints and, hence, deteriorate the speed of the algorithm in some cases. Therefore, suitable mutation rates must be selected. Table V summarizes the mean of the objective values in the case of the mutation rate $\mathrm{P}_m$ alone being changed to 0.0, 0.02, 0.04, 0.06, 0.08, and 0.1 from standard settings. The better cases are highlighted using boldface. In the case of $\mathrm{P}_m = 0$, the results for problems g01, g02, and g03 are much worse than the results when $\mathrm{P}_m$ is greater than or equal to 0.02. Also, in the case of $P_m = 0.1$, the result of problem g02 is much better than the other results. These results highlight the necessity of the mutations. These results show that a mutation rate between 0.06 and 0.1 is an appropriate setting for many problems.

### B. Effect of Multiple Simplexes

The number of search points $N$ adjusts the diversity of search points and the convergence speed of search process. If $N$ is too small, although the convergence speed is very high, the diversity becomes low and the search points often converge to a local optimum. If $N$ is too large, the convergence speed becomes low and the search points cannot reach the global optimum. Therefore, suitable number of search points must be selected. Table VI summarizes the mean of the objective values in the case of the number of search points $N$ alone being changed to 22, 45, 90, 135, 180, and 270 from standard settings. In the case of $N = 22$, the results for problems g01, g02, g04, g05, g07, g09, g10, and g13 are worse than the other results. These results highlight the necessity of the multiple simplexes. Also, these results

TABLE VI
EXPERIMENTAL RESULTS ON 13 BENCHMARK PROBLEMS WITH VARYING $N$; 30 INDEPENDENT RUNS WERE PERFORMED

| $N$ | 22 | 45 | 90 | 135 | 180 | 225 |
|---|---|---|---|---|---|---|
| g01 | -13.860984 | **-15.000000** | -14.999995 | -14.999945 | -14.999783 | -14.999186 |
| ↑g02 | 0.779642 | **0.784786** | 0.784187 | 0.782868 | 0.782529 | 0.780890 |
| ↑g03 | 1.000500 | 1.000500 | 1.000500 | 1.000500 | 1.000500 | 1.000500 |
| g04 | -30665.398833 | **-30665.538672** | **-30665.538672** | **-30665.538672** | **-30665.538672** | **-30665.538672** |
| g05 | 5126.496715 | **5126.496714** | **5126.496714** | **5126.496714** | **5126.496714** | **5126.496714** |
| g06 | -6961.813876 | -6961.813876 | -6961.813876 | -6961.813876 | -6961.813876 | -6961.813876 |
| g07 | 24.865150 | 24.323806 | 24.306262 | **24.306213** | 24.306214 | 24.306218 |
| ↑g08 | 0.095825 | 0.095825 | 0.095825 | 0.095825 | 0.095825 | 0.095825 |
| g09 | 680.630058 | **680.630057** | **680.630057** | **680.630057** | **680.630057** | **680.630057** |
| g10 | 7053.218625 | **7049.248022** | **7049.248022** | 7049.248023 | 7049.248046 | 7049.248251 |
| g11 | 0.749900 | 0.749900 | 0.749900 | 0.749900 | 0.749900 | 0.749900 |
| ↑g12 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 |
| g13 | 0.328149 | 0.162449 | 0.066770 | 0.066770 | **0.054194** | 0.187914 |

TABLE VII
EXPERIMENTAL RESULTS ON 13 BENCHMARK PROBLEMS WITH VARYING $b$; 30 INDEPENDENT RUNS WERE PERFORMED

| $b$ | 0.5 | 0.6 | 0.7 | 0.75 | 0.8 | 0.9 |
|---|---|---|---|---|---|---|
| g01 | **-15.000000** | -14.999999 | -14.999999 | -14.999995 | -14.999532 | -14.997564 |
| ↑g02 | 0.778846 | 0.778605 | 0.784519 | 0.784187 | **0.785748** | 0.780604 |
| ↑g03 | 1.000500 | 1.000500 | 1.000500 | 1.000500 | 1.000500 | 1.000500 |
| g04 | **-30665.538672** | **-30665.538672** | **-30665.538672** | **-30665.538672** | **-30665.538672** | -30665.538669 |
| g05 | 5126.496714 | 5126.496714 | 5126.496714 | 5126.496714 | 5126.496714 | 5126.496714 |
| g06 | -6961.813876 | -6961.813876 | -6961.813876 | -6961.813876 | -6961.813876 | -6961.813876 |
| g07 | 24.308843 | **24.306214** | 24.306215 | 24.306262 | 24.306321 | 24.308757 |
| ↑g08 | 0.095825 | 0.095825 | 0.095825 | 0.095825 | 0.095825 | 0.095825 |
| g09 | **680.630057** | **680.630057** | **680.630057** | **680.630057** | **680.630057** | 680.630249 |
| g10 | **7049.248021** | **7049.248021** | **7049.248021** | 7049.248022 | 7049.248025 | 7049.259718 |
| g11 | 0.749900 | 0.749900 | 0.749900 | 0.749900 | 0.749900 | 0.749900 |
| ↑g12 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 |
| g13 | 0.206813 | 0.130914 | 0.130914 | **0.066770** | 0.079599 | 0.150943 |

show that the number of search points between 90 and 180 is an appropriate setting for many problems.

### C. Effect of Contraction Parameter

The contraction parameter $b$ adjusts the convergence speed of search process. When $b$ is small, the search points approach its centroid rapidly. Although the convergence speed of the search points is high, the search points might skip the global optimum and converge to a local optimum. When $b$ is large, although the risk of convergence to a local optimum becomes low, the convergence speed becomes low and the search points might not reach the global optimum. Therefore, suitable contraction parameter must be selected. Table VII summarizes the mean of the objective values in the case of the contraction parameter $b$ alone being changed to 0.5, 0.6, 0.7, 0.75, 0.8, and 0.9 from standard settings. In the case of $b = 0.5$, the results for problems g07 and g13 are worse than the other results and the result for problem g02 is not good, although the result for problem g01 is good. These results show that a value of the contraction parameter between 0.6 and 0.8 is an appropriate setting for many problems.

### D. Effect of Parameters for the Satisfaction Level

The parameters $b_i$ and $b_j$ are used to define the satisfaction level $\mu$. If $b_i$ or $b_j$ is too small, the satisfaction level becomes zero in wide area and it becomes difficult to maximize the level. Therefore, $b_i$ and $b_j$ should be large enough to let the level be greater than zero in almost the whole search space. In this study, the same value of $b_i$ and $b_j$ is selected for all constraints. In this case, if the value is large enough, the value does not affect the order relation of the satisfaction levels, it does not affect the $\alpha$ level comparison and the result of optimization does not depend on the value when the same $\alpha$ level is hold. Thus, it is almost needless to adjust the value of the parameters when the value is large enough. However, if the value is extremely large, there is risk that the precision of the $\alpha$ level comparison decreases because the precision of real numbers is limited on computers. Table VIII summarizes the mean of the objective values in the

TABLE VIII
EXPERIMENTAL RESULTS ON 13 BENCHMARK PROBLEMS WITH VARYING $b_i$ AND $b_j$; 30 INDEPENDENT RUNS WERE PERFORMED

| $b_i$,$b_j$ | 1 | 10 | 100 | 1000 | 10000 | 100000 |
|---|---|---|---|---|---|---|
| g01 | **-14.999999** | -14.999998 | -14.999995 | -14.999995 | -14.999991 | -14.999704 |
| ↑g02 | 0.781421 | 0.781421 | 0.783715 | 0.784187 | 0.786083 | **0.787600** |
| ↑g03 | **1.000500** | **1.000500** | 1.000499 | **1.000500** | **1.000500** | **1.000500** |
| g04 | -30665.538672 | -30665.538672 | -30665.538672 | -30665.538672 | -30665.538672 | -30665.538672 |
| g05 | 5126.496714 | 5126.496714 | 5126.496714 | 5126.496714 | 5126.496714 | 5126.496714 |
| g06 | -6961.813876 | -6961.813876 | -6961.813876 | -6961.813876 | -6961.813876 | -6961.813876 |
| g07 | **24.306212** | 24.306221 | 24.306245 | 24.306262 | 24.306230 | 24.306230 |
| ↑g08 | 0.095825 | 0.095825 | 0.095825 | 0.095825 | 0.095825 | 0.095825 |
| g09 | 680.630057 | 680.630057 | 680.630057 | 680.630057 | 680.630057 | 680.630057 |
| g10 | **7049.248021** | **7049.248021** | **7049.248021** | 7049.248022 | 7049.248022 | 7049.248022 |
| g11 | 0.749900 | 0.749900 | 0.749900 | 0.749900 | 0.749900 | 0.749900 |
| ↑g12 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 |
| g13 | 0.246372 | 0.105256 | 0.079599 | **0.066770** | 0.079599 | 0.105256 |

TABLE IX
EXPERIMENTAL RESULTS ON 13 BENCHMARK PROBLEMS WITH VARYING $\beta$; 30 INDEPENDENT RUNS WERE PERFORMED

| $\beta$ | 0.015 | 0.03 | 0.045 | 0.06 | 0.075 | 0.1 |
|---|---|---|---|---|---|---|
| g01 | -14.999975 | -14.999995 | -14.999996 | -14.999998 | -14.999999 | **-15.000000** |
| ↑g02 | **0.787168** | 0.784187 | 0.783572 | 0.785179 | 0.782484 | 0.780941 |
| ↑g03 | 1.000324 | **1.000500** | **1.000500** | **1.000500** | **1.000500** | **1.000500** |
| g04 | -30665.538672 | -30665.538672 | -30665.538672 | -30665.538672 | -30665.538672 | -30665.538672 |
| g05 | 5126.496714 | 5126.496714 | 5126.496714 | 5126.496714 | 5126.496714 | 5126.496714 |
| g06 | -6961.813876 | -6961.813876 | -6961.813876 | -6961.813876 | -6961.813876 | -6961.813876 |
| g07 | 24.317074 | 24.306262 | 24.306213 | **24.306211** | 24.306213 | **24.306211** |
| ↑g08 | 0.095825 | 0.095825 | 0.095825 | 0.095825 | 0.095825 | 0.095825 |
| g09 | 680.630057 | 680.630057 | 680.630057 | 680.630057 | 680.630057 | 680.630057 |
| g10 | 7049.248036 | 7049.248022 | **7049.248021** | **7049.248021** | **7049.248021** | **7049.248021** |
| g11 | 0.749900 | 0.749900 | 0.749900 | 0.749900 | 0.749900 | 0.749900 |
| ↑g12 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 |
| g13 | 0.092428 | **0.066770** | **0.066770** | 0.143742 | 0.133357 | 0.253629 |

case of the value of $b_i$ and $b_j$ alone being changed to 1, 10, 100, 1000, 10 000, and 100 000 from standard settings. The value of $b_i$ and $b_j$ little affect the search results. These results show that a value of the parameters for the satisfaction level between 10 and 100 000 is an appropriate setting for many problems.

### *E. Effect of Parameters for Controlling $\alpha$ Level*

In $\alpha$ constrained simplex method, feasible region can be expanded by relaxing the $\alpha$ level. The expanded feasible region can be reduced to the original feasible region by increasing the $\alpha$ level to 1. The parameter $\beta$ adjusts the speed of increasing the $\alpha$ level and the speed of reducing the expanded feasible region. If $\beta$ is small enough, the $\alpha$ level approaches to 1 gradually and the risk that the search points converge to a local optimum is low. But if $\beta$ is too small, the search points must search wide area including infeasible area and the search efficiency becomes low. In this case, the expanded region is still wide at $T_{\max}/2$ and when the $\alpha$ level becomes 1, the search points might converge to a feasible point rapidly and skip the global optimum. Therefore, suitable control parameter must be selected. Table IX summarizes the mean of the objective values in the case of the control parameter $\beta$ alone being changed to 0.015, 0.03, 0.045, 0.06, 0.075, and 0.1 from standard settings. The value of $\beta$ did not affect the results much except for g13. These results show that control parameter $\beta$ between 0.015 and 0.045 is an appropriate setting for many problems.

## VI. CONCLUSION

The nonlinear simplex method is a well-known mathematical method for unconstrained nonlinear optimization. In this paper, we proposed the improved $\alpha$ constrained simplex method by integrating the mathematical method with mutations from evolutionary computation, and applying the $\alpha$ constrained method. Also, to control the convergence speed, operations on multiple simplexes, replacement of the reduction operation with the contraction operation, and the change of the algorithm parameters were introduced. The original $\alpha$ constrained simplex method could only solve a limited number of simple problems. On the other hand, we showed that the improved $\alpha$ constrained simplex method could solve 13 standard benchmark problems very quickly. Also, by comparing the $\alpha$ constrained simplex method with the stochastic ranking method which is a high-performance algorithm for constrained optimization, it was shown that the $\alpha$ constrained simplex method was an efficient and stable algorithm. The experiments for parameter settings were performed and it was shown that the $\alpha$ constrained simplex method could search high-quality solutions when the parameters were changed among appropriate range.

In the penalty function method, feasible points can be found by increasing the penalty coefficient toward infinity in a theoretical sense, although it is difficult to do so computationally. In the $\alpha$ constrained simplex method, feasible points can be found by increasing the $\alpha$ level to 1 and this is easy to do computationally. If the constraints of a point are not satisfied, the satisfaction level is maximized, and the point will become feasible. In the penalty function method, the objective function and the constraints must be evaluated for every new search point. However, in the $\alpha$ constrained simplex method, the objective function and the constraints are treated separately, and the evaluation of the objective function can often be omitted. Therefore, the $\alpha$ constrained simplex method can find feasible points efficiently compared with the penalty function method.

In the future, we will apply the $\alpha$ constrained simplex method to various problems in the real world.

## APPENDIX

### A. Proof of Theorem 3

From theorem 1, $\hat{\boldsymbol{x}}_n$ is an optimal solution of the problem $(\mathrm{P}^{\alpha_n})$ and $f(\hat{\boldsymbol{x}}_n) \leq f(\boldsymbol{x}^*)$.

Let $\bar{\boldsymbol{x}}$ be an accumulation point of the sequence $\{\hat{\boldsymbol{x}}_n\}$ and let $\{\hat{\boldsymbol{x}}_{n_k}\}$ be a subsequence of $\{\hat{\boldsymbol{x}}_n\}$ converging to $\bar{\boldsymbol{x}}$. That is, $\lim_{k\to\infty} \hat{\boldsymbol{x}}_{n_k} = \bar{\boldsymbol{x}}$. By the continuity of $f$, $\lim_{k\to\infty} f(\hat{\boldsymbol{x}}_{n_k}) = f(\bar{\boldsymbol{x}})$. Thus, $f(\boldsymbol{x}^*) \geq \lim_{k\to\infty} f(\hat{\boldsymbol{x}}_{n_k}) = f(\bar{\boldsymbol{x}})$.

On the other hand, from the continuity of $\mu$ and $\lim_{k\to\infty} \alpha_{n_k} = 1$, $\mu(\bar{\boldsymbol{x}}) = \lim_{k\to\infty} \mu(\hat{\boldsymbol{x}}_{n_k}) \geq \lim_{k\to\infty} \alpha_{n_k} = 1$. Then, $\bar{\boldsymbol{x}}$ is a feasible solution of the problem $(\mathrm{P}^1)$ and $f(\boldsymbol{x}^*) \leq f(\bar{\boldsymbol{x}})$.

Therefore, $f(\bar{\boldsymbol{x}}) = f(\boldsymbol{x}^*)$. That is, $\bar{\boldsymbol{x}}$ is an optimal solution of the problem $(\mathrm{P}^1)$. ∎

### B. Test Problems

`g01` [32]:

$$\begin{aligned}
\text{minimize} \quad f(\boldsymbol{x}) &= 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i \\
\text{subject to} \quad g_1(\boldsymbol{x}) &= 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0 \\
g_2(\boldsymbol{x}) &= 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0 \\
g_3(\boldsymbol{x}) &= 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0 \\
g_4(\boldsymbol{x}) &= -8x_1 + x_{10} \leq 0 \\
g_5(\boldsymbol{x}) &= -8x_2 + x_{11} \leq 0 \\
g_6(\boldsymbol{x}) &= -8x_3 + x_{12} \leq 0 \\
g_7(\boldsymbol{x}) &= -2x_4 - x_5 + x_{10} \leq 0 \\
g_8(\boldsymbol{x}) &= -2x_6 - x_7 + x_{11} \leq 0 \\
g_9(\boldsymbol{x}) &= -2x_8 - x_9 + x_{12} \leq 0, \\
& 0 \leq x_i \leq 1 \ (i = 1, \cdots, 9), \\
& 0 \leq x_i \leq 100 \ (i = 10, 11, 12), \\
& 0 \leq x_{13} \leq 1.
\end{aligned}$$

The optimal solution is $\boldsymbol{x}^* = (1,1,1,1,1,1,1,1,1,3,3,3,1)$ and the optimal value is $f(\boldsymbol{x}^*) = -15$.

`g02` [31]:

$$\text{maximize } f(\boldsymbol{x}) = \frac{\left|\sum_{i=1}^{n} \cos^4 x_i - 2\prod_{i=1}^{n} \cos^2 x_i\right|}{\sqrt{\sum_{i=1}^{n} i x_i^2}}$$

$$\begin{aligned}
\text{subject to } g_1(\boldsymbol{x}) &= 0.75 - \prod_{i=1}^{n} x_i \leq 0 \\
g_2(\boldsymbol{x}) &= \sum_{i=1}^{n} x_i - 7.5n \leq 0, \\
& 0 \leq x_i \leq 10 \ (i = 1, \cdots, n), \ n = 20.
\end{aligned}$$

The maximum value is unknown. The known best value is $f(\boldsymbol{x}) = 0.803619$ [12].

`g03` [33]:

$$\begin{aligned}
\text{maximize } f(\boldsymbol{x}) &= (\sqrt{n})^n \prod_{i=1}^{n} x_i \\
\text{subject to } h_1(\boldsymbol{x}) &= \sum_{i=1}^{n} x_i^2 - 1 = 0, \\
& 0 \leq x_i \leq 1 \ (i = 1, \cdots, n), \quad n = 10.
\end{aligned}$$

The optimal solution $\boldsymbol{x}_i^* = (1/\sqrt{n})$ $(i = 1, \cdots, n)$ and the optimal value $f(\boldsymbol{x}^*) = 1$.

`g04` [34]:

$$\begin{aligned}
&\text{minimize} \\
&\quad f(\boldsymbol{x}) = 5.3578547x_3^2 + 0.8356891x_1x_5 \\
&\qquad\quad + 37.293239x_1 - 40792.141 \\
&\text{subject to} \\
&\quad g_1(\boldsymbol{x}) = 85.334407 + 0.0056858x_2x_5 \\
&\qquad\quad + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0 \\
&\quad g_2(\boldsymbol{x}) = -85.334407 - 0.0056858x_2x_5 \\
&\qquad\quad - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0 \\
&\quad g_3(\boldsymbol{x}) = 80.51249 + 0.0071317x_2x_5 \\
&\qquad\quad + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0 \\
&\quad g_4(\boldsymbol{x}) = -80.51249 - 0.0071317x_2x_5 \\
&\qquad\quad - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0 \\
&\quad g_5(\boldsymbol{x}) = 9.300961 + 0.0047026x_3x_5 \\
&\qquad\quad + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0 \\
&\quad g_6(\boldsymbol{x}) = -9.300961 - 0.0047026x_3x_5 \\
&\qquad\quad - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0, \\
&\qquad\quad 78 \leq x_1 \leq 102, \ 33 \leq x_2 \leq 45, \\
&\qquad\quad 27 \leq x_i \leq 45 \ (i = 3, 4, 5).
\end{aligned}$$

The optimal solution $\boldsymbol{x}^* = (78, 33, 29.995256025682, 45, 36.775812905788)$ and the optimal value $f(\boldsymbol{x}^*) = -30665.539$.

g05 [35]:

$$\text{minimize} \quad f(\boldsymbol{x}) = 3x_1 + 0.000001x_1^3 + 2x_2 + \frac{0.000002}{3}x_2^3$$

subject to

$$\begin{aligned} g_1(\boldsymbol{x}) &= x_3 - x_4 - 0.55 \le 0 \\ g_2(\boldsymbol{x}) &= -x_3 + x_4 - 0.55 \le 0 \\ h_3(\boldsymbol{x}) &= 1000\sin(-x_3 - 0.25) \\ &\quad + 1000\sin(-x_4 - 0.25) + 894.8 - x_1 = 0 \\ h_4(\boldsymbol{x}) &= 1000\sin(x_3 - 0.25) \\ &\quad + 1000\sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0 \\ h_5(\boldsymbol{x}) &= 1000\sin(x_4 - 0.25) \\ &\quad + 1000\sin(x_4 - x_3 - 0.25) + 1294.8 = 0, \\ &\quad 0 \le x_i \le 1200 \ (i = 1, 2), \\ &\quad -0.55 \le x_i \le 0.55 \ (i = 3, 4). \end{aligned}$$

The minimum value is unknown. The known best value is $f(\boldsymbol{x}) = 5126.4981$ [31].

g06 [32]:

$$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{x}) = (x_1 - 10)^3 + (x_1 - 20)^3 \\ \text{subject to} \quad & g_1(\boldsymbol{x}) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \le 0 \\ & g_2(\boldsymbol{x}) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \le 0, \\ & \quad 13 \le x_1 \le 100, \ 0 \le x_2 \le 100. \end{aligned}$$

The optimal solution $\boldsymbol{x}^* = (14.095, 0.84296)$ and the optimal value $f(\boldsymbol{x}^*) = -6961.81388$.

g07 [35]:

$$\begin{aligned} \text{minimize} \quad f(\boldsymbol{x}) &= x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 \\ &\quad + (x_3 - 10)^2 + 4(x_4 - 5)^2 \\ &\quad + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 \\ &\quad + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 \\ &\quad + (x_{10} - 7)^2 + 45 \end{aligned}$$

subject to

$$\begin{aligned} g_1(\boldsymbol{x}) &= -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 0 \\ g_2(\boldsymbol{x}) &= 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0 \\ g_3(\boldsymbol{x}) &= -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \le 0 \\ g_4(\boldsymbol{x}) &= 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \le 0 \\ g_5(\boldsymbol{x}) &= 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \le 0 \\ g_6(\boldsymbol{x}) &= x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \le 0 \\ g_7(\boldsymbol{x}) &= 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \le 0 \\ g_8(\boldsymbol{x}) &= -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \le 0, \\ &\quad -10 \le x_i \le 10 \ (i = 1, \cdots, 10). \end{aligned}$$

The optimal solution is $\boldsymbol{x}^* = (2.171996, 2.63683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927)$ and the optimal value $f(\boldsymbol{x}^*) = 24.306209$.

g08 [31]:

$$\begin{aligned} \text{maximize} \quad f(\boldsymbol{x}) &= \frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)} \\ \text{subject to} \quad g_1(\boldsymbol{x}) &= x_1^2 - x_2 + 1 \le 0 \\ g_2(\boldsymbol{x}) &= 1 - x_1 + (x_2 - 4)^2 \le 0, \\ &\quad 0 \le x_i \le 10 \ (i = 1, 2). \end{aligned}$$

The optimal solution $\boldsymbol{x}^* = (1.2279713, 4.2453733)$ and the optimal value $f(\boldsymbol{x}^*) = 0.095825$.

g09 [31]:

$$\begin{aligned} \text{minimize} \quad f(\boldsymbol{x}) &= (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 \\ &\quad + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4, \\ &\quad - 4x_6x_7 - 10x_6 - 8x_7 \end{aligned}$$

subject to

$$\begin{aligned} g_1(\boldsymbol{x}) &= -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \le 0 \\ g_2(\boldsymbol{x}) &= -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \le 0 \\ g_3(\boldsymbol{x}) &= -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \le 0 \\ g_4(\boldsymbol{x}) &= 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0, \\ &\quad -10 \le x_i \le 10 \ (i = 1, \cdots, 7). \end{aligned}$$

The optimal solution $\boldsymbol{x}^* = (2.330499, 1.951372, -0.4775414, 4.365726, -0.6244870, 1.038131, 1.594227)$ and the optimal value $f(\boldsymbol{x}^*) = 680.6300573$.

g10 [35]:

$$\text{minimize} \quad f(\boldsymbol{x}) = x_1 + x_2 + x_3$$

subject to

$$\begin{aligned} g_1(\boldsymbol{x}) &= -1 + 0.0025(x_4 + x_6) \le 0 \\ g_2(\boldsymbol{x}) &= -1 + 0.0025(x_5 + x_7 - x_4) \le 0 \\ g_3(\boldsymbol{x}) &= -1 + 0.01(x_8 - x_5) \le 0 \\ g_4(\boldsymbol{x}) &= -x_1x_6 + 833.33252x_4 + 100x_1 \\ &\quad - 83333.333 \le 0 \\ g_5(\boldsymbol{x}) &= -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \le 0 \\ g_6(\boldsymbol{x}) &= -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \le 0, \\ &\quad 100 \le x_1 \le 10000, \\ &\quad 1000 \le x_i \le 10000 \ (i = 2, 3), \\ &\quad 10 \le x_i \le 1000 \ (i = 4, \cdots, 8). \end{aligned}$$

The minimum value is unknown. The best value we found in this paper is $f(\boldsymbol{x}) = 7049.2480$.

g11 [31]:

$$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{x}) = x_1^2 + (x_2 - 1)^2 \\ \text{subject to} \quad & h(\boldsymbol{x}) = x_2 - x_1^2 = 0, \ -1 \le x_i \le 1 \ (i = 1, 2). \end{aligned}$$

The optimal solution is $\boldsymbol{x}^* = (\pm 1/\sqrt{2}, 1/2)$ and the optimal value $f(\boldsymbol{x}^*) = 0.75$.

g12 [31]:

$$\begin{aligned}\text{maximize } f(\boldsymbol{x}) &= \frac{1}{100}\left\{100 - (x_1-5)^2 - (x_2-5)^2 \right.\\ &\qquad \left. -(x_3-5)^2\right\}\\ \text{subject to } g(\boldsymbol{x}) &= (x_1-p)^2 + (x_2-q)^2 + (x_3-r)^2\\ &\quad - 0.0625 \le 0, 0 \le x_i \le 10\ (i=1,2,3),\\ &\quad p,q,r = 1,2,\cdots,9.\end{aligned}$$

The optimal solution is $\boldsymbol{x}^* = (5,5,5)$ and the optimal value $f(\boldsymbol{x}^*) = 1$.

g13 [35]:

$$\begin{aligned}\text{minimize} \quad & f(\boldsymbol{x}) = e^{x_1 x_2 x_3 x_4 x_5}\\ \text{subject to} \quad & h_1(\boldsymbol{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0\\ & h_2(\boldsymbol{x}) = x_2 x_3 - 5 x_4 x_5 = 0\\ & h_3(\boldsymbol{x}) = x_1^3 + x_2^3 + 1 = 0,\\ & \qquad - 2.3 \le x_i \le 2.3\ (i=1,2),\\ & \qquad - 3.2 \le x_i \le 3.2\ (i=3,4,5).\end{aligned}$$

The optimal solution is $\boldsymbol{x}^* = (-1.717143, 1.595709, 1.827247, -0.7636413, -0.763645)$ and the optimal value $f(\boldsymbol{x}^*) = 0.0539498$.

## REFERENCES


[1] Z. Michalewicz, "A survey of constraint handling techniques in evolutionary computation methods," in *Proc. 4th Annu. Conf. Evol. Program.*, J. R. McDonnell, R. G. Reynolds, and D. B. Fogel, Eds., 1995, pp. 135–155.

[2] Z. Michalewicz and M. Schoenauer, "Evolutionary algorithms for constrained parameter optimization problems," *Evol. Comput.*, vol. 4, no. 1, pp. 1–32, 1996.

[3] C. A. C. Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art," *Comput. Methods Appl. Mech. Eng.*, vol. 191, no. 11–12, pp. 1245–1287, Jan. 2002.

[4] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. New York: Springer-Verlag, 2003.

[5] Z. Michalewicz and G. Nazhiyath, "Genocop III: A co-evolutionary algorithm for numerical optimization with nonlinear constraints," in *Proc. 2nd IEEE Int. Conf. Evol. Comput.*, D. B. Fogel, Ed., 1995, pp. 647–651.

[6] A. I. El-Gallad, M. E. El-Hawary, and A. A. Sallam, "Swarming of intelligent particles for solving the nonlinear constrained optimization problem," *Eng. Intell. Syst. Elect. Eng. Commun.*, vol. 9, no. 3, pp. 155–163, Sep. 2001.

[7] Z. Michalewicz and N. Attia, "Evolutionary optimization of constrained problems," in *Proc. 3rd Annu. Conf. Evol. Program.*, A. Sebald and L. Fogel, Eds., 1994, pp. 98–108.

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

[9] S. Kazarlis and V. Petridis, "Varying fitness functions in genetic algorithms: Studying the rate of increase of the dynamic penalty terms," in *Lecture Notes in Computer Science*, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds. Berlin, Germany: Springer-Verlag, Sep. 1998, vol. 1498, Proc. 5th Parallel Prob. Solving from Nature (PPSN V), pp. 211–220.

[10] C. Reeves, "Genetic algorithms for the operations researcher," *INFORMS J. Comput.*, vol. 9, no. 3, pp. 231–250, 1997.

[11] R. Farmani and J. A. Wright, "Self-adaptive fitness formulation for constrained optimization," *IEEE Trans. Evol. Comput.*, vol. 7, no. 5, pp. 445–455, Oct. 2003.

[12] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 4, no. 3, pp. 284–294, Sep. 2000.

[13] T. Takahama and S. Sakai, "Tuning fuzzy control rules by $\alpha$ constrained method which solves constrained nonlinear optimization problems" (in Japanese), *Trans. Inst. Elect., Inf., Commun. Eng.*, vol. J82-A, no. 5, pp. 658–668, May 1999.

[14] ——, "Tuning fuzzy control rules by the $\alpha$ constrained method which solves constrained nonlinear optimization problems," *Elect. Commun. Jpn., Part 3: Fundamental Elect. Sci.*, vol. 83, no. 9, pp. 1–12, Sep. 2000.

[15] ——, "Learning fuzzy control rules by $\alpha$ constrained Powell's method," in *Proc. IEEE Int. Fuzzy Syst. Conf.*, vol. 2, Seoul, Korea, Aug. 1999, pp. 650–655.

[16] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, no. 2/4, pp. 311–338, 2000.

[17] E. Camponogara and S. N. Talukdar, "A genetic algorithm for constrained and multiobjective optimization," in *Proc. 3rd Nordic Workshop Genetic Algorithms Their Applicat. (3NWGA)*, J. T. Alander, Ed., Vaasa, Finland, Aug. 1997, pp. 49–62.

[18] P. D. Surry and N. J. Radcliffe, "The COMOGA method: Constrained optimization by multiobjective genetic algorithms," *Control Cybern.*, vol. 26, no. 3, pp. 391–412, 1997.

[19] T. Ray, K. Liew, and P. Saini, "An intelligent information sharing strategy within a swarm for unconstrained and constrained optimization problems," *Soft Comput.—A Fusion of Foundations, Method. Applicat.*, vol. 6, no. 1, pp. 38–44, Feb. 2002.

[20] T. P. Runarsson and X. Yao, "Evolutionary search and constraint violations," in *Proc. Congr. Evol. Comput.*, vol. 2, Dec. 2003, pp. 1414–1419.

[21] A. H. Aguirre, S. B. Rionda, C. A. C. Coello, G. L. Lizárraga, and E. M. Montes, "Handling constraints using multiobjective optimization concepts," *Int. J. Numer. Methods Eng.*, vol. 59, no. 15, pp. 1989–2017, Apr. 2004.

[22] M. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives," *Comput. J.*, vol. 7, no. 2, pp. 155–162, 1964.

[23] T. Takahama and S. Sakai, "Learning fuzzy control rules by $\alpha$-constrained simplex method" (in Japanese), *Trans. Inst. Elec., Inf., Commun. Eng.*, vol. J83-D-I, no. 7, pp. 770–779, Jul. 2000.

[24] ——, "Learning fuzzy control rules by $\alpha$-constrained simplex method," *Syst. Comput. Jpn.*, vol. 34, no. 6, pp. 80–90, Jun. 2003.

[25] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, 1965.

[26] T. Takahama and S. Sakai, "Constrained optimization by $\alpha$ constrained genetic algorithm ($\alpha$GA)" (in Japanese), *Trans. Inst. Elec., Inf., Commun. Eng.*, vol. J86-D-I, no. 4, pp. 198–207, Apr. 2003.

[27] ——, "Constrained optimization by $\alpha$ constrained genetic algorithm ($\alpha$ GA)," *Syst. Comput. Jpn.*, vol. 35, no. 5, pp. 11–22, May 2004.

[28] J. Yen and B. Lee, "A simplex genetic algorithm hybrid," in *Proc. IEEE Int. Conf. Evol. Comput.*, Indianapolis, IN, Apr. 1997, pp. 175–180.

[29] J. Yen, J. Liao, B. Lee, and D. Randolph, "A hybrid approach to modeling metabolic systems using genetic algorithms and simplex method," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 2, pp. 173–191, Apr. 1998.

[30] M. Musil, M. J. Wilmut, and N. R. Chapman, "A hybrid simplex genetic algorithm for estimating geoacoustic parameters using matched-field inversion," *IEEE J. Ocean. Eng.*, vol. 24, pp. 358–369, 1999.

[31] S. Koziel and Z. Michalewicz, "Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization," *Evol. Comput.*, vol. 7, no. 1, pp. 19–44, 1999.

[32] C. Floundas and P. Pardalos, *A Collection of Test Problems for Constrained Global Optimization*, ser. Lecture Notes in Computer Science. Berlin, Germany: Springer-Velag, 1987, vol. 455.

[33] Z. Mchalewicz, G. Nazhiyath, and M. Michalewicz, "A note on usefullness of geometrical crossover of numerical optimization problems," in *Proc. 5th Annu. Conf. Evol. Program.*, L. J. Fogel, P. J. Angeline, and T. Bäck, Eds., 1996, pp. 305–312.

[34] D. Hmmelblau, *Applied Nonlinear Programming*. New York: MacGraw-Hill, 1972.

[35] W. Hock and K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, ser. Lecture Notes in Economics and Mathematical Systems. Berlin, Germany: Springer-Verlag, 1981.

**Tetsuyuki Takahama** (M'99) received the B.Eng., M.Eng., and D.Eng. degrees in electrical engineering from Kyoto University, Kyoto, Japan, in 1982, 1984, and 1997, respectively.

He is currently a Professor in the Department of Intelligent Systems, Faculty of Information Sciences, Hiroshima City University, Hiroshima, Japan. He was a Research Associate from 1987 to 1994 and a Lecturer from 1994 to 1998 at Fukui University, Fukui, Japan. His current research interests are focused on the constrained optimization using natural computation including evolutionary computation and swarm intelligence, and the structural learning of fuzzy rules and neural networks using natural computation. His research interests include nonlinear optimization, machine learning, CAI, and natural language processing.

**Setsuko Sakai** (M'03) received the B.Ed. degree from Fukui University, Fukui, Japan, in 1979, and the M.Eng. and D.Eng. degrees in informatics and mathematical science from Osaka University, Osaka, Japan, in 1981 and 1987, respectively.

She is currently a Professor in the Department of Business Administration, Faculty of Commercial Sciences, Hiroshima Shudo University, Hiroshima, Japan. From 1986 to 1990, she was a Lecturer in Koshien University, Takarazuka, Japan. From 1990 to 1998, she was an Associate Professor at Fukui University. Her current research interests are focused on multiobjective optimization and constrained optimization using a constrained method and E constrained method. Her research interests include game theory, decision making, fuzzy mathematical programming, and CAI.