Efficient Constrained Optimization by the ε Constrained Differential Evolution with Rough Approximation Using Kernel Regression

Tetsuyuki Takahama
Department of Intelligent Systems
Hiroshima City University
Asaminami-ku, Hiroshima, 731-3194 Japan
Email: takahama@info.hiroshima-cu.ac.jp

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

***Abstract*—We have proposed to utilize a rough approximation model, which is an approximation model with low accuracy and without learning process, to reduce the number of function evaluations in unconstrained optimization. Although the approximation errors between the true function values and the approximation values estimated by the rough approximation model are not small, the rough model can estimate the order relation of two points with fair accuracy. In order to use this nature of the rough model, we have proposed estimated comparison which omits the function evaluations when the result of comparison can be judged by approximation values. In this study, we propose to utilize the estimated comparison in constrained optimization and propose the εDE^{kr}, which is the combination of the ε constrained method and the estimated comparison using kernel regression. The εDE^{kr} is a very efficient constrained optimization algorithm that can find high-quality solutions in very small number of function evaluations. It is shown that the εDE^{kr} can find near optimal solutions stably in very small number of function evaluations compared with various other methods on well known nonlinear constrained 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. There exist many studies on solving constrained optimization problems using evolutionary algorithms (EAs) [1]–[4]. EAs basically lack a mechanism to incorporate the constraints of a given problem in the fitness value of individuals. Thus, many studies have been dedicated to handle the constraints in EAs. In most successful constraint-handling techniques, the objective function value and the sum of constraint violations, or the constraint violation, are separately handled and an optimal solution is searched with balancing the optimization of the function value and the optimization of the constraint violation.

We have proposed the ε constrained differential evolution (εDE), which adopted one of such techniques called the ε *constrained method* and also adopted differential evolution (DE) as an optimization engine. The εDE can solve constrained problems successfully and stably [5]–[8], including engineering design problems [9]. The ε constrained method [6] is an algorithm transformation method, which can convert algorithms for unconstrained problems to algorithms for constrained problems using the ε level comparison, which compares search points based on the pair of objective value and constraint violation of them. It has been shown that the method has general-purpose properties.

Generally, a disadvantage of EAs is that they need a large number of function evaluations before a well acceptable solution can be found. An effective method for reducing the function evaluations is to build an approximation model for the objective function and to solve the problems using the approximation values [10]. If an approximation model with high accuracy can be build, it is possible to reduce the function evaluations largely. However, building the high-quality approximation model is very difficult and time-consuming. It needs to learn the model from many pairs of known solution and its function value. Also, a proper approximation model depends on the problems to be optimized. It is difficult to design a general-purpose approximation model with high accuracy.

We have proposed to utilize an approximation model with low accuracy and without learning process to reduce the number of function evaluations effectively. In the following, the approximation model is called a *rough approximation model*. Although the approximation errors between the true function values and the approximation values estimated by the rough approximation model are not small, the approximation model can estimate whether the function value of a point is smaller than that of the other point or not with fair accuracy. In order to use this nature of the rough model, we have proposed *estimated comparison* [11]–[13]. In the estimated comparison, the approximation values are compared first. When a value is worse enough than the other value, the estimated comparison returns an estimated result without evaluating the objective function. By using the estimated comparison, the evaluation of the objective function is sometimes omitted and the number of function evaluations can be reduced.

In this study, we propose to utilize the estimated comparison in constrained optimization and propose the εDE^{kr}, which is the combination of the ε constrained method and the estimated comparison using kernel regression. The kernel regression without learning process is adopted as a rough approximation model. The εDE^{kr} is a very efficient constrained optimization algorithm that can find high-quality solutions in very small number of function evaluations. Well known thirteen constrained problems mentioned in [2] are solved by the εDE^{kr} within very fewer, about one fourth, number of function evaluations. The effectiveness of the εDE^{kr} is shown by comparing it with various methods on the problems.

In Section II, constrained optimization methods and approximation methods are reviewed. The ε constrained method and the estimated comparison using kernel regression are explained in Section III and IV, respectively. The εDE^{kr} is proposed in Section V. In Section VI, experimental results on thirteen constrained problems are shown and the results of the εDE^{kr} are compared with those of other methods. Finally, conclusions are described in Section VII.

II. Constrained Optimization and Previous works

A. Constrained Optimization Problems

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

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

where $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ is an n dimensional vector, $f(\boldsymbol{x})$ is an objective function, $g_j(\boldsymbol{x}) \le 0$ and $h_j(\boldsymbol{x}) = 0$ are q inequality constraints and $m-q$ equality constraints, respectively. Functions f, g_j and h_j are linear or nonlinear real-valued functions. Values u_i and l_i are the upper bound and the lower bound of x^i, respectively. Also, let the feasible space in which every point satisfies all constraints be denoted by $\mathcal{F}$ and the search space in which every point satisfies the upper and lower bound constraints be denoted by $\mathcal{S}$ ($\supset \mathcal{F}$).

B. Constrained optimization methods

EAs for constrained optimization can be classified into several categories according to the way the constraints are treated as follows [3]:

(1) Constraints are only used to see whether a search point is feasible or not. Approaches in this category are usually called death penalty methods. In this category, generating initial feasible points is difficult and computationally demanding when the feasible region is very small.

(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 [14]–[17]. The main difficulty of the method is the selection of an appropriate value for the penalty coefficient that adjusts the strength of the penalty. In order to solve the difficulty, some methods, where a kind of the penalty coefficient is adaptively controlled [18], [19], are proposed.

(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. Deb [20] proposed a method that adopts the extended objective function, which realizes the lexicographic ordering. Takahama and Sakai proposed the α constrained method [21], and ε constrained method [22] that adopt a lexicographic ordering with relaxation of the constraints. Runarsson and Yao [23] proposed the stochastic ranking method that adopts the stochastic lexicographic order, which ignores the constraint violation with some probability. Mezura-Montes and Coello [24] proposed a comparison mechanism that is equivalent to the lexicographic ordering. Venkatraman and Yen [25] proposed a two-step optimization method, which first optimizes constraint violation and then objective function. 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 [26]–[32]. But in many cases, solving multiobjective optimization problems is a more difficult and expensive task than solving single objective optimization problems.

(5) Hybridization methods. In this category, constrained problems are solved by combining some of above mentioned methods. Mallipeddi and Suganthan [33] proposed a hybridization of the methods in category (2), (3) and (4).

C. Evolutionary algorithms using Approximation Models

In this section, EAs using approximation models are briefly reviewed.

Various approximation models are utilized to approximate the objective function. In most approximation models, model parameters are learned by least square method, gradient method, maximum likelihood method and so on. In general, learning model parameters is time-consuming process, especially in order to obtain models with higher accuracy and models of larger functions such as functions with large dimensions.

EAs with approximation models can be classified some types:

(1) All individuals have only approximation values. Very high quality approximation model is built and the objective function is optimized using approximation values only. It is possible to reduce function evaluations greatly. However, these methods can be applied well-informed objective function and cannot be applied to general problems.

(2) Some individuals have approximation values and others have true values. Methods in this type are called evolution control approaches and can be classified into individual-based and generation-based control. The individual-based control means that good individuals (or randomly selected individuals) use true values and others use approximation values in each generation [34], [35]. The generation-based control means that all individuals use true values once in a fixed number of generations and use approximation values in other generations [34], [36]. In the approaches, the approximation model should be accurate because approximation values are compared with true values. Also, it is known that approximation models with high accuracy sometimes generate a false optimum or hide a true optimum. Individuals may converge into a false optimum while they are optimized using the approximation models in some generations. Thus, these approaches are much affected by the quality of approximation models. It is difficult to utilize rough approximation models.

(3) All individuals have true values. Some methods in this type are called surrogate approaches. In the surrogate approaches, an estimated optimum is searched using an approximation model that is usually a local model. The estimated optimum is evaluated to obtain the true value and also to improve the approximation model [37]–[39]. If the true value is good, the

value is included as an individual. In the approaches, rough approximation models might be used because approximation values are compared with other approximation values. These approaches are less affected by the approximation model than the evolution control approaches. However, they have the process of optimization using the approximation model only. If the process is repeated many times, they are much affected by the quality of approximation models.

The estimated comparison method is classified into the last category because all individuals have true values. However, the method is different from the surrogate approaches. It uses a global approximation model of current individuals using the potential model. It does not search for an estimated optimum, but it judges whether a new individual is worth evaluating its true value or not. Also, it can specify the margin of approximation error when the comparison is carried out. Thus, it is not affected by the approximation model much.

III. The ε Constrained Method

A. Constraint violation and ε level comparisons

In the ε constrained method, constraint violation $\phi(\boldsymbol{x})$ is defined. The constraint violation can be given by the maximum of all constraints or the sum of all constraints.

$$\phi(\boldsymbol{x}) = \max\{\max_j\{0, g_j(\boldsymbol{x})\}, \max_j |h_j(\boldsymbol{x})|\} \quad (2)$$

$$\phi(\boldsymbol{x}) = \sum_j ||max\{0, g_j(\boldsymbol{x})\}||^p + \sum_j ||h_j(\boldsymbol{x})||^p \quad (3)$$

where p is a positive number.

The *ε level comparison* is defined as an order relation on a pair of objective function value and constraint violation $(f(\boldsymbol{x}), \phi(\boldsymbol{x}))$. If the constraint violation of a point is greater than 0, the point is not feasible and its worth is low. The ε level comparisons are defined basically as 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 $f(\boldsymbol{x})$. This precedence can be adjusted by the parameter ε.

Let f_1 (f_2) and ϕ_1 (ϕ_2) be the function values and the constraint violation at a point $\boldsymbol{x}_1$ $(\boldsymbol{x}_2)$, respectively. Then, for any ε satisfying $\varepsilon \geq 0$, ε level comparisons $<_\varepsilon$ and $\leq_\varepsilon$ between (f_1, ϕ_1) and (f_2, ϕ_2) are defined as follows:

$$(f_1, \phi_1) <_\varepsilon (f_2, \phi_2) \Leftrightarrow \begin{cases} f_1 < f_2, \text{ if } \phi_1, \phi_2 \leq \varepsilon \\ f_1 < f_2, \text{ if } \phi_1 = \phi_2 \\ \phi_1 < \phi_2, \text{otherwise} \end{cases} \quad (4)$$

$$(f_1, \phi_1) \leq_\varepsilon (f_2, \phi_2) \Leftrightarrow \begin{cases} f_1 \leq f_2, \text{ if } \phi_1, \phi_2 \leq \varepsilon \\ f_1 \leq f_2, \text{ if } \phi_1 = \phi_2 \\ \phi_1 < \phi_2, \text{otherwise} \end{cases} \quad (5)$$

In case of ε=∞, the ε level comparisons $<_\infty$ and $\leq_\infty$ are equivalent to the ordinary comparisons $<$ and $\leq$ between function values. Also, in case of $\varepsilon = 0$, $<_0$ and $\leq_0$ are equivalent to the lexicographic orders in which the constraint violation $\phi(\boldsymbol{x})$ precedes the function value $f(\boldsymbol{x})$.

B. The properties of the ε constrained method

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

$$(\mathrm{P}_{\leq_\varepsilon}) \quad \text{minimize}_{\leq_\varepsilon} \quad f(\boldsymbol{x}), \quad (6)$$

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 constraints of (P), that is, $\phi(\boldsymbol{x}) = 0$, is relaxed and replaced with $\phi(\boldsymbol{x}) \leq \varepsilon$:

$$\begin{array}{lll} (\mathrm{P}^\varepsilon) & \text{minimize} & f(\boldsymbol{x}) \\ & \text{subject to} & \phi(\boldsymbol{x}) \leq \varepsilon \end{array} \quad (7)$$

It is obvious that (P^0) is equivalent to (P).

For the three types of problems, (P^ε), $(\mathrm{P}_{\leq_\varepsilon})$ and (P), the following theorems are given based on the ε constrained method [22].

Theorem 1: If an optimal solution (P^0) exists, any optimal solution of $(\mathrm{P}_{\leq_\varepsilon})$ is an optimal solution of (P^ε).

Theorem 2: If an optimal solution of (P) exists, any optimal solution of $(\mathrm{P}_{\leq_0})$ is an optimal solution of (P).

Theorem 3: Let $\{\varepsilon_n\}$ be a strictly decreasing non-negative sequence and converge to 0. Let $f(\boldsymbol{x})$ and $\phi(\boldsymbol{x})$ be continuous functions of $\boldsymbol{x}$. Assume that an optimal solution $\boldsymbol{x}^*$ of (P^0) exists and an optimal solution $\hat{\boldsymbol{x}}_n$ of $(\mathrm{P}_{\leq_{\varepsilon_n}})$ exists for any ε_n. Then, any accumulation point to the sequence $\{\hat{\boldsymbol{x}}_n\}$ is an optimal solution of (P^0).

Theorem 1 and 2 show that a constrained optimization problem can be transformed into an equivalent unconstrained optimization problem by using the ε level comparison. So, if the ε level comparison is incorporated into an existing unconstrained optimization method, constrained optimization problems can be solved. Theorem 3 shows that, in the ε constrained method, an optimal solution of (P^0) can be given by converging ε to 0 as well as by increasing the penalty coefficient to infinity in the penalty method.

IV. Estimated Comparison Using Kernel Regression

The kernel regression is explained and the estimated comparison is briefly described.

A. Kernel Regression

The kernel regression is a nonparametric regression to estimate the regression function $y = f(x) + \varepsilon$ using a data set $\{(x_i, y_i) \,|\, i = 1, 2, ..., N\}$, where N is the number of data and ε is a small noise. The following Nadaraya-Watson estimator [40], [41], or a weighted average of function values y_i, where the weighting function is a kernel, is often used.

$$\hat{f}(x) = \frac{\sum_i K_h(x - x_i) y_i}{\sum_i K_h(x - x_i)} \quad (8)$$

$$K_h(u) = \frac{1}{h} K(u/h) \quad (9)$$

where $\hat{f}$ is the estimated function of f, and K is the kernel with a bandwidth h. The kernel K is a non-negative integrable function satisfying the following conditions:

$$\int_{-\infty}^{\infty} K(u) du = 1 \quad (10)$$

$$K(-u) = K(u) \quad \text{,for all } u \quad (11)$$

For example, the followings are representative kernels.

- Epanechnikov:

$$K(u) = \begin{cases} \frac{3}{4}(1-u^2) & (|u| \le 1) \\ 0 & \text{(otherwise)} \end{cases} \quad (12)$$

- Gaussian:

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2} \quad (13)$$

In this study, the following multiplicative kernel and the kernel regression are used for the multidimensional kernel and the approximation model:

$$K_{\boldsymbol{h}}(\boldsymbol{u}) = K_{h_1}(u_1)K_{h_2}(u_2)\cdots K_{h_n}(u_n) \quad (14)$$

$$\hat{f}(\boldsymbol{x}) = \frac{\sum_i K_{\boldsymbol{h}}(\boldsymbol{x}-\boldsymbol{x}_i)y_i}{\sum_i K_{\boldsymbol{h}}(\boldsymbol{x}-\boldsymbol{x}_i)} \quad (15)$$

where $\{(\boldsymbol{x}_i, y_i) \,|\, i = 1, 2, \cdots, N\}$ is a data set for estimation, $K_{\boldsymbol{h}}$ is a kernel, and $\boldsymbol{h}$ is a kernel bandwidth vector.

B. Bandwidth Selection

The performance of kernel regression are affected by the bandwidth or tuning parameter which controls the degree of complexity. The choice of bandwidth is very important. The asymptotically optimal bandwidth, which minimizes the asymptotic mean integrated squared error (AMISE), is the candidate of the bandwidth. The asymptotically optimal bandwidth for the j-th dimension is defined as follows [42]:

$$h_j^* = \hat{\sigma}_j C_\nu(K,n) N^{-1/(2\nu+n)} \quad (16)$$

$$C_\nu(K,n) = \left(\frac{\pi^{n/2}2^{n+\nu-1}(\nu!)^2R(K)^n}{\nu\kappa_\nu^2(K)((2\nu-1)!!+(n-1)((\nu-1)!!)^2)}\right)^{1/(2\nu+n)} \quad (17)$$

where ν is the order of the kernel and ν=2 in the kernels described above. $\hat{\sigma}_j$ is the sample variance of $\{x_{ij}\}$. $R(g) = \int_{-\infty}^{\infty} g(u)^2 du$ is the roughness of a function g. $\kappa_\nu(K) = \int_{-\infty}^{\infty} u^\nu K(u) du$ is the moment of the kernel. $R(K) = 1/(2\sqrt{\pi})$, $\kappa_\nu(K) = 1$ in Gaussian kernel and $R(K) = 3/5$, $\kappa_\nu(K) = 1/5$ in Epanechnikov kernel. The odd factorial means $(2s-1)!! = (2s-1)\cdots 5 \cdot 3 \cdot 1$.

In this study, Gaussian kernel is adopted. Therefore, the following bandwidth is used.

$$h_j = \alpha_h \hat{\sigma}_j \left(\frac{4}{n+2}\right)^{1/(4+n)} N^{-1/(4+n)} \quad (18)$$

We introduce the smoothness parameter α_h to realize smooth function approximation and $\alpha_h >= 1$ usually.

C. Estimated Comparison

When the true function values of all points in $P = \{\boldsymbol{x}_i, i = 1, 2, \cdots, N\}$ are known and a new child point $\boldsymbol{x}'_k$ is generated from a parent point $\boldsymbol{x}_k$, the approximation values at points $\boldsymbol{x}'_k$ are given as follows:

$$\hat{f}(\boldsymbol{x}'_k) = \frac{\sum_{i \ne k} K_{\boldsymbol{h}}(\boldsymbol{x}'_k - \boldsymbol{x}_i)y_i}{\sum_{i \ne k} K_{\boldsymbol{h}}(\boldsymbol{x}'_k - \boldsymbol{x}_i)} \quad (19)$$

Also, $\hat{f}(\boldsymbol{x}_k)$ is given by replacing $\boldsymbol{x}'_k$ with $\boldsymbol{x}_k$.

It should be noted that the parent point $\boldsymbol{x}_k (k = i)$ is omitted in the equation. If the parent point is not omitted, the approximation value of the parent point becomes almost true value. As the result, the difference between the precision of approximation at the parent point and that at the child point becomes big, and it is difficult to compare the approximation values.

The estimated comparison judges whether the child point is better than the parent point. In the comparison, the estimation error of the approximation model σ and a margin parameter for the approximation error δ are introduced. The function of the estimated comparison for constrained optimization using the ε constrained method can be defined as follows:

```
better_ε($\boldsymbol{x}'_k$, $\boldsymbol{x}_k$, $\sigma$) {
    if($\hat{f}(\boldsymbol{x}'_k) < \hat{f}(\boldsymbol{x}_k) + \delta\sigma$) {
        Evaluate $\boldsymbol{x}'_k$;
        if($(f(\boldsymbol{x}'_k), \phi(\boldsymbol{x}'_k)) <_\varepsilon (f(\boldsymbol{x}_k), \phi(\boldsymbol{x}_k))$)
            return yes;
    }
    return no;
}
```

where the true values at the parent point $(f(\boldsymbol{x}_k), \phi(\boldsymbol{x}_k))$ is known. The parameter $\delta \ge 0$ controls the margin value for the approximation error. When δ is 0, the estimated comparison can reject many children and omit a large number of function evaluations. However, the possibility of rejecting good child becomes high and a true optimum sometimes might be skipped. When δ is large, the possibility of rejecting good child becomes low. However, the estimated comparison can reject less children and omit a small number of function evaluations. Thus, δ should have a proper value.

The estimation error σ can be given as the standard deviation of errors between approximation values and true values.

V. The εDE^{kr}

In this section, DE is described first and then the ε constrained DE with estimated comparison using kernel regression (εDE^{kr}) is defined.

A. Differential Evolution

Differential evolution is proposed by Storn and Price [43]. 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.

There are some variants of DE that have been proposed, such as DE/best/1/bin and DE/rand/1/exp. The variants are classified using the notation DE/*base*/*num*/*cross*. "*base*" indicates the method of selecting a parent that will form the base vector. For example, DE/rand selects the parent for the base vector at random from the population. DE/best selects the best individual in the population. In DE/rand/1, for each individual $\boldsymbol{x}_i$, three individuals $\boldsymbol{x}_{p1}$, $\boldsymbol{x}_{p2}$ and $\boldsymbol{x}_{p3}$ are chosen from the population without overlapping $\boldsymbol{x}_i$ and each other. Fig. 1 shows that a new vector, or a mutant vector $\boldsymbol{x}^m$ is generated by the base vector $\boldsymbol{x}_{p1}$ and the difference vector $\boldsymbol{x}_{p2} - \boldsymbol{x}_{p3}$, where F is a scaling factor. "*num*" indicates the number of difference vectors used to perturb the base vector. "*cross*" indicates the crossover mechanism used to create a child. For example, 'bin' shows that the crossover is controlled by binomial crossover using constant crossover rate, and 'exp' shows that the crossover is controlled by a kind of two-point crossover using exponentially decreasing the crossover rate. A new child $\boldsymbol{x}'_i$ is generated from the parent $\boldsymbol{x}_i$ and the mutant vector $\boldsymbol{x}^m$, where CR is a crossover rate in Fig. 1.

```
mutation DE/rand/1
  $p1$=randint(1,$N$) s.t. $p1 \neq i$;
  $p2$=randint(1,$N$) s.t. $p2 \notin \{i,p1\}$;
  $p3$=randint(1,$N$) s.t. $p3 \notin \{i,p1,p2\}$;
  $\boldsymbol{x}^m$=$\boldsymbol{x}_{p1}$+$F(\boldsymbol{x}_{p2}-\boldsymbol{x}_{p3})$
exponential crossover DE/·/·/exp
  $k$=1; $j$=randint(1,$n$);
  do {
       $x'_{i,j}$=$x^m_j$;
       $k$=$k$+1; $j$=$(j+1)\%n$;
  } while($k \le n$ && $u(0,1) < CR$);
  while($k \le n$) {
       $x'_{i,j}$=$x_{i,j}$;
       $k$=$k$+1; $j$=$(j+1)\%n$;
  }
```

Fig. 1. Mutation and crossover operation, where randint(1,n) generates an integer randomly from $[1, n]$ and $u(l, r)$ is a uniform random number generator in $[l, r]$.

B. The algorithm of the εDE^{kr}

The εDE^{kr} is the DE that adopts the ε constrained method and the estimated comparison using kernel regression.

The algorithm of the εDE^{kr} is as follows:

1. Initialization of the individuals. Initial N individuals $\{\boldsymbol{x}_i, i = 1, 2, \cdots, N\}$ are generated randomly in search space $\mathcal{S}$ and form an initial population.
2. Initialization of the ε level. An initial ε level is given by the ε level control function $\varepsilon(0)$.
3. Termination condition. If the number of function evaluations exceeds the maximum number of evaluations $FE_{\max}$, the algorithm is terminated.
4. DE operation. Each individual $\boldsymbol{x}_i$ is selected as a parent. A trial vector or a child $\boldsymbol{x}'_i$ is generated by DE/rand/1/exp operation with a scaling factor F and a crossover rate CR.
5. Survivor selection. The estimated comparison is used for comparing the trial vector and the parent. The child $\boldsymbol{x}'_i$ is accepted for the next generation if the trial vector is better than the parent $\boldsymbol{x}_i$ by using the estimated comparison. Until all individuals are selected, go back to 4 in order to select the next individual as a parent.
6. Control of the ε level. The ε level is updated by the ε level control function $\varepsilon(t)$.
7. Go back to 3.

C. Controlling the ε level

The ε level is controlled according to Eqs. (20) and (21). The initial ε level $\varepsilon(0)$ is the constraint violation of the top θ-th individual in the initial search points. 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 violation.

```
$\varepsilon$DE$^{kr}$/rand/1/exp()
{
// Initialize the individuals
  $P$=$N$ individuals $\{\boldsymbol{x}_i\}$ generated randomly in $\mathcal{S}$;
// Initialize the $\varepsilon$ level
  $\varepsilon$=$\varepsilon(0)$;
  for($t$=1; termination condition is false; $t$++) {
    $\sigma$=estimation of approximation error in $P$;
    for($i$=1; $i \le N$; $i$++) {
      $\boldsymbol{x}'_i$=generated by DE/rand/1/exp operation;
// estimated comparison
      if(better$_\varepsilon$($\boldsymbol{x}'_i$, $\boldsymbol{x}_i$, $\sigma$)) $\boldsymbol{x}_i$=$\boldsymbol{x}'_i$;
    }
// Control the $\varepsilon$ level
    $\varepsilon$=$\varepsilon(t)$;
  }
}
```

Fig. 2. The algorithm of the ε constrained differential evolution with estimated comparison using kernel regression, where $\varepsilon(t)$ is the ε level control function.

$$\varepsilon(0) = \phi(\boldsymbol{x}_\theta) \quad (20)$$

$$\varepsilon(t) = \begin{cases} \varepsilon(0)(1-\frac{t}{T_c})^{cp}, & 0 < t < T_c, \\ 0, & t \ge T_c \end{cases} \quad (21)$$

where $\boldsymbol{x}_\theta$ is the top θ-th individual and $\theta = 0.2N$. This control is effective to solve problems with equality constraints.

Fig. 2 shows the algorithm of the εDE^{kr}.

VI. SOLVING NONLINEAR OPTIMIZATION PROBLEMS

In this paper, thirteen benchmark problems that are mentioned in some studies [3], [23], [24] are optimized, and the results by the εDE^{kr} are compared with those results.

A. Test problems and the experimental conditions

In the thirteen benchmark problems, problems g03, g05, g11 and g13 contain equality constraints. In problems with equality constraints, the equality constraints are relaxed and converted to inequality constraints according to Eq. (22), which is adopted in many methods:

$$|h_j(\boldsymbol{x})| \le 10^{-4} \quad (22)$$

Problem g12 has disjointed feasible regions. Table I shows the outline of the thirteen problems [24], [44]. The table contains the number of variables n, the form of the objective function, the number of linear inequality constraints (LI), nonlinear inequality constraints (NI), linear equality constraints (LE), nonlinear equality constraints (NE) and the number of constraints active at the optimal solution.

The parameters for εDE^{kr}are as follows: The number of search points $N = 40$, the maximum number of evaluations $FE_{\max} = 50,000$, the scaling factor $F = 0.7$, and the crossover rate $CR = 0.9$. The parameters for the ε constrained method are as follows: Every constraint violation is defined as a simple sum of constraints, or $p = 1$ in Eq. (3). The ε level is controlled using Eqs. (20) and (21) for problems with equality constraints and is 0 for the other problems. The control generation $T_c = 1500$ and the control parameter $cp = 5$. For the estimated comparison, the margin $\delta = 0.001$ and the

TABLE I. SUMMARY OF TEST PROBLEMS

f	n	Form of f	LI	NI	LE	NE	active
g01	13	quadratic	9	0	0	0	6
g02	20	nonlinear	1	1	0	0	1
g03	10	polynomial	0	0	0	1	1
g04	5	quadratic	0	6	0	0	2
g05	4	cubic	2	0	0	3	3
g06	2	cubic	0	2	0	0	2
g07	10	quadratic	3	5	0	0	6
g08	2	nonlinear	0	2	0	0	0
g09	7	polynomial	0	4	0	0	2
g10	8	linear	3	3	0	0	6
g11	2	quadratic	0	0	0	1	1
g12	3	quadratic	0	9^3	0	0	0
g13	5	nonlinear	0	0	1	2	3

smoothness $\alpha_h = 1.8$. In this paper, 30 independent runs are performed.

B. Experimental results

Table II summarizes the experimental results. The table shows the known "optimal" solution for each problem and the statistics from the 30 independent runs. These include the best, median, mean, and worst values and the standard deviation of the objective values found. Also, the average number of evaluations of the objective function and the constraints to find the best solution in each run is shown in the columns labeled #func and #const respectively. The last column shows how many evaluations of objective function can be omitted.

For problems g01, g04, g05, g06, g08, g09, g11, g12 and g13, the optimal solutions are found consistently in all 30 runs. For problems g03, g07 and g10, the optimal or near-optimal solutions are found in all 30 runs. These results show that the εDE^{kr} is a very efficient and stable algorithm. As for the problem g02, the problem is a multi-modal problem that has many local optima with peaks near the global optimum within the feasible region. Many other methods cannot constantly obtain high quality solutions, but the εDE^{kr} attained about -0.8 on average within 50,000 FEs. Thus, it is thought that the εDE^{kr} has a high ability to solve multi-modal problems.

In the ε constrained 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 constraint violation of the constraints, the objective function is not evaluated, or the evaluation of the objective function can often be omitted. Thus, the number of evaluations of the objective function is less than the number of evaluations of the constraints. This nature of the εDE^{kr} contributes to the efficiency of the algorithm especially when the objective function is computationally demanding. The number of evaluations of the constraint violations to find the best solution ranged from about 2,500 to 50,000. The number of evaluations of the objective function ranged between about 1,600 and 20,000. For these problems, the εDE^{kr} can omit the evaluation of the objective function about 20% to 85%. Therefore, the εDE^{kr} can find optimal solutions very efficiently, especially from the viewpoint of the number of evaluations for the objective function.

These results show that the εDE^{kr} is a very efficient and stable algorithm.

C. Comparison with other methods

There are some methods that solved the same thirteen problems. In the methods, for comparative studies we chose the simple multimembered evolution strategy (SMES) proposed by Mezura-Montes and Coello [24], the adaptive trade-off model (ATMES) proposed by Wang *et al.* [19], multiobjective method (HCOEA) proposed by Wang *et al.* [32], ECTHT-EP2 proposed by Mallipeddi and Suganthan [33], and the εDE proposed by Takahama and Sakai [6], because the results of these methods are better than the results of the other methods, and they reported good quality statistical information. Also, A-DDE proposed by Mezura-Montes and Palomeque-Ortiz [45], which adopts adaptive parameter control, is included in the comparison.

Table III shows the comparisons of the best, median, average, worst values and the standard deviation for the seven methods. The maximum number of FEs is also shown in "$FE_{\max}$".

All methods found optimal solutions in all 30 runs for g01, g03, g04, g08, g11 and g12. In other problems, from the viewpoint of quality of solutions, it is thought that the εDE are the best methods followed by ECHT-EP2 and the εDE^{kr}, where the difference between ECHT-EP2 and the εDE^{kr} is very small. However, the number of the function evaluations in the εDE^{kr} is very small, that is only about one fourth, compared with that in the εDE and ECHT-EP2. Thus, it is thought that the εDE^{kr} is better than the εDE and ECHT-EP2 from the viewpoint of the efficiency.

VII. CONCLUSIONS

Differential evolution is known as a simple, efficient and robust search algorithm that can solve unconstrained optimization problems. In this study, we proposed a new scheme of combining the ε constrained method and the estimated comparison using kernel regression to improve the efficiency, and proposed the εDE^{kr}. We showed that the εDE^{kr} could solve thirteen benchmark problems most efficiently compared with many other methods.

In the future, we will apply the εDE^{kr} to various real world problems that have expensive objective functions.

Acknowledgment

This research is supported in part by Grant-in-Aid for Scientific Research (C) (No. 22510166,24500177) of Japan society for the promotion of science and Hiroshima City University Grant for Special Academic Research (General Studies).

REFERENCES

[1] Z. Michalewicz, "A survey of constraint handling techniques in evolutionary computation methods," in *Proc. 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 applying the α constrained method to the nonlinear simplex method with mutations," *IEEE Trans. on Evolutionary Computation*, vol. 9, no. 5, pp. 437–451, Oct. 2005.

[4] E. Mezura-Montes and C. A. C. Coello, "Constraint-handling in nature-inspired numerical optimization: Past, present and future," *Swarm and Evolutionary Computation*, vol. 1, pp. 173–194, 2011.

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

f	optimal	best	median	mean	worst	st. dev.	#func	#const	omit(%)
g01	-15.000	-15.000000	-15.000000	-15.000000	-15.000000	3.451e-08	12881.6	49832.3	74.2
g02	-0.803619	-0.803239	-0.802214	-0.799512	-0.776774	6.698e-03	19535.4	49070.0	60.2
g03	-1.000	-1.000500	-1.000500	-1.000500	-1.000498	5.375e-07	19740.1	49869.6	60.4
g04	-30665.539	-30665.538672	-30665.538672	-30665.538672	-30665.538672	0.000e+00	12353.5	33361.6	63.0
g05	5126.498	5126.496714	5126.496714	5126.496714	5126.496714	0.000e+00	13103.9	46537.3	71.8
g06	-6961.814	-6961.813876	-6961.813876	-6961.813876	-6961.813876	2.583e-12	3374.5	7719.3	56.3
g07	24.306	24.306218	24.306244	24.306265	24.306492	5.675e-05	12748.8	49814.1	74.4
g08	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825	0.000e+00	2041.2	2537.9	19.6
g09	680.630	680.630057	680.630057	680.630057	680.630057	0.000e+00	14799.8	33221.5	55.5
g10	7049.248	7049.249458	7049.253941	7049.257873	7049.317701	1.332e-02	8353.0	49651.0	83.2
g11	0.750	0.749900	0.749900	0.749900	0.749900	0.000e+00	13119.3	31303.7	58.1
g12	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	0.000e+00	1615.1	3687.2	56.2
g13	0.053950	0.0539415	0.0539415	0.0539415	0.0539415	3.639e-17	14804.8	48602.9	69.5

[5] T. Takahama and S. Sakai, “Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites,” in *Proc. of the 2006 IEEE Congress on Evolutionary Computation*, July 2006, pp. 308–315.

[6] T. Takahama and S. Sakai, “Fast and stable constrained optimization by the ε constrained differential evolution,” *Pacific Journal of Optimization*, vol. 5, no. 2, pp. 261–282, May 2009.

[7] T. Takahama and S. Sakai, “Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation,” in *Proc. of the 2010 IEEE Congress on Evolutionary Computation*, July 2010, pp. 1680–1688.

[8] T. Takahama and S. Sakai, “Efficient constrained optimization by the ε constrained adaptive differential evolution,” in *Proc. of the 2010 IEEE Congress on Evolutionary Computation*, July 2010, pp. 2052–2059.

[9] T. Takahama, S. Sakai, and N. Iwane, “Solving nonlinear constrained optimization problems by the ε constrained differential evolution,” in *Proc. of the 2006 IEEE Conference on Systems, Man, and Cybernetics*, Oct. 2006, pp. 2322–2327.

[10] Y. Jin, “A comprehensive survey of fitness approximation in evolutionary computation,” *Soft Computing*, vol. 9, pp. 3–12, 2005.

[11] T. Takahama and S. Sakai, “Reducing function evaluations in differential evolution using rough approximation-based comparison,” in *Proc. of the 2008 IEEE Congress on Evolutionary Computation*, June 2008, pp. 2307–2314.

[12] T. Takahama and S. Sakai, “Efficient optimization by differential evolution using rough approximation model with adaptive control of error margin,” in *Proc. of the Joint 4th International Conference on Soft Computing and Intelligent Systems and 9th International Symposium on advanced Intelligent Systems*, Sept. 2008, pp. 1412–1417.

[13] S. Sakai and T. Takahama, “A parametric study on estimated comparison in differential evolution with rough approximation model,” in *Social Systems Solution by Legal Informatics, Economic Sciences and Computer Sciences*, M.Kitahara and K.Morioka, Eds. Fukuoka: Kyushu University Press, 2010, pp. 112–134.

[14] A. Homaifar, S. H. Y. Lai, and X. Qi, “Constrained optimization via genetic algorithms,” *Simulation*, vol. 62, no. 4, pp. 242–254, 1994.

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

[16] Z. Michalewicz and N. Attia, “Evolutionary optimization of constrained problems,” in *Proc. of the 3rd Annual Conference on Evolutionary Programming*, A. Sebald and L. Fogel, Eds. River Edge, NJ: World Scientific Publishing, 1994, pp. 98–108.

[17] C. A. C. Coello, “Use of a self-adaptive penalty approach for engineering optimization problems,” *Computers in Industry*, vol. 41, no. 2, pp. 113–127, 2000.

[18] B. Tessema and G. Yen, “A self adaptive penalty function based algorithm for constrained optimization,” in *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, G. G. Yen, S. M. Lucas, G. Fogel, G. Kendall, R. Salomon, B.-T. Zhang, C. A. C. Coello, and T. P. Runarsson, Eds. Vancouver, BC, Canada: IEEE Press, 16-21 July 2006, pp. 246–253.

[19] Y. Wang, Z. Cai, Y. Xhau, and W. Zeng, “An adaptive tradeoff model for constrained evolutionary computation,” *IEEE Trans. on Evolutionary Computation*, vol. 12, no. 1, pp. 80–92, 2008.

[20] 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.

[21] T. Takahama and S. Sakai, “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.

[22] 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)*, May 2005, pp. 1019–1029. [Online]. Available: http://www.ints.info.hiroshima-cu.ac.jp/˜takahama/eng/papers/ePSO05c.pdf

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

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] C. A. C. Coello, “Constraint-handling using an evolutionary multiobjective optimization technique,” *Civil Engineering and Environmental Systems*, vol. 17, pp. 319–346, 2000.

[29] T. Ray, K. M. Liew, and P. Saini, “An intelligent information sharing strategy within a swarm for unconstrained and constrained optimization problems,” *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, vol. 6, no. 1, pp. 38–44, Feb. 2002.

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

[31] 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.

[32] Y. Wang, Z. Cai, G. Cuo, and Z. Zhou, “Multiobjective optimization and hybrid evolutionary algorthm to solve constrained optimization problems,” *IEEE Trans. on Systems, Man and Cybernetics, Part B*, vol. 37, no. 3, pp. 560–575, 2007.

[33] R. Mallipeddi and P. N. Suganthan, “Ensemble of constraint handling

TABLE III. COMPARISON OF STATISTICAL RESULTS AMONG THE εDE^{kr}, THE εDE [6], SMES [24], ATMES [19], HCOEA [32], ECHT-EP2 [33] AND A-DDE [45].

f & optimal	Statistics	εDE^{kr}	εDE	SMES	ATMES	HCOEA	ECHT-EP2	A-DDE
	$FE_{\max}$	**50,000**	200,000	240,000	240,000	240,000	240,000	180,000
g01 -15.000	best	-15.000000	-15.000000	-15.000	-15.000	-15.000000	-15.0000	-15.000
	median	-15.000000	-15.000000	-15.000	-15.000	-15.000000	-15.0000	-15.000
	mean	-15.000000	-15.000000	-15.000	-15.000	-15.000000	-15.0000	-15.000
	worst	-15.000000	-15.000000	-15.000	-15.000	-14.999998	-15.0000	-15.000
	σ	3.45e-08	**0.00e+00**	**0.00e+00**	1.6e-14	4.297e-07	**0.00e+00**	7.00e-06
g02 -0.803619	best	-0.803239	-0.803618	-0.803601	-0.803388	-0.803241	-0.8036191	-0.803605
	median	-0.802214	-0.803614	-0.792549	-0.792420	-0.802556	-0.8033239	-0.777368
	mean	-0.799512	-0.803613	-0.785238	-0.790148	-0.801258	-0.7998220	-0.771090
	worst	-0.776774	-0.803588	-0.751322	-0.756986	-0.792363	-0.7851820	-0.609853
	σ	6.698e-03	**5.59e-06**	1.67e-02	1.3e-02	3.832e-03	6.29e-03	3.66e-02
g03 -1.0005	best	-1.000500	-1.000500	-1.000	-1.000	-1.000000	-1.0005	-1.000
	median	-1.000500	-1.000500	-1.000	-1.000	-1.000000	-1.0005	-1.000
	mean	-1.000500	-1.000500	-1.000	-1.000	-1.000000	-1.0005	-1.000
	worst	-1.000498	-1.000500	-1.000	-1.000	-1.000000	-1.0005	-1.000
	σ	5.38e-07	6.46e-09	2.09e-04	5.9e-05	1.304e-12	**0.0e+00**	9.30e-12
g04 -30665.5387	best	-30665.538672	-30665.538670	-30665.539	-30665.539	-30665.539	-30665.5387	-30665.539
	median	-30665.538672	-30665.538670	-30665.539	-30665.539	-30665.539	-30665.5387	-30665.539
	mean	-30665.538672	-30665.538670	-30665.539	-30665.539	-30665.539	-30665.5387	-30665.539
	worst	-30665.538672	-30665.538670	-30665.539	-30665.539	-30665.539	-30665.5387	-30665.539
	σ	**0.00e+00**	**0.00e+00**	**0.00e+00**	7.4e-12	5.404e-07	**0.0e+00**	3.20e-13
g05 5126.4967	best	5126.496714	5126.496714	5126.599	5126.498	5126.4981	5126.4967	5126.497
	median	5126.496714	5126.496714	5160.198	5126.776	5126.4981	5126.4967	5126.497
	mean	5126.496714	5126.496714	5174.492	5127.648	5126.4981	5126.4967	5126.497
	worst	5126.496714	5126.496714	5304.167	5135.256	5126.4984	5126.4967	5126.497
	σ	**0.00e+00**	1.82e-12	5.006e+01	1.8e+00	1.727e-07	**0.0e+00**	2.10e-11
g06 -6961.8139	best	-6961.813876	-6961.813876	-6961.814	-6961.814	-6961.81388	-6961.8139	-6961.814
	median	-6961.813876	-6961.813876	-6961.814	-6961.814	-6961.81388	-6961.8139	-6961.814
	mean	-6961.813876	-6961.813876	-6961.284	-6961.814	-6961.81388	-6961.8139	-6961.814
	worst	-6961.813876	-6961.813876	-6952.482	-6961.814	-6961.81388	-6961.8139	-6961.814
	σ	2.58e-12	**0.00e+00**	1.85e+00	4.6e-12	8.507e-12	**0.00e+00**	2.11e-12
g07 24.3062	best	24.306218	24.306209	24.327	24.306	24.3064582	24.3062	24.306
	median	24.306244	24.306209	24.426	24.313	24.3073055	24.3063	24.306
	mean	24.306265	**24.306209**	24.475	24.316	24.3073989	24.3063	24.306
	worst	24.306492	24.306209	24.843	24.359	24.3092401	24.3063	24.306
	σ	5.675e-05	**4.27e-09**	1.32e-01	1.1e-02	7.118e-04	3.19e-05	4.20e-05
g08 -0.095825	best	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825	-0.09582504	-0.095825
	median	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825	-0.09582504	-0.095825
	mean	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825	-0.09582504	-0.095825
	worst	-0.095825	-0.095825	-0.095825	-0.095825	-0.095825	-0.09582504	-0.095825
	σ	**0.00e+00**	**0.00e+00**	**0.00e+00**	2.8e-17	2.417e-17	**0.0e+00**	9.10e-10
g09 680.630057	best	680.630057	680.630057	680.632	680.630	680.6300574	680.630057	680.63
	median	680.630057	680.630057	680.642	680.633	680.6300574	680.630057	680.63
	mean	680.630057	680.630057	680.643	680.639	680.6300574	680.630057	680.63
	worst	680.630057	680.630057	680.719	680.673	680.6300578	680.630057	680.63
	σ	**0.00e+00**	**0.00e+00**	1.55e-02	1.0e-02	9.411e-08	2.61e-08	1.15e-10
g10 7049.248	best	7049.249458	7049.248021	7051.903	7052.253	7049.286598	7049.2483	7049.248
	median	7049.253941	7049.248021	7253.603	7215.357	7049.486145	7049.2488	7049.248
	mean	7049.257873	**7049.248021**	7253.047	7250.437	7049.525438	7049.2490	7049.248
	worst	7049.317701	7049.248021	7638.366	7560.224	7049.984208	7049.2501	7049.248
	σ	1.33e-02	**0.00e+00**	1.36e+02	1.2e+02	1.502e-01	6.60e-04	3.23e-4
g11 0.749900	best	0.749900	0.749900	0.75	0.75	0.750000	0.7499	0.75
	median	0.749900	0.749900	0.75	0.75	0.750000	0.7499	0.75
	mean	0.749900	0.749900	0.75	0.75	0.750000	0.7499	0.75
	worst	0.749900	0.749900	0.75	0.75	0.750000	0.7499	0.75
	σ	**0.00e+00**	**0.00e+00**	1.52e-04	3.4e-04	1.546e-12	**0.0e+00**	5.35e-15
g12 -1.000	best	-1.000000	-1.000000	-1.0000	-1.000	-1.000000	-1.0000	-1.000
	median	-1.000000	-1.000000	-1.0000	-1.000	-1.000000	-1.0000	-1.000
	mean	-1.000000	-1.000000	-1.0000	-1.000	-1.000000	-1.0000	-1.000
	worst	-1.000000	-1.000000	-1.0000	-0.994	-1.000000	-1.0000	-1.000
	σ	**0.00e+00**	**0.00e+00**	**0.00e+00**	1.0e-03	**0.00e+00**	**0.0e+00**	4.10e-11
g13 0.0539415	best	0.0539415	0.053942	0.053986	0.053950	0.0539498	0.053941514	0.053942
	median	0.0539415	0.053942	0.061873	0.053952	0.0539498	0.053941514	0.053942
	mean	**0.0539415**	**0.053942**	0.166385	0.053959	0.0539498	**0.053941514**	0.079627
	worst	0.0539415	0.053942	0.468294	0.053999	0.0539499	0.053941514	0.438803
	σ	3.64e-17	**0.00e+00**	1.77e-01	1.3e-05	8.678e-08	1.00e-12	9.60e-02

techniques," *IEEE Transactions on Evolutionary Computation*, 2010, to appear.

[34] Y. Jin, M. Olhofer, and B. Sendhoff, "On evolutionary optimization with approximate fitness functions," in *Proc. of the Genetic and Evolutionary Computation Conference*. Morgan Kaufmann, 2000, pp. 786–792.

[35] Y. Jin and B. Sendhoff, "Reducing fitness evaluations using clustering techniques and neural networks ensembles," in *Genetic and Evolutionary Computation Conference*, ser. LNCS, vol. 3102. Springer, 2004, pp. 688–699.

[36] Y. Jin, M. Olhofer, and B. Sendhoff, "A framework for evolutionary optimization with approximate fitness functions," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 5, pp. 481–494, 2002.

[37] D. Büche, N. N. Schraudolph, and P. Koumoutsakos, "Accelerating evolutionary algorithms with gaussian process fitness function models," *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 35, no. 2, pp. 183–194, May 2005.

[38] Y. S. Ong, Z. Zhou, and D. Lim, "Curse and blessing of uncertainty in evolutionary algorithm using approximation," in *Proc. of the 2006 IEEE Congress on Evolutionary Computation*, Vancouver, BC, Canada, July 2006, pp. 9833–9840.

[39] F. G. Guimarães, E. F. Wanner, F. Campelo, R. H. Takahashi, H. Igarashi, D. A. Lowther, and J. A. Ramírez, "Local learning and search in memetic algorithms," in *Proc. of the 2006 IEEE Congress on Evolutionary Computation*, Vancouver, BC, Canada, July 2006, pp. 9841–9848.

[40] E. A. Nadaraya, "On estimating regression," *Theory of Probability and its Applications*, vol. 9, no. 1, pp. 141–142, 1964.

[41] G. S. Watson, "Smooth regression analysis," *Sankhya*, vol. 26, pp. 359–372, 1964.

[42] D. J. Henderson and C. F. Parmeter, "Normal reference bandwidths for the general order, multivariate kernel density derivative estimator," *Statistics & Probability Letters*, vol. 82, no. 12, pp. 2198–2205, 2012.

[43] 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.

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

[45] E. Mezura-Montes and A. G. Palomeque-Ortiz, "Parameter control in differential evolution for constrained optimization," in *Proc. of the 2009 IEEE Congress on Evolutionary Computation*, 2009, pp. 1375–1382.