2006 IEEE Congress on Evolutionary Computation
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Constrained Optimization by the ε Constrained Differential Evolution with Gradient-Based Mutation and Feasible Elites

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

***Abstract*— While research on constrained optimization using evolutionary algorithms has been actively pursued, it has had to face the problem that the ability to solve multi-modal problems, which have many local solutions within a feasible region, is insufficient, that the ability to solve problems with equality constraints is inadequate, and that the stability and efficiency of searches is low. We proposed the εDE, defined by applying the ε constrained method to a differential evolution (DE). DE is a simple, fast and stable population based search algorithm that is robust to multi-modal problems. The εDE is improved to solve problems with many equality constraints by introducing a gradient-based mutation that finds feasible point using the gradient of constraints at an infeasible point. Also the εDE is improved to find feasible solutions faster by introducing elitism where more feasible points are preserved as feasible elites. The improved εDE realizes stable and efficient searches that can solve multi-modal problems and those with equality constraints. The advantage of the εDE is shown by applying it to twenty four constrained problems of various types.**

I. INTRODUCTION

There are many studies on solving constrained optimization problems using evolutionary algorithms (EAs) [1], [2]. However these studies have had to address some problems:

1) The ability to solve multi-modal problems is insufficient. When multi-modal problems that have many local solutions in a feasible region are solved, even if the EAs can locate the feasible region, they are sometimes trapped to a local solution and cannot search for an optimal solution.
2) Obtained solutions in problems with equality constraints are inadequate. Many EAs for constrained optimization solve problems with equality constraints by converting equality constraints into relaxed inequality constraints. As a result, the feasibility of the obtained solutions is inadequate. Also, it is difficult for them to solve problems with many equality constraints.
3) The stability and efficiency of searches is low. Sometimes EAs cannot overcome the effect of randomness in the search process of some problems. Thus, the stability of the search becomes low. Also, many EAs need rank-based selection or replacement, stochastic selection and mutations based on Gaussian or Cauchy distributions that incur high computational costs.

To overcome these problems, we propose the εDE, defined by applying the ε constrained method [3] to a differential evolution (DE) [4]. By incorporating DE, problems (1) and (3) can be solved; DE is a simple, fast and stable search algorithm that is robust to multi-modal problems. The εDE is stable because it uses a simple and stable selection and replacement mechanism excluding stochastic operations. The εDE is also efficient because it uses a simple arithmetic operation and does not use any rank-based operations or mutations based on Gaussian and Cauchy distributions. The problem (2) is solved by using a simple way of controlling the relaxation of equality constraints for the ε constrained method to directly solve problems with equality constraints. But it is difficult for the εDE to solve problems with many equality constraints. The εDE is improved to solve the problems by introducing a gradient-based mutation that finds a feasible point from an infeasible point using the gradient of constraints at the infeasible point. Also, it is difficult for the εDE to find feasible points in earlier generation, because it finds infeasible points with better objective values until the relaxation of equality constraints is reduced completely or becomes zero. The εDE is improved to find feasible solutions faster by introducing elitism where more feasible points are preserved as feasible elites.

The improved εDE realizes stable and efficient searches that can solve multi-modal problems and those with many equality constraints. The advantage of the εDE is shown by applying it to twenty four constrained problems of various types.

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).

II. PREVIOUS WORKS

EAs for constrained optimization can be classified into several categories by the way the constraints are treated:

1) Constraints are only used to see whether a search point is feasible or not [5], [6].
2) The constraint violation, which is the sum of the violation of all constraint functions, is combined with the objective function. Approaches in this category are called penalty function methods.
3) The constraint violation and the objective function are used separately and are optimized separately [7]–[10]. For example, Takahama and Sakai proposed the α constrained method [11], [12] and the ε constrained method [3]. These methods adopt a lexicographic order, in which the constraint violation precedes the objective function, with relaxation of the constraints. The methods can effectively optimize problems with equality constraints through the relaxation of the constraints. Deb [7] proposed a method using an extended

0-7803-9487-9/06/$20.00/©2006 IEEE

objective function that realizes the lexicographic ordering. Runarsson and Yao [8] proposed a stochastic ranking method based on ES and using a stochastic lexicographic order that ignores constraint violations with some probability. These methods have been successfully applied to various problems.

4) The constraints and the objective function are optimized by multiobjective optimization methods [13]–[16].

In category 1), 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. In category 2), it is difficult to select an appropriate value for the penalty coefficient that adjusts the strength of the penalty. Several methods that dynamically control the penalty coefficient have been proposed [17]–[19]. However, ideal control of the coefficient is problem dependent [20] and it is difficult to determine a general control scheme. In category 4), the difficulty is that solving multiobjective optimization problems is a more difficult and expensive task than solving single objective optimization problems.

In this paper, we investigate the ε constrained method in the promising category 3). The α and the ε constrained methods are new types of transformation methods that convert an algorithm for unconstrained optimization into an algorithm for constrained optimization by replacing the ordinal comparisons with the α and the ε level comparisons in direct search methods. We call the methods *algorithm transformation methods*. The α constrained method was applied to Powell's method, the nonlinear simplex method by Nelder and Mead, a genetic algorithm and particle swarm optimization (PSO); and the α constrained Powell's method [11], [12], the α constrained simplex method [21], [22], the αGA [23] and the αPSO [24] were proposed. The ε constrained method was applied to PSO, and the ε constrained PSO [3] and hybrid algorithm [25] were proposed.

Of these methods, the α constrained simplex method with mutations [22] is the best as it can stably and efficiently search solutions. However, the method is often trapped by local solutions in multi-modal problems and an optimal solution cannot be found. Contrarily, the εDE can stably solve multi-modal problems.

III. The ε constrained method

In this section, the ε constrained method [3] is described.

A. Constrained optimization problems

The general constrained optimization problem (P) with inequality, equality, upper bound and lower bound constraints is defined as follows:

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

where $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ 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 and lower bounds of x_i, respectively. The upper and lower bounds define the *search space* $\mathcal{S}$. Inequality and equality constraints define the *feasible region* $\mathcal{F}$. Feasible solutions exist in $\mathcal{F} \subseteq \mathcal{S}$.

B. The constraint violation

We introduce the constraint violation $\phi(\boldsymbol{x})$ to indicate by how much a search point $\boldsymbol{x}$ violates the constraints. The constraint violation $\phi(\boldsymbol{x})$ is the following function:

$$\begin{cases} \phi(\boldsymbol{x}) = 0 \ (\boldsymbol{x} \in \mathcal{F}) \\ \phi(\boldsymbol{x}) > 0 \ (\boldsymbol{x} \notin \mathcal{F}) \end{cases} \tag{2}$$

Some types of constraint violations, which are adopted as a penalty in penalty function methods, can be defined as follows:

$$\phi(\boldsymbol{x}) = \max\{\max_j\{0, g_j(\boldsymbol{x})\}, \max_j |h_j(\boldsymbol{x})|\} \tag{3}$$

$$\phi(\boldsymbol{x}) = \sum_j max\{0, g_j(\boldsymbol{x})\}^p + \sum_j |h_j(\boldsymbol{x})|^p \tag{4}$$

where p is a positive number. In this paper, the simple sum of constraints, $p = 1$, is used.

C. The ε level comparison

The ε *level comparison* is defined as an order relation on the set of $(f(\boldsymbol{x}), \phi(\boldsymbol{x}))$. The ε level comparisons are defined by a lexicographic order in which $\phi(\boldsymbol{x})$ 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 $\phi_1\,(\phi_2)$ be the function value and the constraint violation respectively, at a point $\boldsymbol{x}_1\,(\boldsymbol{x}_2)$. Then, for any ε satisfying $\varepsilon \ge 0$, the ε level comparisons $<_\varepsilon$ and $\le_\varepsilon$ between (f_1, ϕ_1) and (f_2, ϕ_2) are defined as follows:

$$(f_1, \phi_1) <_\varepsilon (f_2, \phi_2) \quad \Leftrightarrow \quad \begin{cases} f_1 < f_2, & \text{if } \phi_1, \phi_2 \le \varepsilon \\ f_1 < f_2, & \text{if } \phi_1 = \phi_2 \\ \phi_1 < \phi_2, & \text{otherwise} \end{cases} \tag{5}$$

$$(f_1, \phi_1) \le_\varepsilon (f_2, \phi_2) \quad \Leftrightarrow \quad \begin{cases} f_1 \le f_2, & \text{if } \phi_1, \phi_2 \le \varepsilon \\ f_1 \le f_2, & \text{if } \phi_1 = \phi_2 \\ \phi_1 \le \phi_2, & \text{otherwise} \end{cases} \tag{6}$$

In the case of $\varepsilon = 0$, $<_0$ and $\le_0$ are equivalent to the lexicographic order in which the constraint violation $\phi(\boldsymbol{x})$ precedes the function value $f(\boldsymbol{x})$. Also, in the case of $\varepsilon = \infty$, the ε level comparisons $<_\infty$ and $\le_\infty$ are equivalent to the ordinal comparisons $<$ and $\le$ between function values.

D. The properties of the ε constrained method

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

$$(\text{P}_{\le_\varepsilon}) \quad \text{minimize}_{\le_\varepsilon} \quad f(\boldsymbol{x}), \tag{7}$$

where minimize$_{\leq_\varepsilon}$ denotes the minimization based on the ε level comparison $\leq_\varepsilon$. Also, a problem (P^ε) is defined such that the constraint of (P), that is, $\phi(\boldsymbol{x}) = 0$, is relaxed and replaced with $\phi(\boldsymbol{x}) \leq \varepsilon$:

$$(\mathrm{P}^\varepsilon) \quad \begin{array}{ll} \text{minimize} & f(\boldsymbol{x}), \\ \text{subject to} & \phi(\boldsymbol{x}) \leq \varepsilon. \end{array} \tag{8}$$

It was shown that a constrained optimization problem (P^0) can be transformed into an equivalent unconstrained optimization problem ($\mathrm{P}_{\leq_0}$) using ε level comparisons. So, if ε level comparisons are incorporated into an existing unconstrained optimization method, constrained optimization problems can be solved. It is thought that the ε constrained method converts an algorithm for unconstrained optimization into an algorithm for constrained optimization by replacing ordinary comparisons with ε level comparisons. Also, it was shown that, in the ε constrained method, an optimal solution of (P^0) can be obtained by converging ε to 0, in a fashion similar to increasing the penalty coefficient to infinity in the penalty function method.

IV. The improved εDE

A. Differential Evolution

Differential evolution is a variant of ES proposed by Storn and Price [4]. DE is a stochastic direct search method using population or multiple search points. DE has been successfully applied to the optimization problems including non-linear, non-differentiable, non-convex and multi-modal functions. It has been shown that DE is fast and robust to these functions.

In DE, initial individuals are randomly generated within the search space and form an initial population. Each individual contains n genes as decision variables or a decision vector. At each generation or iteration, all individuals are selected as parents. Each parent is processed as follows: The mutation process begins by choosing $1 + 2\,num$ individuals from the parents except for the parent in the processing. The first individual is a base vector. All subsequent individuals are paired to create num difference vectors. The difference vectors are scaled by the scaling factor F and added to the base vector. The resulting vector is then mated or recombined with the parent. The probability of recombination at an element is controlled by the crossover factor CR. This crossover process produces a trial vector. Finally, for survivor selection, the trial vector is accepted for the next generation if the trial vector is better than the parent.

In this study, DE/rand/1/exp variant, where exponential crossover is adopted and the number of difference vector is 1 or $num = 1$, is used. Other variant DE/rand/1/bin, where binomial crossover is adopted, has been studied well. However, it seems that exponential crossover can optimize more types of problems using a same value of crossover factor than binomial crossover in our experience and we decided to adopt exponential crossover.

B. The εDE with gradient-based mutation and feasible elites

The εDE is an algorithm where the ε constrained method is applied into DE, or ordinary comparisons in DE are replaced with the ε level comparisons.

The gradient-based mutation is an operation similar to the gradient-based repair method proposed by Chootinan and Chen [26]. The vector of constraint functions $C(\boldsymbol{x})$, the vector of constraint violations $\Delta C(\boldsymbol{x})$ and the increment of a point $\boldsymbol{x}$ to satisfy constraints $\Delta\boldsymbol{x}$ are defined as follows:

$$\nabla C(\boldsymbol{x})\Delta\boldsymbol{x} = -\Delta C(\boldsymbol{x}) \tag{9}$$

$$\Delta\boldsymbol{x} = -\nabla C(\boldsymbol{x})^{-1}\Delta C(\boldsymbol{x}) \tag{10}$$

where $C(\boldsymbol{x}) = (g_1(\boldsymbol{x}) \cdots g_q(\boldsymbol{x})\, h_{q+1}(\boldsymbol{x}) \cdots h_m(\boldsymbol{x}))^T$, $\Delta C(\boldsymbol{x})$=$(\Delta g_1(\boldsymbol{x}) \cdots \Delta g_q(\boldsymbol{x}) h_{q+1}(\boldsymbol{x}) \cdots h_m(\boldsymbol{x}))^T$, $\Delta g_j(\boldsymbol{x})$ = $\max\{0, g_j(\boldsymbol{x})\}$. Although the gradient matrix ∇C is not invertible in general, the Moore-Penrose inverse or pseudoinverse[1] $\nabla C(\boldsymbol{x})^+$ [27], which gives an approximate or best (least squares) solution to a system of linear equations, can be used instead in Eq. (10).

This mutation or repair operation, $\boldsymbol{x}^{\text{new}} = \boldsymbol{x} + \Delta\boldsymbol{x}$, is executed with gradient-based mutation probability P_g. In [26], only non-zero elements of $\Delta C(\boldsymbol{x})$ are repaired and the repair operation is repeated with some probability while amount of repair is not small. In this study, however, non-zero inequality constraints and all equality constraints are considered to keep the feasibility of equality constraints. The mutation operation is repeated fixed times R_g while the point is infeasible.

In feasible elites preserving strategy, N_e most feasible points, which have minimum constraint violation, are preserved as elites in the initial generation. At each generation, a base vector and difference vectors are selected from not only a population but also feasible elites. When a trial vector is generated, it is compared with feasible elites. If the trial vector has smaller constraint violation than the worst feasible elite, the elite is replaced with the vector. This strategy is applied only when the ε level is not zero. When the ε level is zero, more feasible individuals can survive naturally and it does not need to use elites preserving strategy.

C. Controlling the ε level

Usually, the ε level does not need to be controlled. Many constrained problems can be solved based on the lexicographic order where the ε level is constantly 0. However for problems with equality constraints, the ε level should be controlled properly to obtain high quality solutions.

In this study, a simple way of controlling the ε level is defined according to the equation (11). The ε level is updated until the number of iterations t becomes the control generation T_c. After the number of iterations exceeds T_c, the ε level is set to 0 to obtain solutions with minimum constraint

[1] Pseudoinverse of a matrix A can be obtained using the singular value decomposition; A=$U\Sigma V^T$ and $A^+ = V\Sigma^+U^T$, where Σ^+ is a matrix given by inverting each non-zero element on the diagonal of Σ.

violation.

$$\varepsilon(0) = \phi(\boldsymbol{x}_\theta) \tag{11}$$
$$\varepsilon(t) = \begin{cases} \varepsilon(0)(1 - \frac{t}{T_c})^{cp}, & 0 < t < T_c, \\ 0, & t \geq T_c \end{cases}$$

where $\boldsymbol{x}_\theta$ is the top θ-th individual and $\theta = 0.2N$, and cp is a parameter to control the speed of reducing relaxation of constraints.

D. The algorithm of the improved εDE

The algorithm of the improved εDE based on DE/rand/1/exp variant, which is used in this study, is as follows:

Step0 Initialization. Initial N individuals $\boldsymbol{x}^i$ are generated as the initial search points, where there is an initial population $P(0) = \{\boldsymbol{x}^i, i = 1, 2, \cdots, N\}$. An initial ε level is given by the ε level control function $\varepsilon(0)$. If $\varepsilon(0)$ is not zero, the N_e most feasible points, which have minimum constraint violation, are preserved as elites $P_e(0)(\subset P(0))$.

Step1 Termination condition. If the number of generations (iterations) exceeds the maximum generation $T_{\max}$, the algorithm is terminated.

Step2 Mutation. For each individual $\boldsymbol{x}^i$ in $P(t)$, three different individuals $\boldsymbol{x}^{p1}$, $\boldsymbol{x}^{p2}$ and $\boldsymbol{x}^{p3}$ are chosen from population $P(t) \cup P_e(t)$ without overlapping $\boldsymbol{x}^i$. A new vector $\boldsymbol{x}'$ is generated by the base vector $\boldsymbol{x}^{p1}$ and the difference vector $\boldsymbol{x}^{p2} - \boldsymbol{x}^{p3}$ as follows:

$$\boldsymbol{x}' = \boldsymbol{x}^{p1} + F(\boldsymbol{x}^{p2} - \boldsymbol{x}^{p3}) \tag{12}$$

where F is a scaling factor.

Step3 Crossover. The vector $\boldsymbol{x}'$ is recombined with the parent $\boldsymbol{x}^i$. A crossover point j is chosen randomly from all dimensions $[1, n]$. The element at the j-th dimension of the trial vector $\boldsymbol{x}^{\text{new}}$ is inherited from the j-th element of the vector $\boldsymbol{x}'$. The elements of subsequent dimensions are inherited from $\boldsymbol{x}'$ with exponentially decreasing probability defined by a crossover factor CR. In real processing, Step2 and Step3 are integrated as one operation.

Step4 Gradient-based mutation. If $\boldsymbol{x}^i$ is not ε-feasible, or $\phi(\boldsymbol{x}^i) > \varepsilon(t)$, $\boldsymbol{x}^{\text{new}}$ is changed by the gradient-based mutation with probability P_g. In this case, $\boldsymbol{x}^{\text{new}}$ is replaced by $\boldsymbol{x}^{\text{new}} - \nabla C(\boldsymbol{x}^{\text{new}})^{+} \Delta C(\boldsymbol{x}^{\text{new}})$ until the number of the replacement becomes R_g or $\boldsymbol{x}^{\text{new}}$ becomes ε-feasible, or $\phi(\boldsymbol{x}^{\text{new}}) \leq \varepsilon(t)$.

Step5 Survivor selection. The trial vector $\boldsymbol{x}^{\text{new}}$ is accepted for the next generation if the trial vector is better than the parent $\boldsymbol{x}^i$. Also, if the trial vector has smaller constraint violation than the least feasible elite in $P_e(t)$, the elite is replaced with the vector.

Step6 Controlling the ε level. The ε level is updated by the ε level control function $\varepsilon(t)$. If $\varepsilon(t)$ is zero, the population for elites is made empty, $P_e(t) = \emptyset$.

Step7 Go back to Step1.

V. Experimental Results

Twenty four benchmark problems and format for reporting results are specified in “Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization” [28].

A. PC Configure

The ε constrained differential evolution with gradient-based mutation and feasible elites was executed on the following system.

System: Cygwin on Dynabook SS3500, CPU: Mobile Pentium III 1.33GHz, RAM: 240MB, Language: C.

B. Parameters Setting

a) All parameters to be adjusted.
Parameters for DE are population size (N), scaling factor (F) and crossover factor (CR). Parameters for ε constrained method are control generations (T_c) and control factor (cp). The others are gradient-based mutation rate (P_g), the number of repeating mutation (R_g) and the number of feasible elites (N_e).

b) Corresponding dynamic ranges.
Dynamic ranges are not studied enough, but we recommend the following ranges: $N \in [2n, 20n]$, $F \in [0.6, 0.8]$, $CR \in [0.6, 0.95]$, $T_c \in [0.1T_{max}, 0.8T_{max}]$, $cp \in [2, 10]$, $P_g \in [0.01, 0.1]$, $R_g \in [1, 5]$, $N_e \in [1, 0.2N]$.

c) Guidelines on how to adjust the parameters.
Higher N, higher F, higher CR, higher T_c, lower cp, higher P_g, higher R_g and lower N_e make search process more robust, but less fast.

d) Estimated cost of parameter tuning in terms of number of function evaluations (FES).
The cost of parameter tuning is not high, because the following settings works well in many problems.

e) Actual parameter values used.
$N = 40$, $F = 0.7$, $CR = 0.9$, $T_c = 0.2T_{max} = 2500$ ($T_{max} = 12500$), $cp = 5$, $P_g = 0.01$, $R_g = 3$ and $N_e = 3$.

C. Results Achieved

Tables I, II, III and IV show best, worst, median, mean, and standard deviation of the difference between the best value found f^{best} and the optimal value f^*, or $f^{\text{best}} - f^*$, after 5×10^3 , 5×10^4 and 5×10^5 FES for independent 25 runs. Numbers in parenthesis after objective function values show the corresponding number of violated constraints. Number of constraints, which are infeasible at a median solution by more than 1, 0.01, and 0.0001, are shown in c, respectively. Mean violation for the median solution is shown in $\bar{v}$. Table V shows the number of FES needed for satisfying the success condition of $f^{\text{best}} - f^* \leq 0.0001$ and $\boldsymbol{x}^{\text{best}}$ being feasible. The ratio of runs where feasible solutions or successful solutions can be found, and the estimated FES to find successful solutions for the problems are shown, too. In the εDE, the objective function and the constraint violation are treated separately, and the evaluation of the objective function can often be omitted when the ε-level comparison can be calculated only by the constraint violation. Numbers in parenthesis after the estimated FES show the substantial number of evaluations for objective function and the number of evaluations for gradient matrix to find successful solutions, respectively.

Tables I–IV show that the εDE succeeded to find feasible and (near) optimal solutions for all problems in all runs except for g20 and g22. For g22, the εDE succeeded to find feasible solutions in all runs and find better objective values than old best known objective value 382.902205 in all runs. Although the εDE could not find the new best known solution, the εDE found the objective values less than 240 at 4 times and less than 250 at 10 times. Also, the new best known objective value 236.430975504001054 in [28] was found by the εDE. For g20, the εDE could not find any feasible solutions. However, the εDE improved the mean constraint violation ($\bar{v}$) of the old best known solution (0.154198) and found the more feasible solution (0.00718768), which appears in [28]. So, the εDE succeeded in finding more feasible solutions.

Figures 1, 2, 3 and 4 show the graphs of $f^{\text{best}} - f^*$ and $\bar{v}$ over FES at the median run. Table V and Figures 1–4 show that the εDE could find near optimal solutions very fast; Success performance is less than 5,000 for 3 problems, is less than 50,000 for 9 problems, is less than 100,000 for 16 problems and is less than 150,000 for 20 problems. Especially, the substantial number of evaluations for objective function is less than 5,000 for 5 problems, is less than 50,000 for 21 problems. Also, although the εDE needs the calculation of gradient matrix, the number of evaluations for the matrix is fairly small.

Table VI shows the time of 10000 evaluations for all functions ($T1$), the time of algorithm for 10000 FES ($T2$) and their ratio. The value $(T2 - T1)/T1$ shows that the algorithm of εDE is very efficient and the evaluation of the gradient matrix is not time-consuming.

TABLE VI

COMPUTATIONAL COMPLEXITY

$T1$	$T2$	$(T2 - T1)/T1$
0.7554	1.1093	0.4685

VI. CONCLUSIONS

Differential evolution is a recently proposed variant of an evolutionary strategy. In this study, we proposed the εDE by applying the ε constrained method to DE and showed that the εDE can solve constrained optimization problems. Also, to solve problems with many equality constraints faster, which are very difficult problems for numerical optimization, we proposed gradient-based mutation and feasible elites preserving strategy. We showed that the εDE could solve 23 problems out of 24 benchmark problems very fast.

In the future, we will apply the εDE to various real world problems that have large numbers of decision variables and constraints.

ACKNOWLEDGMENTS

This research is supported in part by Grant-in-Aid for Scientific Research (C) (No. 16500083, 17510139) of Japan society for the promotion of science.

REFERENCES

[1] Z. Michalewicz, “A survey of constraint handling techniques in evolutionary computation methods,” in *Proceedings of the 4th Annual Conference on Evolutionary Programming*. Cambridge, Massachusetts: The MIT Press, 1995, pp. 135–155.

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

[3] T. Takahama and S. Sakai, “Constrained optimization by ε constrained particle swarm optimizer with ε-level control,” in *Proc. of the 4th IEEE International Workshop on Soft Computing as Transdisciplinary Science and Technology (WSTST’05)*, Muroran, Japan, May 2005, pp. 1019–1029.

[4] R. Storn and K. Price, “Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[5] Z. Michalewicz and G. Nazhiyath, “Genocop III: A co-evolutionary algorithm for numerical optimization with nonlinear constraints,” in *Proceedings of the Second IEEE International Conference on Evolutionary Computation*. IEEE Press, 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,” *Engineering Intelligent Systems for Electrical Engineering and Communications*, vol. 9, no. 3, pp. 155–163, Sept. 2001.

[7] K. Deb, “An efficient constraint handling method for genetic algorithms,” *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2/4, pp. 311–338, 2000.

[8] T. P. Runarsson and X. Yao, “Stochastic ranking for constrained evolutionary optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, Sept. 2000.

[9] E. Mezura-Montes and C. A. C. Coello, “A simple multimembered evolution strategy to solve constrained optimization problems,” *IEEE Trans. on Evolutionary Computation*, vol. 9, no. 1, pp. 1–17, Feb. 2005.

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

[11] T. Takahama and S. Sakai, “Learning fuzzy control rules by α constrained powell’s method,” in *Proceedings of 1999 IEEE International Fuzzy Systems Conference*, vol. 2, Seoul, Korea, Aug. 1999, pp. 650–655.

[12] ——, “Tuning fuzzy control rules by the α constrained method which solves constrained nonlinear optimization problems,” *Electronics and Communications in Japan, Part3: Fundamental Electronic Science*, vol. 83, no. 9, pp. 1–12, Sept. 2000.

[13] E. Camponogara and S. N. Talukdar, “A genetic algorithm for constrained and multiobjective optimization,” in *3rd Nordic Workshop on Genetic Algorithms and Their Applications (3NWGA)*, J. T. Alander, Ed. Vaasa, Finland: University of Vaasa, Aug. 1997, pp. 49–62.

[14] P. D. Surry and N. J. Radcliffe, “The COMOGA method: Constrained optimisation by multiobjective genetic algorithms,” *Control and Cybernetics*, vol. 26, no. 3, pp. 391–412, 1997.

[15] T. P. Runarsson and X. Yao, “Evolutionary search and constraint violations,” in *Proceedings of the 2003 Congress on Evolutionary Computation*, vol. 2. Piscataway, New Jersey: IEEE Service Center, Dec. 2003, pp. 1414–1419.

[16] A. H. Aguirre, S. B. Rionda, C. A. C. Coello, G. L. Lizárraga, and E. M. Montes, “Handling constraints using multiobjective optimization concepts,” *International Journal for Numerical Methods in Engineering*, vol. 59, no. 15, pp. 1989–2017, Apr. 2004.

[17] J. Joines and C. Houck, “On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs,” in *Proceedings of the first IEEE Conference on Evolutionary Computation*, D. Fogel, Ed. Orlando, Florida: IEEE Press, 1994, pp. 579–584.

TABLE I

ERROR VALUES ACHIEVED WHEN FES= 5×10^3 , FES= 5×10^4, FES= 5×10^5 FOR PROBLEMS 1-6.

FES		g01	g02	g03	g04	g05	g06
5×10^3	Best	9.1694e+00(0)	4.3267e-01(0)	2.2441e-01(0)	2.4820e+01(0)	4.0399e-02(0)	1.2219e-02(0)
	Median	1.0488e+01(0)	5.0111e-01(0)	9.6156e-01(0)	5.7311e+01(0)	1.2674e+00(0)	1.2382e-01(0)
	Worst	1.3370e+01(0)	5.5870e-01(0)	1.0005e+00(0)	1.1140e+02(0)	2.5976e+01(0)	2.7061e+00(0)
	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0	0	0	0	0	0
	Mean	1.0536e+01	4.9947e-01	8.4655e-01	5.9547e+01	4.7516e+00	3.9619e-01
	Std	1.0087e+00	3.1941e-02	2.1969e-01	2.2260e+01	7.3865e+00	6.3579e-01
5×10^4	Best	3.8990e-04(0)	2.4480e-02(0)	1.3997e-02(0)	1.0914e-11(0)	8.1313e-04(0)	1.1823e-11(0)
	Median	7.7833e-04(0)	7.1277e-02(0)	5.1722e-02(0)	3.6380e-11(0)	1.1755e-02(0)	1.1823e-11(0)
	Worst	1.4376e-03(0)	1.2255e-01(0)	1.1363e-01(0)	1.4188e-10(0)	1.7621e-01(0)	1.1823e-11(0)
	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0	0	0	0	0	0
	Mean	8.4426e-04	6.9361e-02	5.4388e-02	4.9185e-11	2.2166e-02	1.1823e-11
	Std	2.5185e-04	2.7681e-02	2.2945e-02	3.4020e-11	3.4215e-02	0.0000e+00
5×10^5	Best	0.0000e+00(0)	4.0394e-09(0)	-4.4409e-16(0)	0.0000e+00(0)	0.0000e+00(0)	1.1823e-11(0)
	Median	0.0000e+00(0)	3.0933e-08(0)	-4.4409e-16(0)	0.0000e+00(0)	0.0000e+00(0)	1.1823e-11(0)
	Worst	0.0000e+00(0)	7.3163e-08(0)	-4.4409e-16(0)	0.0000e+00(0)	0.0000e+00(0)	1.1823e-11(0)
	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0	0	0	0	0	0
	Mean	0.0000e+00	3.0333e-08	-4.4409e-16	0.0000e+00	0.0000e+00	1.1823e-11
	Std	0.0000e+00	1.7523e-08	2.9582e-31	0.0000e+00	0.0000e+00	0.0000e+00

TABLE II

ERROR VALUES ACHIEVED WHEN FES=5×10^3, FES=5×10^4, FES=5×10^5 FOR PROBLEMS 7-12.

FES		g07	g08	g09	g10	g11	g12
5×10^3	Best	4.6886e+01(0)	5.5511e-17(0)	2.6961e+00(0)	3.8108e+03(0)	1.0128e-04(0)	7.7127e-09(0)
	Median	7.9176e+01(0)	5.5511e-17(0)	7.0678e+00(0)	6.7691e+03(0)	9.6709e-04(0)	3.7804e-06(0)
	Worst	1.9172e+02(0)	6.9389e-17(0)	1.5734e+01(0)	1.1338e+04(1)	2.7315e-02(0)	2.6201e-04(0)
	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0	0	0	0	0	0
	Mean	9.4585e+01	5.7176e-17	7.1479e+00	7.0671e+03	5.1297e-03	4.1230e-05
	Std	4.2166e+01	4.5097e-18	2.8734e+00	2.1834e+03	7.5607e-03	6.8340e-05
5×10^4	Best	1.8787e-03(0)	4.1633e-17(0)	6.0254e-12(0)	1.8756e-01(0)	8.5377e-07(0)	0.0000e+00(0)
	Median	3.4991e-03(0)	4.1633e-17(0)	2.7626e-11(0)	1.4117e+00(0)	4.4192e-06(0)	0.0000e+00(0)
	Worst	8.6878e-03(0)	4.1633e-17(0)	5.2864e-11(0)	4.8867e+00(0)	1.8001e-05(0)	0.0000e+00(0)
	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0	0	0	0	0	0
	Mean	3.8821e-03	4.1633e-17	2.4074e-11	1.6720e+00	5.6616e-06	0.0000e+00
	Std	1.5955e-03	1.2326e-32	1.4547e-11	1.1794e+00	4.3725e-06	0.0000e+00
5×10^5	Best	-1.8474e-13(0)	4.1633e-17(0)	0.0000e+00(0)	-1.8190e-12(0)	0.0000e+00(0)	0.0000e+00(0)
	Median	-1.8474e-13(0)	4.1633e-17(0)	0.0000e+00(0)	-9.0949e-13(0)	0.0000e+00(0)	0.0000e+00(0)
	Worst	-1.7764e-13(0)	4.1633e-17(0)	0.0000e+00(0)	-9.0949e-13(0)	0.0000e+00(0)	0.0000e+00(0)
	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0	0	0	0	0	0
	Mean	-1.8360e-13	4.1633e-17	0.0000e+00	-1.2005e-12	0.0000e+00	0.0000e+00
	Std	2.1831e-15	1.2326e-32	0.0000e+00	4.2426e-13	0.0000e+00	0.0000e+00

TABLE III

ERROR VALUES ACHIEVED WHEN FES=5×10^3, FES=5×10^4, FES=5×10^5 FOR PROBLEMS 13-18.

FES		g13	g14	g15	g16	g17	g18
5×10^3	Best	6.8682e-03(0)	-5.5835e+01(3)	1.7087e-03(0)	1.5655e-02(0)	-1.2518e+04(4)	3.0115e-01(0)
	Median	9.4527e-01(0)	-6.3934e+00(3)	5.3313e+00(0)	3.1744e-02(0)	2.2086e+01(4)	5.0207e-01(0)
	Worst	1.9008e+01(3)	3.1467e+01(3)	1.0583e+01(0)	5.3421e-02(0)	5.3931e+03(4)	7.4351e-01(0)
	c	0,0,0	1,2,0	0,0,0	0,0,0	4,0,0	0,0,0
	$\overline{v}$	0	0.537825	0	0	17.2838	0
	Mean	1.6460e+00	7.1255e+00	3.5957e+00	3.1936e-02	9.4350e+01	5.2434e-01
	Std	3.6392e+00	1.8956e+01	3.4024e+00	9.8520e-03	2.8771e+03	1.1801e-01
5×10^4	Best	9.6609e-06(0)	1.0179e+00(2)	1.2786e-04(0)	4.4409e-15(0)	3.1910e+00(0)	1.0482e-04(0)
	Median	2.9700e-05(0)	3.6165e+00(2)	2.8144e-01(0)	4.8850e-15(0)	2.8851e+01(0)	3.3936e-04(0)
	Worst	2.4225e-04(0)	9.0128e+00(3)	8.9106e+00(0)	8.2157e-15(0)	9.6000e+01(0)	9.9006e-04(0)
	c	0,0,0	0,0,2	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0	0.00100191	0	0	0	0
	Mean	7.4130e-05	3.9167e+00	1.3269e+00	5.0271e-15	4.3680e+01	3.9829e-04
	Std	6.9894e-05	1.6546e+00	2.2109e+00	8.1616e-16	3.6996e+01	2.2200e-04
5×10^5	Best	-9.7145e-17(0)	1.4211e-14(0)	0.0000e+00(0)	4.4409e-15(0)	1.8190e-12(0)	3.3307e-16(0)
	Median	-9.7145e-17(0)	2.1316e-14(0)	0.0000e+00(0)	4.4409e-15(0)	1.8190e-12(0)	3.3307e-16(0)
	Worst	-9.7145e-17(0)	2.1316e-14(0)	0.0000e+00(0)	4.4409e-15(0)	1.8190e-12(0)	4.4409e-16(0)
	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0	0	0	0	0	0
	Mean	-9.7145e-17	2.1032e-14	0.0000e+00	4.4409e-15	1.8190e-12	3.3751e-16
	Std	0.0000e+00	1.3924e-15	0.0000e+00	1.5777e-30	1.2117e-27	2.1756e-17

TABLE IV

ERROR VALUES ACHIEVED WHEN FES=5×10^3, FES=5×10^4, FES=5×10^5 FOR PROBLEMS 19-24.

FES		g19	g20	g21	g22	g23	g24
$\mathbf{5 \times 10^3}$	Best	5.9492e+02(0)	-1.5484e-01(11)	3.7492e+00(0)	1.9260e+03(8)	-1.2860e+02(5)	2.3436e-08(0)
	Median	9.1435e+02(0)	-1.0152e-01(19)	1.5609e+02(0)	4.4288e+03(9)	6.6472e+02(2)	5.5794e-08(0)
	Worst	1.6073e+03(0)	-2.7685e-02(8)	8.0628e+02(0)	1.9764e+04(9)	7.7755e+02(4)	1.5085e-06(0)
	c	0,0,0	0,18,1	0,0,0	6,3,0	1,1,0	0,0,0
	$\overline{v}$	0	0.0570694	0	144697	0.206346	0
	Mean	9.6318e+02	-1.0437e-01	2.3846e+02	1.2538e+04	4.2495e+02	2.4962e-07
	Std	2.6696e+02	3.4515e-02	2.6674e+02	5.7869e+03	2.3660e+02	3.1250e-07
$\mathbf{5 \times 10^4}$	Best	7.1662e+00(0)	-1.5484e-01(11)	1.7963e+00(0)	4.1336e+00(4)	1.9146e+02(0)	5.7732e-14(0)
	Median	1.1604e+01(0)	-1.5484e-01(11)	2.0297e+01(0)	2.6074e+02(3)	3.7298e+02(0)	5.7732e-14(0)
	Worst	1.9302e+01(0)	-2.3216e-02(12)	2.2767e+02(0)	1.6530e+04(12)	7.7215e+02(0)	5.7732e-14(0)
	c	0,0,0	0,7,0	0,0,0	3,0,0	0,0,0	0,0,0
	$\overline{v}$	0	0.055165	0	3.33359	0	0
	Mean	1.1761e+01	-1.0463e-01	3.0442e+01	5.2812e+03	3.8749e+02	5.7732e-14
	Std	2.9576e+00	3.8246e-02	4.2519e+01	5.4645e+03	1.4470e+02	2.5244e-29
$\mathbf{5 \times 10^5}$	Best	5.2162e-08(0)	-6.6522e-02(9)	-2.8422e-14(0)	1.9518e+00(0)	0.0000e+00(0)	5.7732e-14(0)
	Median	5.2162e-08(0)	-2.4674e-02(8)	-2.8422e-14(0)	1.2332e+01(0)	0.0000e+00(0)	5.7732e-14(0)
	Worst	5.9840e-05(0)	1.4007e-01(9)	1.4211e-13(0)	6.8922e+01(0)	5.6843e-14(0)	5.7732e-14(0)
	c	0,0,0	0,1,0	0,0,0	0,0,0	0,0,0	0,0,0
	$\overline{v}$	0	0.0110007	0	0	0	0
	Mean	5.3860e-06	-1.6626e-02	-2.1600e-14	1.8369e+01	2.2737e-15	5.7732e-14
	Std	1.2568e-05	4.2362e-02	3.3417e-14	1.5690e+01	1.1139e-14	2.5244e-29

TABLE V

NUMBER OF FES TO ACHIEVE THE FIXED ACCURACY LEVEL $((f(\vec{x}) - f(\vec{x}*)) \leq 0.0001)$, SUCCESS RATE, FEASIBLE RATE AND SUCCESS PERFORMANCE.

Prob.	**Best**	**Median**	**Worst**	**Mean**	**Std**	**Feasible Rate**	**Success Rate**	**Success Performance**
g01	57122	59345	61712	59308	1156.2937	100%	100%	59308 (16979,0)
g02	126152	146911	175206	149825	15719.2512	100%	100%	149825 (59351,0)
g03	86748	89473	91328	89407	1077.3246	100%	100%	89407 (40217,188)
g04	24800	26098	28206	26216	909.2025	100%	100%	26216 (10125,0)
g05	96812	97379	98589	97431	402.5792	100%	100%	97431 (43998,445)
g06	6499	7316	8382	7381	460.6339	100%	100%	7381 (3544,0)
g07	69506	74476	78963	74303	2409.3691	100%	100%	74303 (20430,0)
g08	327	1182	1334	1139	194.0499	100%	100%	1139 (490,0)
g09	19530	23172	24790	23121	1152.4431	100%	100%	23121 (10791,0)
g10	93743	105799	122387	105234	6776.9647	100%	100%	105234 (17743,0)
g11	5407	16821	29510	16420	6570.7151	100%	100%	16420 (11064,67)
g12	1645	4155	5540	4124	812.5536	100%	100%	4124 (366,0)
g13	8287	33594	68608	34738	15958.0883	100%	100%	34738 (15152,113)
g14	106816	112526	121656	113439	4195.0850	100%	100%	113439 (37864,470)
g15	57729	87185	90593	84216	7147.1924	100%	100%	84216 (40857,423)
g16	12347	13001	13923	12986	400.6285	100%	100%	12986 (4770,0)
g17	97274	99021	100144	98861	791.6359	100%	100%	98861 (36786,246)
g18	51035	59232	72112	59153	4321.6982	100%	100%	59153 (11936,0)
g19	319636	354060	451685	356350	28215.6159	100%	100%	356350 (40415,0)
g20	—	—	—	—	—	0%	0%	—
g21	126194	133224	216905	135143	16958.7554	100%	100%	135143 (34633,483)
g22	—	—	—	—	—	100%	0%	—
g23	158742	193593	281071	200765	27671.0160	100%	100%	200765 (34437,533)
g24	2661	2928	3474	2952	184.8424	100%	100%	2952 (1925,0)

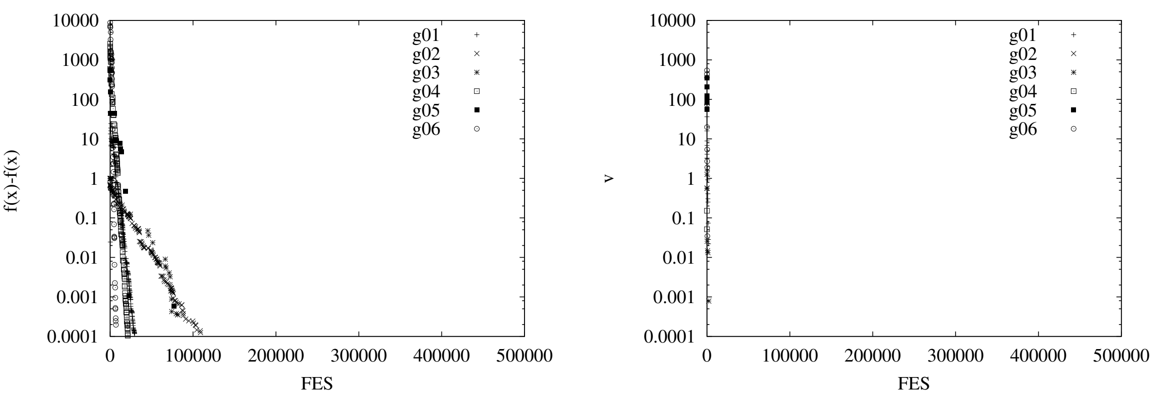

Fig. 1. Convergence Graph for Problems 1-6

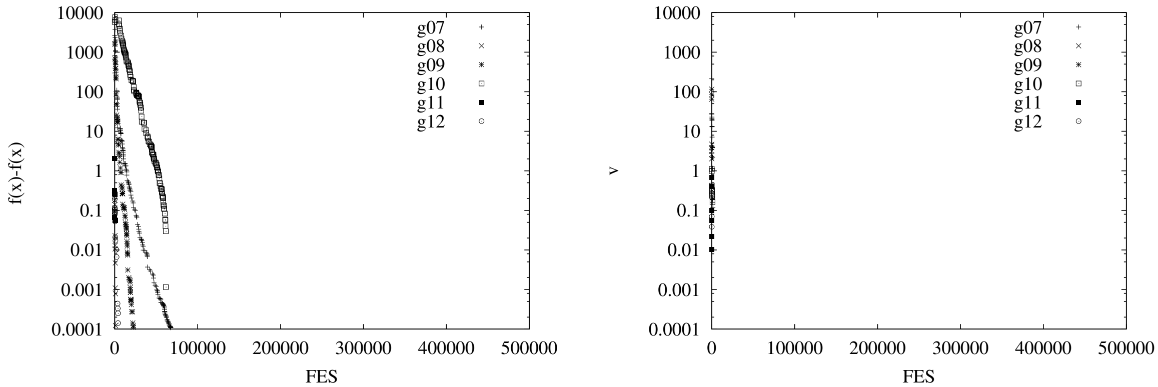

Fig. 2. Convergence Graph for Problems 7-12

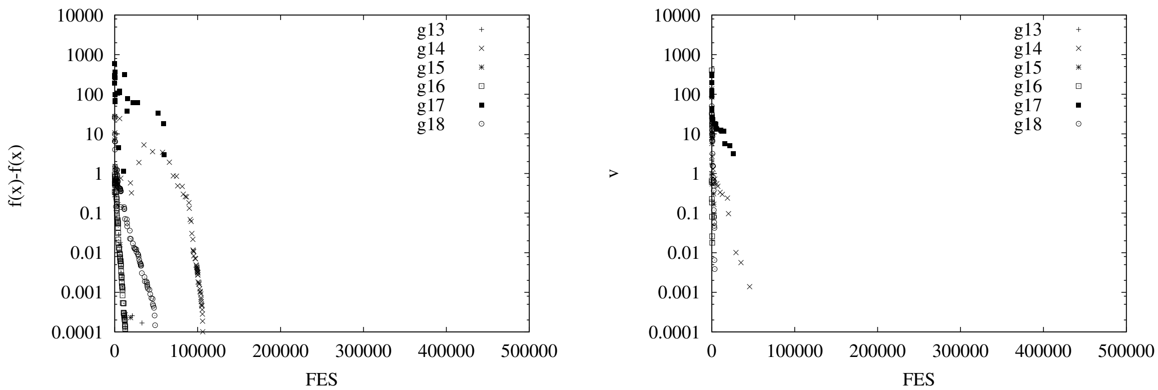

Fig. 3. Convergence Graph for Problems 13-18

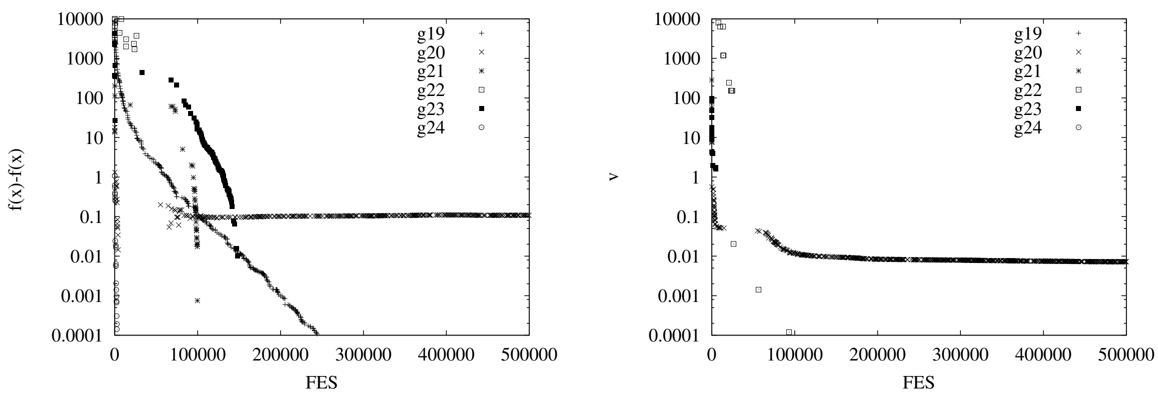

Fig. 4. Convergence Graph for Problems 19-24

[18] S. Kazarlis and V. Petridis, "Varying fitness functions in genetic algorithms: Studying the rate of increase of the dynamic penalty terms," in *Proceedings of the 5th Parallel Problem Solving from Nature (PPSN V)*, Amsterdan, The Netherlands. Heidelberg, Germany: Springer-Verlag, Sept. 1998, pp. 211–220.

[19] R. Farmani and J. A. Wright, "Self-adaptive fitness formulation for constrained optimization," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 5, pp. 445–455, Oct. 2003.

[20] C.Reeves, "Genetic algorithms for the operations researcher," *INFORMS Journal on Computing*, vol. 9, no. 3, pp. 231–250, 1997.

[21] T. Takahama and S. Sakai, "Learning fuzzy control rules by α-constrained simplex method," *Systems and Computers in Japan*, vol. 34, no. 6, pp. 80–90, June 2003.

[22] ——, "Constrained optimization by applying the α constrained method to the nonlinear simplex method with mutations," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 5, pp. 437–451, Oct. 2005.

[23] ——, "Constrained optimization by α constrained genetic algorithm (αGA)," *Systems and Computers in Japan*, vol. 35, no. 5, pp. 11–22, May 2004.

[24] ——, "Constrained optimization by the α constrained particle swarm optimizer," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 9, no. 3, pp. 282–289, May 2005.

[25] T. Takahama, S. Sakai, and N. Iwane, "Constrained optimization by the ε constrained hybrid algorithm of particle swarm optimization and genetic algorithm," in *Proc. of the 18th Australian Joint Conference on Artificial Intelligence*. Springer-Verlag, Dec. 2005, pp. 389–400.

[26] P. Chootinan and A. Chen, "Constraint handling in genetic algorithms using gradient-based repair method," *Computers & Operations Research*, vol. 33, no. 8, pp. 2263–2281, Aug. 2005, to appear.

[27] S.L.Campbell and C. J. Meyer, *Generalized Inverses of Linear Transformations*. Dover Publications, 1979.

[28] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. A. C. Coello, and K. Deb, "Problem definitions and evaluation criteria for the CEC2006 special session on constrained real-parameter optimization," 2006. [Online]. Available: http://www.ntu.edu.sg/home/EPNSugan/cec2006/technical_report.pdf