Constrained Optimization by Combining the $\alpha$ Constrained Method with Particle Swarm Optimization

Tetsuyuki Takahama  
Department of Intelligent Systems  
Hiroshima City University  
Hiroshima, 731-3194 Japan  
email:takahama@its.hiroshima-cu.ac.jp

Setsuko Sakai  
Faculty of Commercial Sciences  
Hiroshima Shudo University  
Hiroshima, 731-3195 Japan  
email:setuko@shudo-u.ac.jp

Abstract—Recently, Particle Swarm Optimization (PSO) has been applied to various application fields. In this paper, a new optimization method "$\alpha$ Constrained Particle Swarm Optimizer ($\alpha$PSO)", which is the combination of the $\alpha$ constrained method and PSO, is proposed. The $\alpha$PSO is applied to several test problems such as nonlinear programming problems and problems with non-convex constraints. It is compared with GENOCOP5.0 which is one of the famous constrained optimization systems. It is shown that the $\alpha$PSO is a very fast and good optimization algorithm for constrained optimization problems.

I. INTRODUCTION

Constrained optimization problems, especially nonlinear optimization problems, where objective functions are minimized under given constraints, are very important and frequently appear in the real world.

Simplex method is well known as an efficient method to solve linear programming problems. Also, there are some efficient methods to solve nonlinear optimization problems, such as a recursive quadratic programming, a projection method, and a generalized reduced gradient method [1]. These methods assume the differentiability of the objective function. However, it is difficult to apply these methods to problems, of which the objective function is not differentiable or the feasible set is not convex.

Generally, the constrained problems are solved by the combination of a transformation method and a direct search method. The transformation method [2] converts a constrained problem into an unconstrained one, and the direct search method optimizes the objective function by using only the value of it. There are some transformation methods such as a penalty method and a multiplier method. The penalty method is often used to solve optimization problems, because the solutions are often near the boundary of the feasible set and the method is used easily for its simplicity. However, it is difficult to know what value of the penalty coefficient leads to a feasible solution and how much a search point satisfies the constraints [3].

Recently, Particle Swarm Optimization (PSO) was proposed [4], [5] and has been applied to various application fields [6], [7]. In this paper, we propose a new constrained optimization method "$\alpha$ Constrained Particle Swarm Optimizer ($\alpha$PSO)", which is the combination of the $\alpha$ constrained method [8], [9], [10], [11] and PSO. PSO was inspired by the movement of a group of animals such as a bird flock or fish school, in which the animals avoid predators and seek foods and mates as a group (not as an individual) while maintaining proper distance between themselves and their neighbors. PSO imitates the movement to solve optimization problems and is considered as a probabilistic multi-point search method like genetic algorithms (GAs) [12]. Several approaches to the constrained optimization have been proposed using GAs [13], [14], [15], [16], [17].

In the $\alpha$ constrained method, the satisfaction level of constraints is introduced to indicate how much a search point satisfies the constraints. Also, $\alpha$ level comparison that is an order relation and gives priority to satisfaction level over the value of objective function is defined. The $\alpha$ constrained method is a special transformation method. Usually transformation methods convert a constrained optimization problem into an unconstrained problem by changing the objective function. In contrast with the methods, the $\alpha$ constrained method does not transform the objective function, but transforms an algorithm for unconstrained optimization into an algorithms for constrained optimization by replacing the ordinal comparison with the $\alpha$ level comparison in direct search methods. In [8], [9], we proposed the $\alpha$ constrained Powell’s method which is the combination of the $\alpha$ constrained method and Powell’s direct search method [18]. In [10], [11], we proposed the $\alpha$ constrained Simplex method which is the combination of the $\alpha$ constrained method and Simplex method proposed by Nelder & Mead [19]. In [13], [14], we proposed the $\alpha$ constrained Genetic Algorithm ($\alpha$GA) which is the combination of the $\alpha$ constrained method and GAs. We showed that the constrained optimization problems, of which the objective functions are not differentiable, are effectively solved by these methods. In this work, we also show that the $\alpha$ constrained method is applicable to PSO.

In $\alpha$PSO, the agent or point which satisfies the constraints will move to optimize the objective function and the agent which does not satisfy the constraints will move to satisfy the
constraints, naturally. In this paper, the effectiveness of αPSO is shown by comparing αPSO with GENOCOP 5.0 [20], [21], which is the method of developing GENOCOP III [17], [22], [23], on various types of test problems, such as linear programming problems, nonlinear programming problems, and problems with non-convex constraints.

II. CONSTRAINED OPTIMIZATION PROBLEMS

The following optimization problem (P) with the inequality constraints, the equality constraints, the upper bound constraints and the lower bound constraints will be discussed in this study:

\[
\begin{align*}
\text{(P) minimize } & \ f(x) \\
\text{subject to } & \ g_j(x) \leq 0, \ j = 1, \ldots, q \\
& \ h_j(x) = 0, \ j = q + 1, \ldots, m \\
& \ i \leq x_i \leq u_i, \ i = 1, \ldots, n,
\end{align*}
\]

where \( x = (x_1, x_2, \ldots, x_n) \) is an \( n \) dimensional vector, \( f(x) \) is an objective function, \( g_j(x) \leq 0 \) are \( q \) inequality constraints and \( h_j(x) = 0 \) are \( m - q \) equality constraints. \( f, g_j \) and \( h_j \) are linear or nonlinear real-valued functions. \( l_i \) and \( u_i \), \( i = 1, \ldots, n \) are the upper bounds and the lower bounds of \( x_i \), respectively. Also, let the feasible space in which every point satisfies all constraints be denoted by \( \mathcal{F} \) and the set in which every point satisfies the upper and lower bound constraints be denoted by \( \mathcal{S} \cap \mathcal{F} \).

III. THE α CONSTRAINED METHOD

We survey briefly the \( \alpha \) constrained method [8], [9], [10], [11], [13], [14].

A. The constrained optimization problems and the satisfaction level

In the \( \alpha \) constrained method, the satisfaction level of constraints \( \mu(x) \), which indicates how much a search point \( x \) satisfies the constraints, is introduced. The satisfaction level \( \mu(x) \) is the following function:

\[
\begin{align*}
\mu(x) = 1, & \text{ if } g_i(x) \leq 0, h_j(x) = 0 \\
& \text{ for all } i, j \\
0 \leq \mu(x) < 1, & \text{ otherwise}
\end{align*}
\]

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

\[
\text{Minimization } \mu(x) = \min_{i,j} \{ \mu_{g_i}(x), \mu_{h_j}(x) \} \tag{5}
\]

B. The \( \alpha \) level comparison

The \( \alpha \) level comparison is defined as an order relation on the set \( \{ f(x), \mu(x) \} \). 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 \( \mu(x) \) proceeds \( f(x) \), because the feasibility of \( x \) is more important than the minimization of \( f(x) \).

Let \( f_1(x_2) \) and \( \mu_1(x_2) \) be the function values and the satisfaction levels at a point \( x_1(x_2) \), respectively. Then, for any \( \alpha \) satisfying \( 0 \leq \alpha \leq 1 \), \( \alpha \) level comparison \( <_\alpha \) and \( \leq_\alpha \) between \( (f_1, \mu_1) \) and \( (f_2, \mu_2) \) is defined as follows:

\[
(f_1, \mu_1) <_\alpha (f_2, \mu_2) \Leftrightarrow \begin{cases}
\mu_1 < f_1, & \text{if } \mu_1, \mu_2 \geq \alpha \\
\mu_1 = f_1, & \text{if } \mu_1 = \mu_2 \\
\mu_1 > f_1, & \text{otherwise}
\end{cases}
\tag{6}
\]

\[
(f_1, \mu_1) \leq_\alpha (f_2, \mu_2) \Leftrightarrow \begin{cases}
\mu_1 < f_1, & \text{if } \mu_1, \mu_2 \geq \alpha \\
\mu_1 = f_1, & \text{if } \mu_1 = \mu_2 \\
\mu_1 > f_1, & \text{otherwise}
\end{cases}
\tag{7}
\]

In case of \( \alpha=0 \), \( \alpha \) level comparison \( <_0 \) and \( \leq_0 \) are equivalent to the ordinal comparison \( < \) and \( \leq \) between function values. Also, in case of \( \alpha = 1 \), \( <_1 \) and \( \leq_1 \) are equivalent to the lexicographic order in which the satisfaction level \( \mu(x) \) precedes the function value \( f(x) \).

C. The properties of the \( \alpha \) constrained method

The \( \alpha \) constrained method converts a constrained optimization problem into an unconstrained one by replacing the order relation in direct search methods with the \( \alpha \) level comparison. An optimization problem solved by the \( \alpha \) constrained method, that is, a problem in which the ordinary comparison is replaced with the \( \alpha \) level comparison, \( (P_{\leq_\alpha}) \), is defined as follows:

\[
(P_{\leq_\alpha}) \text{ minimize } \mu(x), \tag{8}
\]

where \( \mu(x) \) is defined based on the \( \alpha \) level comparison \( \leq_\alpha \). Also, a problem \( (P') \) is defined that the constraints of \( (P) \), that is, \( \mu(x) = 1 \), is relaxed and replaced with \( \mu(x) \geq \alpha \):

\[
(P') \text{ minimize } f(x) \text{ subject to } \mu(x) \geq \alpha \tag{9}
\]

It is obvious that \( (P') \) is equivalent to \( (P) \).

For the three types of problems, \( (P') \), \( (P_{\leq_\alpha}) \) and \( (P) \), the following theorems are given [8], [9].

Theorem 1: If an optimal solution \( (P') \) exists, any optimal solution of \( (P_{\leq_\alpha}) \) is an optimal solution of \( (P') \).

Theorem 2: If an optimal solution of \( (P) \) exists, any optimal solution of \( (P_{\leq_\alpha}) \) is an optimal solution of \( (P) \).
Theorem 3: Let \( \{ \alpha_n \} \) be a strictly increasing non-negative sequence and converge to 1. Let \( f(x) \) and \( \mu(x) \) be continuous functions of \( x \). Assume that an optimal solution \( x^* \) of (P1) exists and an optimal solution \( \hat{x}_n \) of \( P_{\leq n} \) exists for any \( \alpha_n \). Then, any accumulation point to the sequence \( \{ \hat{x}_n \} \) is an optimal solution of (P1).

Theorem 1 and 2 show that a constrained optimization problem can be transformed into an equivalent unconstrained optimization problem by using the \( \alpha \) level comparison. So, if the \( \alpha \) level comparison is incorporated into an existing unconstrained optimization method, constrained optimization problems can be solved. Theorem 3 shows that, in the \( \alpha \) constrained method, an optimal solution of (P1) can be given by converging \( \alpha \) to 1 as well as by increasing the penalty coefficient to infinity in the penalty method.

IV. THE \( \alpha \) CONSTRAINED PARTICLE SWARM OPTIMIZER \( \alpha \)PSO

In this section, the \( \alpha \) constrained swarm optimizer \( \alpha \)PSO is defined by combining the \( \alpha \) constrained method with the direct search method PSO.

A. Particle swarm optimization PSO

Some kinds of animals avoid predators, and seek foods and mates as a group, not as an individual or an agent. In a group of animals, such as a bird flock, a fish school and an insect swarm, they synchronously move with maintaining proper distance between themselves and their neighbors. The group often changes the direction suddenly, scatters and regroups. It is thought that the behavior of the group can be explained by some simple rules. The agents in the group share information among all members in searching food. They profit from not only the private history but also the experience of all other members. The information sharing brings great advantage into the agents when the resource is unpredictably distributed in the searching field.

PSO is the multipoint search method that is developed through the simulation of above simplified social models. For PSO is based on such a very simple concept, it can be realized by primitive mathematical operators. It computationally is very efficient, runs very fast, and requires few memories.

Searching procedures by PSO can be described as follows: A group of agents optimizes a certain objective function \( f(\cdot) \). At any time \( t \), each agent \( i \) knows its current position \( x_i^t \). It also remembers its private best value until now \( pbest_i \) and the position \( x_i^* \).

\[
pbest_i = \min_{t=1,\ldots,k} f(x_i^t) \quad \text{(10)}
\]

\[
x_i^* = \arg \min_{t=1,\ldots,k} f(x_i^t) \quad \text{(11)}
\]

Moreover, every agent knows the best value in the group until now \( gbest \) and the position \( x_G^* \).

\[
\text{gbest} = \min_i pbest_i \quad \text{(12)}
\]

\[
x_G^* = \arg \min_i f(x_i^*) \quad \text{(13)}
\]

The modified velocity \( v_i^{k+1} \) of each agent can be calculated by using the current velocity \( v_i^k \) and the difference among \( x_i^k, x_G^k \) and \( x_G^* - x_i^* \) as shown below:

\[
v_i^{k+1} = w v_i^k + c_1 rand(x_i^k - x_i^*) + c_2 rand(x_G^k - x_i^*) \quad \text{(14)}
\]

where \( rand \) is a random number in the interval \([0, 1]\).

B. Algorithm of \( \alpha \)PSO

The algorithm of \( \alpha \)PSO can be defined by replacing the ordinary comparisons with the \( \alpha \) level comparisons in PSO as follows:

1) Initializing agents

Each agent is initialized with a point \( x \) and a velocity \( v \). Each point is generated as a random point in \( S \). Each element of the velocity is generated as a random number in \([-v_{max}, v_{max}]\).

2) Finding the best agent

All agents are compared by the \( \alpha \) level comparison \( <_{\alpha} \) and the best point \( x_G^* \) is decided.

3) Updating agents

For each agent, the velocity \( v \) is updated based on the current velocity, the best point until now of the agent \( x_i^* \), and the best point until now in the group \( x_G^* \). The new point for the agent is updated by eqs. (14) and (15). If the current point of the agent is better than the best visited point of the agent, the best point is replaced by the current point. If the current point is better than the best visited point in the group, the best point is replaced by the current point.

4) Iteration

If the number of iteration is smaller than the predefined number \( T \), go back to 2. Otherwise the execution is stopped.

Following is the sample algorithm of \( \alpha \)PSO coded by C-like language based on Shi’s PSO code [24], [25].

\[
\text{\( \alpha \text{PSO}() \)
\}
\]

\[
\text{\{ \text{w=\text{w}^0} ;}
\]

\[
\text{\text{Initialize \( P(0) \) ;}}
\]

\[
\text{x_G^*=\arg\min_{x} f(x), \quad x \in \text{P(0)} ;}
\]

\[
\text{for(t=1; t \leq T; t++)}
\]

\[
\text{\{}
\]

\[
\text{\text{for(each agent i in P(t))}}
\]

\[
\text{\{
\text{w_i=\text{w}_i+\text{\phi}_1(x_i^*-x_i)+\text{\phi}_2(x_G^*-x_i) ;}
\text{v_i=\text{v}_i+\text{\alpha}_1}\text{x_i^*+\alpha_2}\text{x_G^*} ;\}
\text{\}
\}
\]

\[
\}
\]
\[ x_i = x_i + v_i; \]
\[ \text{if } f(x_i) < f(x^*_i) \}
\[ \quad x^*_i = x_i; \]
\[ \text{if } f(x_i) < f(x^*_j) \}
\[ \quad x^*_j = x_i; \]
\[ \}
\[ w = w + (w^T - w^0) / T; \]
\[ \}

where \( w \) is the weight for the current velocity and is decreased from \( w^0 \) to \( w^T \) linearly, and \( \phi_1 \) and \( \phi_2 \) are uniform random variables in \([-2, 2]\).

V. CONSTRAINED NONLINEAR PROGRAMMING PROBLEMS

In this paper, some test problems are optimized, and the results by \( \alpha \)PSO are compared with those by GENOCOP5.0.

A. Test problems and the experimental conditions

The six problems are tested. Five test problems \( G_1 - G_5 \) are given by Michalewicz [16], [23] and a test problem \( S_1 \) is given by Sakawa [3]. These test problems are shown in Table I. All problems are nonlinear constrained programming problems. All constraints of \( G_1 \) are linear functions and the objective function of \( G_2 \) is a linear function. Also, \( G_4 \) has the equality constraints and \( S_1 \) is a nonlinear problem with non-convex constraints.

The standard settings in GENOCOP 5.0 are as follows: The population size of search points and that of reference points are 70 respectively. The maximum generations \( T = 5000 \) and the parameter of nonlinear ranking selection \( c = 0.1 \). Each crossover rate for the simple crossover, the arithmetical crossover and the heuristic crossover is 4/70. Also, each mutation rate for the boundary mutation, the uniform mutation, the non-uniform mutation and the whole non-uniform mutation is 4/70.

In GENOCOP 5.0, reference points that satisfy all constraints are searched randomly. However, when the feasible region is very narrow as in \( G_4 \), any reference point cannot be found, or even if some reference points were found, it couldn’t be optimized sufficiently. Thus, based on Michalewicz’s proposal, the equality constraints are relaxed, that is, all equality constraints \( h_j(x) = 0, \ j = q + 1, \ldots, m \) are replaced by inequalities:

\[ |h_j(x)| \leq \epsilon, \ \epsilon > 0 \]  

(16)

In this experiments, \( \epsilon = 10^{-3} \). Furthermore, when any reference point cannot be found, reference points are searched additionally, by minimizing the sum of squares of the functions in the equality constraints, \( \sum_j h_j(x)^2 \), with the simplex method of Nelder & Mead.

In \( \alpha \)PSO, the same settings are used for all problems. The parameters for the \( \alpha \) constrained method are defined as follows: Every satisfaction level are defined as piecewise linear functions like eqs.(3) and (4), and set the constants \( b_i = b_j = 10000 \). The satisfaction levels are combined by minimization eq.(5). The \( \alpha \) level is fixed as \( \alpha = 1 \) for

\begin{table}[h]
\centering
\caption{TEST PROBLEMS}
\begin{tabular}{ll}
\hline
\textbf{\( G_1(x) \)} & \( 5 \sum_{i=1}^{14} x_i - 5 \sum_{i=1}^{13} x_i^2 - \sum_{i=5}^{15} x_i \), subject to \\
\hline
2(x_1 + x_2) + x_{10} + x_{11} & \leq 10, \\
2(x_1 + x_4) + x_{10} + x_{12} & \leq 10, \\
2(x_2 + x_3) + x_{11} + x_{12} & \leq 10, \\
-8x_1 + x_{10} & \leq 0, \\
-8x_2 + x_{11} & \leq 0, \\
-8x_3 + x_{12} & \leq 0, \\
-2x_4 - x_5 + x_{10} & \leq 0, \\
-2x_8 - x_9 + x_{11} & \leq 0, \\
-2x_8 + x_9 + x_{12} & \leq 0, \\
0 & \leq x_i \leq 1, i = 1, \ldots, 9, \\
0 & \leq x_{10} \leq 100, i = 10, 11, 12, \ \ 0 \leq x_{13} \leq 1.
\hline
\textbf{\( G_2(x) \)} & \( x_1 + x_2 + x_3 \), subject to \\
\hline
1 & -0.0025(x_4 + x_6) \geq 0, \\
1 & -0.0025(x_5 + x_7 - x_4) \geq 0, \\
1 & -0.01(x_9 - x_5) \geq 0, \\
x_1 + x_6 - 833.3252x_4 - 100x_1 + 8333.333x_3 \geq 0, \\
x_2x_7 - 1250x_5 - x_2x_4 + 1250x_3 \geq 0, \\
x_1x_6 - 1250000 - x_2(x_5 - 2500x_3) \geq 0, \\
100 & \leq x_1 \leq 10000, \ \ \ 1000 & \leq x_2 \leq 10000, i = 2, 3, \\
10 & \leq x_i \leq 1000, i = 4, \ldots, 8.
\hline
\textbf{\( G_3(x) \)} & \( (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^2 + 3(x_4 - 11)^2 \\
& + 10x_5^2 + 7x_6^2 + x_7^2 - 42x_4x_7 - 10x_6 - 8x_7, \\
\hline
\textbf{\( G_4(x) \)} & \( e^{x_1^2 + x_2^2 + x_3^2} \\
\hline
\textbf{\( G_5(x) \)} & \( x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 = 10, \\
x_2x_3 - 5x_4x_5 = 0, \\
x_1^2 + x_2^2 = 1 - 2.5 \leq x_i \leq 3.1, \ -3.2 \leq x_i \leq 3.2, i = 3, 4, 5. \\
\textbf{\( G_6(x) \)} & \( x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 = 10, \\
x_2x_3 - 5x_4x_5 = 0, \\
x_1^2 + x_2^2 = 1 - 2.5 \leq x_i \leq 3.1, \ -3.2 \leq x_i \leq 3.2, i = 3, 4, 5. \\
\textbf{\( G_7(x) \)} & \( x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 = 10, \\
x_2x_3 - 5x_4x_5 = 0, \\
x_1^2 + x_2^2 = 1 - 2.5 \leq x_i \leq 3.1, \ -3.2 \leq x_i \leq 3.2, i = 3, 4, 5. \\
\textbf{\( S_1(x) \)} & \( x_1^2 + (x_2 - 5)^2 + 3(x_3 - 9)^2 - 12x_3 + 2x_4^2 \\
& + 4x_5^2 + (x_6 - 5)^2 - 6x_2^2 + 3(x_7 - 2)x_8^2 \\
& - x_9x_{10} + 4x_5^3 + 5x_1x_3 - 3x_1x_7 + 2x_4x_7, \\
\hline
\end{tabular}
\end{table}
all problems. Also, an equality constraint isn’t transformed into eq.(16), but eq.(4) is used as the satisfaction level. The parameters for PSO are as follows: The number of agents $N = 70$, $v_{max} = 2$, $w^0 = 1.0$, $w^T = 0.0$. The maximum number of iterations is $T = 5000$.

In works of Michalewicz, 10 trials were tested for each problem. Since good results might continue in 10 trials by chance, 100 trials are tested in order to evaluate more exactly in this paper.

**B. Experimental results**

Experimental results on the test problems are shown in Table II, in which each value is the average of 100 trials. "Optimal" is the optimal value in each problem. Also, "best", "average", "worst", and "σ" are the best value, the average value, the worst value, and the standard deviation of the objective function’s values for the best agent in each trial, respectively. CPU(s) is the average execution time (seconds) for a trial using a computer with UltraSPARCIII (750MHz). #feasible shows the number of trials when feasible solutions are found. "const.err" shows how far the best agent is away from the feasible region, that is, $\max_{i,j} \{0, g_i(x), h_j(x)\}$.

In all problems, αPSO found the better solutions than GENOCOP5.0 on average. In $G_1$, αPSO and GENOCOP5.0 found the optimal solution. In $G_2, G_3, G_4$ and $S_1$, the best values found by αPSO are smaller than those by GENOCOP5.0, or αPSO found better solutions than GENOCOP5.0. Especially in $S_1$, αPSO found the solutions better than the solution that was found by Sakawa.

In all problems, αPSO could find feasible solutions. In $G_1$, αPSO found the feasible solutions 86 times, but GENOCOP5.0 cannot find any feasible solution of $G_4$. The solutions obtained by GENOCOP5.0 are away about $3 \times 10^{-4} \sim 1 \times 10^{-3}$ from the feasible region because the constraints are relaxed. Thus, it is thought that the ability of αPSO searching feasible solutions is higher for the problems with the equality constraints.

As for the execution time, αPSO is more than 10 times faster than GENOCOP5.0. Therefore, it is thought that αPSO is a very faster optimization algorithm than GENOCOP5.0.

Table III shows $\rho = |F \cap S|/|S|$, which is the ratio between the size of the feasible space and the whole search space. The ratio $\rho$ for each problem is obtained by generating 10,000,000 random points within $S$ and counting the points within $F$. From the table, the feasible region of $G_3$ is very wide and the feasible region of $G_5$ is very narrow. In both of the problems with wide and narrow feasible region, the considerably better solutions were found by αPSO than by GENOCOP5.0. Thus, αPSO is a method that the solution with high precision can be obtained even if the problem has the wide and narrow feasible region.

**VI. CONCLUSIONS**

We proposed αPSO that combined the α constrained method with PSO. By applying αPSO to the six constrained optimization problems, it was shown that αPSO obtained an approximate solution near the optimal one for every problem by the numerical experiments and αPSO was a high precision and stable optimization algorithm. Also, by comparing αPSO with GENOCOP5.0 which is known as an efficient algorithm for the constrained optimization problems, it was shown that αPSO was a very fast and good algorithm.

In this paper, only simple version of PSO is used. In the future, we are planning to introduce some operators such as boundary mutation in order to improve the performance of αPSO. Also, we will apply αPSO to various application fields.

**Acknowledgement**

This research is supported in part by Grant-in-Aid for Scientific Research (C)(2)(No. 14580498, 16500083) of Japan society for the promotion of science.
REFERENCES


