Ensemble of Constraint Handling Techniques

Rammohan Mallipeddi and Ponnuthurai N. Suganthan, *Senior Member, IEEE*

***Abstract*—During the last three decades, several constraint handling techniques have been developed to be used with evolutionary algorithms (EAs). According to the no free lunch theorem, it is impossible for a single constraint handling technique to outperform all other techniques on every problem. In other words, depending on several factors such as the ratio between feasible search space and the whole search space, multimodality of the problem, the chosen EA, and global exploration/local exploitation stages of the search process, different constraint handling methods can be effective during different stages of the search process. Motivated by these observations, we propose an ensemble of constraint handling techniques (ECHT) to solve constrained real-parameter optimization problems, where each constraint handling method has its own population. A distinguishing feature of the ECHT is the usage of every function call by each population associated with each constraint handling technique. Being a general concept, the ECHT can be realized with any existing EA. In this paper, we present two instantiations of the ECHT using four constraint handling methods with the evolutionary programming and differential evolution as the EAs. Experimental results show that the performance of ECHT is better than each single constraint handling method used to form the ensemble with the respective EA, and competitive to the state-of-the-art algorithms.**

***Index Terms*—Constrained optimization, differential evolution, ensemble of constraint handling techniques, ensemble of optimization algorithms, evolutionary programming.**

I. Introduction

Many OPTIMIZATION problems in science and engineering involve constraints. The presence of constraints reduces the feasible region and complicates the search process. Evolutionary algorithms (EAs) always perform unconstrained search. When solving constrained optimization problems, they require additional mechanisms to handle constraints. In the literature, several constraint handling techniques have been proposed to be used with the EAs [1].

When solving constrained optimization problems, solution candidates that satisfy all the constraints are feasible individuals while individuals that fail to satisfy any of the constraints are infeasible individuals. One of the major issues in constraint optimization is how to deal with the infeasible individuals throughout the search process. One way to handle it is to completely disregard infeasible individuals and continue the search process with feasible individuals only. This approach may be ineffective as EAs are probabilistic search methods and potential information present in infeasible individuals can be wasted. If the search space is discontinuous, then the EA can also be trapped in one of the local minima. Therefore, different techniques have been developed to exploit the information in infeasible individuals. Michalewicz and Schoenauer [2] grouped the methods for handling constraints within EAs into four categories: preserving feasibility of solutions [3], penalty functions, make a separation between feasible and infeasible solutions, and hybrid methods. A constrained optimization problem can also be formulated as a multiobjective [4] problem, but it is computationally intensive due to nondomination sorting.

According to the no free lunch (NFL) theorem [5], no single state-of-the-art constraint handling technique can outperform all others on every problem. Hence, solving a particular constrained problem requires numerous trial-and-error runs to choose a suitable constraint handling technique and to fine tune the associated parameters. This approach clearly suffers from unrealistic computational requirements in particular if the objective function is computationally expensive [6] or solutions are required in real-time. In this paper, an ensemble of constraint handling techniques (ECHT) with four constraint handling techniques [1], [7]–[9] is proposed as an efficient alternative to the trial-and-error-based search for the best constraint handling technique with its best parameters for a given problem. In ECHT, each constraint handling technique has its own population and each function call is efficiently utilized by each of these populations.

Different EAs such as differential evolution (DE) [7], [10]–[15], particle swarm optimizer [16], [17], evolution strategies [8], [18]–[22], evolutionary programming (EP) [23], [24] and others [25], [26] have been used to solve constrained optimization problems. In addition, EP and ES are similar. Recently the usage of DE to solve constrained problems is also gaining importance. Being a general concept, the ECHT can be realized with any of the existing EAs. In this paper, we present two instantiations of the ECHT using the DE (ECHT-DE) and the EP (ECHT-EP).

The remainder of this paper is organized as follows. Section II presents the global constrained optimization problem and constraint handling techniques used with the EAs. Section III presents the generalized ECHT algorithm. Sections IV and V present ECHT-EP and ECHT-DE, respectively. The results and comparative discussions are presented in Section VI. Section VII concludes the paper.

Manuscript received May 24, 2008; revised August 4, 2009. Date of publication January 22, 2010; date of current version July 30, 2010. This work was supported by the Agency for Science, Technology, and Research, Singapore, under Grant 052 101 0020.

The authors are with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore 639798, Singapore (e-mail: mallipeddi.ram@gmail.com; epnsugan@ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEVC.2009.2033582

II. Constraint Handling Methods

Most practical optimization problems have constraints. A constrained optimization problem with n parameters to be optimized is usually written as a nonlinear programming problem of the following form [27]:

$$\begin{aligned} \text{Minimize :} \quad & f(X), \qquad X = (x_1, x_2, \ldots, x_n) \text{ and } X \in S \\ \text{subject to :} \quad & g_i(X) \leq 0, \quad i = 1, \ldots, p \\ & h_j(X) = 0, \quad j = p+1, \ldots, m. \end{aligned} \tag{1}$$

Here, f does not need to be continuous but it must be bounded. S is the whole search space. p is the number of inequality constraints. The number of equality constraints is $(m - p)$. The inequality constraints that satisfy $g_i(X) = 0$ at the global optimum solution are called active constraints. All equality constraints are active constraints. The equality constraints can be transformed into inequality form and can be combined with other inequality constraints as

$$G_i(X) = \begin{cases} \max\{g_i(X), 0\}, & i = 1, \ldots, p \\ \max\{|h_i(X)| - \delta, 0\}, & i = p+1, \ldots, m \end{cases} \tag{2}$$

where δ is a tolerance parameter for the equality constraints. An adaptive setting of the tolerance parameter, which is originally proposed in [28] and used in [18], [19], and [22] is adopted in this paper with some modifications as explained in Section VI. Therefore, the objective is to minimize the fitness function $f(X)$ such that the optimal solution obtained satisfies all the inequality constraints $G_i(X)$. The overall constraint violation for an infeasible individual is a weighted mean of all the constraints, which is expressed as

$$\upsilon(X) = \frac{\sum_{i=1}^{m} w_i(G_i(X))}{\sum_{i=1}^{m} w_i} \tag{3}$$

where w_i $(= 1/G_{\max_i})$ is a weight parameter, $G_{\max_i}$ is the maximum violation of constraint $G_i(X)$ obtained so far. Here, w_i is set as $1/G_{\max_i}$ which varies during the evolution in order to balance the contribution of every constraint in the problem irrespective of their differing numerical ranges.

The search process of finding the feasible global optimum in a constrained problem can be divided into three phases [22] depending on the number of feasible solutions present in the combined parent population and its offspring population as: 1) Phase 1—no feasible solution; 2) Phase 2—at least one feasible solution; and 3) Phase 3—combined offspring-parent population has more feasible solutions than the size of next generation parent population. Different constraint handling techniques perform differently during each of these three phases.

A. Superiority of Feasible Solutions (SF)

In SF [29], [30], when two solutions X_i and X_j are compared, X_i is regarded as superior to X_j under the following conditions.

1) X_i is feasible and X_j is not.
2) X_i and X_j are both feasible and X_i has a smaller objective value (in a minimization problem) than X_j.
3) X_i and X_j are both infeasible, but X_i has a smaller overall constraint violation $\upsilon(X_i)$ as computed by using (3).

Therefore, in SF, feasible ones are always considered better than the infeasible ones. Two infeasible solutions are compared based on their overall constraint violations only, while two feasible solutions are compared based on their objective function values only. Comparison of infeasible solutions based on the overall constraint violation aims to push the infeasible solutions to feasible region, while comparison of two feasible solutions on the objective value improves the overall solution. Therefore, in Phase 1, infeasible solutions with low overall constraint violation are selected. In Phase 2, all the feasible ones are selected first and then infeasible ones with low overall constraint violation are selected. In Phase 3, only feasible ones with best objective values are selected.

By primarily using the SF comparisons presented above, a simple multimembered evolution strategy (SMES) [19] was proposed. The main features of SMES are: 1) a simple diversity mechanism similar to $(1 + \lambda)$-ES to allow infeasible solutions to remain in the population; 2) the combined recombination; and 3) reduction in the initial step size and self-adaptation of ES. In SMES, the selection procedure in all the three phases of evolution remains the same as in the SF but preserves the best infeasible individual (infeasible individual with the least constraint violation and best objective value) by adopting a diversity mechanism.

B. Self-Adaptive Penalty (SP)

The simplest and the earliest method of involving infeasible individuals in the search process even after sufficient number of feasible solutions are obtained, is the static penalty method. In this method, a penalty value is added to the fitness value of each infeasible individual so that it will be penalized for violating the constraints. Static penalty functions are popular due to their simplicity but they usually require different parameters to be defined by the user to control the amount of penalty added when multiple constraints are violated. The parameters are usually problem-dependant. To overcome this difficulty, adaptive penalty functions [31] are suggested where information gathered from the search process will be used to control the amount of penalty added to infeasible individuals. Adaptive penalty functions are easy to implement and they do not require users to define parameters.

In [9], a self-adaptive penalty function method is proposed to solve constrained optimization problems. Two types of penalties are added to each infeasible individual to identify the best infeasible individuals in the current population. The amount of the added penalties is controlled by the number of feasible individuals currently present in the combined population. If there are a few feasible individuals, a higher amount of penalty is added to infeasible individuals with a higher amount of constraint violation. On the other hand, if there are several feasible individuals, then infeasible individuals with high-fitness values will have small penalties added to their fitness values. These two penalties will allow the algorithm to switch between finding more feasible solutions and searching for the optimum solution at anytime during the search process.

This algorithm requires no parameter tuning. The final fitness value based on which the population members are ranked is given as $F(X) = d(X) + p(X)$, where $d(X)$ is the distance value and $p(X)$ is the penalty value. The distance value is computed as follows:

$$d(X) = \begin{cases} \upsilon(X), & \text{if } r_f = 0 \\ \sqrt{f''(X)^2 + \upsilon(X)^2}, & \text{otherwise} \end{cases} \quad (4)$$

where r_f = (Number of feasible individuals)/(population size), $\upsilon(X)$ is the overall constrain violation as defined in (3), $fc(X) = (f(X) - f_{\min})/(f_{\max} - f_{\min})$, where $f_{\max}$ and $f_{\min}$ are the maximum and minimum values of the objective function $f(X)$ in the current combined population. The penalty value is defined as $p(X) = (1 - r_f)M(X) + r_f N(X)$ where

$$M(X) = \begin{cases} 0\,, & \text{if } r_f = 0 \\ \upsilon(X), & \text{otherwise} \end{cases} \quad (5)$$

$$N(X) = \begin{cases} 0, & \text{if } X \text{ is a feasible individual} \\ f''(X), & \text{if } X \text{ is an infeasible individual.} \end{cases} \quad (6)$$

Therefore, in [9], [31], the selection of individuals in all the three phases is based on a value determined by the overall constraint violation and objective values. Thus, there is a chance for an individual with lower overall constraint violation and higher fitness to get selected over a feasible individual with lower fitness even in Phase 3, where there is sufficient number of feasible solutions to form the parent population using only the feasible solutions.

In [22], an adaptive tradeoff model (ATMES) was proposed. In Phase 1, ATMES uses hierarchical nondominated individual selection scheme to select individuals with less constraint violation. In Phase 2, it selects feasible as well as infeasible ones based on a value determined by overall constraint violation and objective function similar to the one used in self-adaptive penalty. In Phase 3, it selects only feasible individuals similar to the SF.

C. ε-Constraint (EC)

The ε-constraint handling method was proposed in [14] in which the relaxation of the constraints is controlled by using the ε parameter. As solving a constrained optimization problem becomes tedious when active constraints are present, proper control of the ε parameter is essential [14] to obtain high-quality solutions for problems with equality constraints. The ε level is updated until the generation counter k reaches the control generation T_c. After the generation counter exceeds T_c, the ε level is set to zero to obtain solutions with no constraint violation

$$\varepsilon(0) = \upsilon(X_\theta) \quad (7)$$

$$\varepsilon(k) = \begin{cases} \varepsilon(0)\left(1 - \frac{k}{T_c}\right)^{cp}, & 0 < k < T_c \\ 0, \quad k \geq T_c \end{cases} \quad (8)$$

where X_θ is the top θth individual and $\theta = (0.05 * NP)$. The recommended parameter ranges are [14]: $T_c \in [0.1T_{\max}, 0.8T_{\max}]$ and $cp \in [2,\ 10]$.

TABLE I
STOCHASTIC RANKING

If (no constraint violation or rand $< p_f$)
Rank based on the objective value only
else
Rank based on the constraint violation only
End

The selection of individuals in the three phases of evolution by using the ε-constraint technique is similar to the SF, but in the EC, a solution is regarded as feasible if its overall constraint violation is lower than $\varepsilon(k)$.

D. Stochastic Ranking (SR)

Runarsson and Yao [8] introduced SR method to achieve a balance between objective and the overall constraint violation stochastically. A probability factor p_f is used to determine whether the objective function value or the constraint violation value determines the rank of each individual. Basic form of the SR [8] is presented in Table I.

In [21], an improved version of the SR (ISR) was proposed using evolution strategies and differential variation. In SR, comparison between two individuals may be based on objective value alone or overall constraint violation alone as randomly determined. Thus, infeasible solutions with better objective value have a chance to be selected in all three phases of evolution. In this paper, a modified version of the SR presented in [8] is used. Here, the value of p_f is not maintained a constant instead, decreased linearly from $p_f = 0.475$ in the initial generation to $p_f = 0.025$ in the final generation.

E. Multiobjective Constraint Handling

In [4], a multiobjective method with local and global search operators was proposed to solve constrained optimization problems. In this method each constraint is treated as an objective to be optimized. In such methods, an infeasible individual with better objective value and less constraint violation can be selected. A generic framework using genetic algorithms (AGFCOGA) [26] for solving the constrained optimization problem also uses a similar approach, but in Phase 1, the objective function is completely disregarded and the problem is treated as a constraint satisfaction problem. Thus, in Phase 1, only individuals with the least constraint violation is selected. In Phases 2 and 3, the problem is treated as a bi-objective problem with simultaneous optimization of the objective function and the overall constraint violation.

From the above discussions, we can observe that each of the constraint handling method used in ECHT differs in at least one of the three phases. ATMES [22] and SAFF [31] are similar to SP. SMES [19] is similar to superiority of feasible solutions. A multiobjective-based method [4] uses nondomination sorting which is time-consuming. Hence, these methods are not employed in our proposed ECHT. However, it should be noted that the ECHT approach is general and can be formulated with any search method and constraint handling techniques.

III. Ensemble of Constraint Handling Techniques

Each constrained optimization problem would be unique in terms of the ratio between feasible search space and the whole search space, multimodality and the nature of constraint functions. As evolutionary algorithms are stochastic in nature, the evolution paths can be different in every run even when the same problem is solved by using the same algorithm. In other words, the search process passes through different phases at different points during the search process. Therefore, depending on several factors such as the ratio between feasible search space and the whole search space, multimodality of the problem, nature of equality/inequality constraints, the chosen EA, and global exploration/local exploitation stages of the search algorithm, different constraint handling methods can be effective during different stages of the search process. Due to the strong interactions between these diverse factors and the stochastic nature of the evolutionary algorithms, it is not straightforward to determine which constraint handling method is the best during a particular stage of the evolution to solve a given problem using a given EA. Motivated by these observations, we develop the ECHT to implicitly benefit from the match between constraint handling methods, characteristics of the problem being solved, chosen EA and the exploration-exploitation stages of the search process.

A real-world problem can take several minutes to several hours to compute the objective function value [6]. Therefore, finding a better constraint handling method for such problem by trial-and-error may become difficult. The computation time wasted in searching for a better constraint handling method can be saved by using the proposed ensemble method.

In this paper, we present ECHT with four constraint handling techniques discussed in Sections II-A–II-D. Each constraint handling technique has its own population and parameters. Each population corresponding to a constraint handling method produces its offspring and evaluates them. The parent population corresponding to a particular constraint handling method not only competes with its own offspring population but also with offspring population of the other three constraint handling methods. Due to this, an offspring produced by a particular constraint handling method may be rejected by its own population, but could be accepted by the populations of other constraint handling methods. Hence, in ECHT every function call is utilized effectively. If the evaluation of objective/constraint functions is computationally expensive, more constraint handling methods can be included in the ensemble to benefit more from each function call. And if a particular constraint handling technique is best suited for the search method and the problem during a point in the search process, the offspring population produced by the population of that constraint handling method will dominate the other and enter other populations too. In the subsequent generations, these superior offspring will become parents in other populations too. Therefore, ECHT transforms the burden of choosing the best constraint handling technique and tuning the associated parameter values for a particular problem into an advantage. If the constraint handling methods selected to form an ensemble are similar in nature then the populations associated with each of them may lose diversity and the search ability of ECHT may deteriorate. Thus, the performance of ECHT can be improved by selecting constraint handling methods with diverse and competitive nature. The general framework of the ensemble algorithm is illustrated in the flowchart shown in Fig. 1.

As ECHT employs different constraint handling methods each having its own population, it can be compared with hybrid methods like memetic algorithms [32]–[34]. Some methods like island models [35] sometimes called "Migration model" or "Coarse Grained model," also employ subpopulations in their approach. The main difference between the ECHT and the island model is that in island model, subpopulations in different islands evolve separately with occasional communication between them to maintain diversity while in ECHT the communication between different populations is by sharing of all offspring and thus facilitating efficient usage of each function call.

IV. ECHT-EP

EP has been investigated for over four decades [36]–[38]. EP simulates evolution at the phenotypic level emphasizing behavioral evolution rather than genetic evolution. This is one of the major reasons why EP does not use any crossover operator and relies on asexual reproduction. EP commences with a randomly initialized population which generates an offspring population by mutation using either Gaussian (classical EP, CEP) [36], [39] or Levy [40] or Cauchy (fast EP, FEP) [41] or a mixed mutation [42] strategy. Based on the mutation strategy used, different variants of EP such as the CEP [36], FEP, improved FEP (IFEP) [41] and mixed EP (MEP) [42] are available.

In this paper, a modified CEP called adaptive CEP is used as the search algorithm. In adaptive CEP the strategy parameters are adapted based on the performance over the last few generations. The comparison between the CEP and the adaptive CEP in [24] on 13 problems with the same four individual constraint handling methods show that the adaptive CEP performs much better. In adaptive CEP, the η values are randomly initialized scaled to the search range of the parameters. The η values that produce a successful offspring in a generation are collected (s_eta). The η values are updated based on the mean of the s_eta values (ms_eta) of the previous lp generations. After evaluating the offspring population, the parent population for the next generation is selected by performing tournament selection over the union of parents and offspring. In tournament selection, for every member X in the combined parent and offspring population, a specified number of members (tournament size, in this case 20) are randomly selected from the combined population and each one is compared against X to determine how many times X is better than its opponents based on the constraint handling method used. This count will be the winning score of member X. After every member in the combined population completes this competition, one half of the population with the highest wins is selected as the next generation parent population. The ECHT-EP algorithm is presented in Table II.

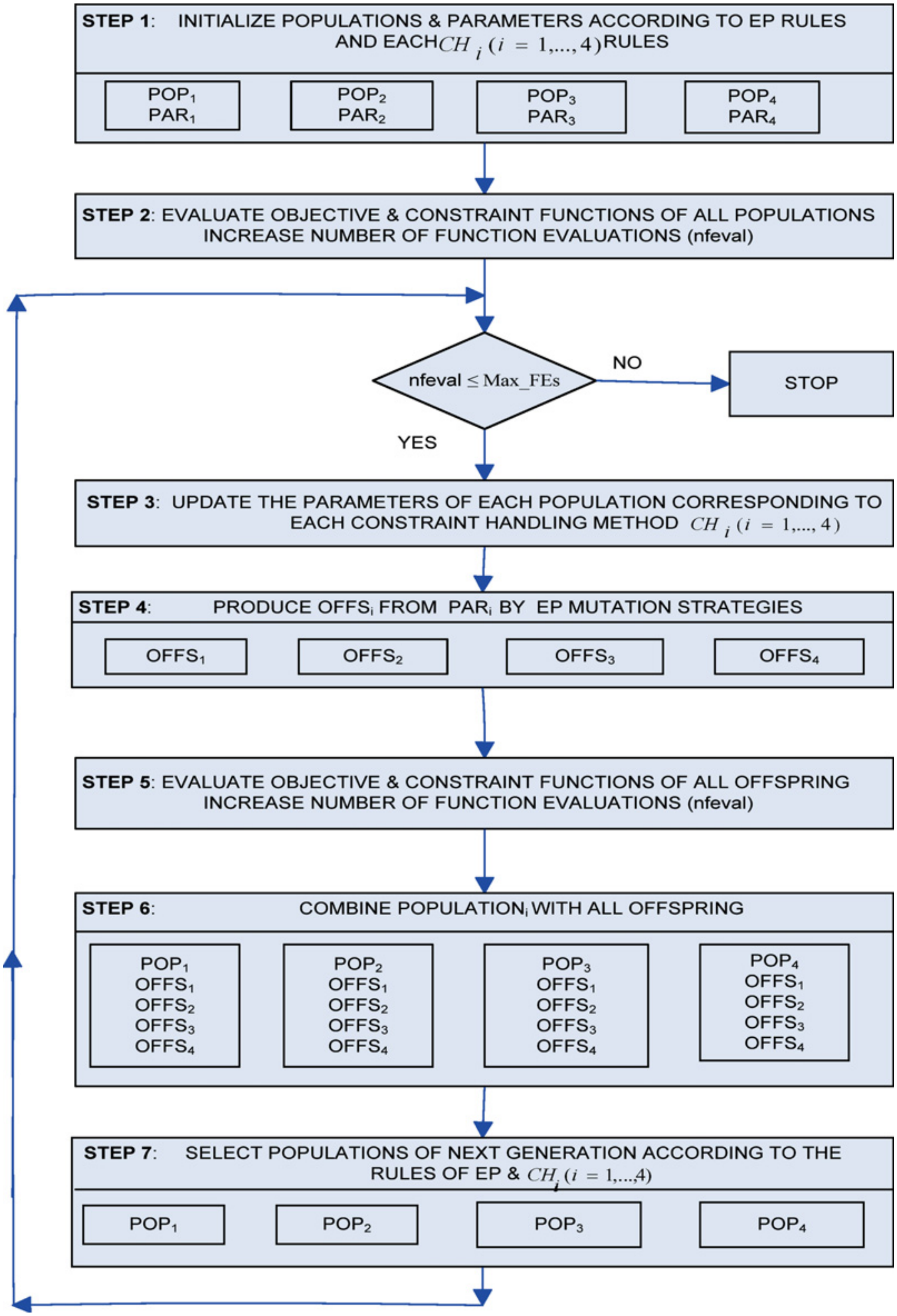


Fig. 1. Flowchart of ECHT. CH: constraint handling method; POP: population; PAR: parameters; OFF: offspring; Max_FEs: maximum number of function evaluations.

TABLE II
ECHT-EP

STEP 1: Each of the four constraint handling techniques (SF, SP, EC, and SR in Section II) has its own population of μ individuals each with dimension n ($\mathrm{POP_i}, i = 1, \ldots, 4$) and parameters $\mathrm{PAR_i}, i = 1, \ldots, 4$ initialized according to the rules of adaptive CEP and the corresponding constraint handling method $\mathrm{CH_i}, i = 1, \ldots, 4$. Set the generation counter $k = 1$. The learning period $lp = 10$. Parameter η is randomly initialized scaled to the search range for every member in each population (instead of a constant value used in [41]) as

$$\eta_l^i = \lambda * rand(1, n). * (X\max - X\min), \quad i = 1, .., 4 \text{ and } l = 1, \ldots, \mu \tag{9}$$

where $X\min$ and $X\max$ are the upper and lower bounds of the search space and λ= 0.8/sqrt(n) is the scaling parameter [22].
STEP 2: Evaluate the objective/constraint function values and the overall constraint violation for each individual $X_l^i, \forall l \in \{1, \ldots, \mu\}$ of every population ($\mathrm{POP_i}, i = 1, \ldots, 4$) using (1)–(3).
STEP 3: The parameter values of constraint handling methods are updated according to Section II. In every generation the η values of the successful offspring are saved in s_eta memory. The η parameter for the next generation is updated based on the mean of s_eta values (ms_eta) of the previous lp generations as

$$\eta_i = ms\_eta + (2 * rand(1, n) - 1) * ms\_eta \tag{10}$$

where ms_eta is the mean of s_eta of previous lp generations.
STEP 4: Each parent population ($\mathrm{POP_i}, i = 1, \ldots, 4$) produces offspring population ($\mathrm{OFFS_i}, i = 1, \ldots, 4$). The best parent in each population X_1^i ($i = 1, .., 4$) produces an offspring using the differential variation equation [21]

$$X_1^{i'} = X_1^i + \gamma(X_{r1}^i - X_{r2}^i) \tag{11}$$

where $r1, r2$ are randomly generated and $r1 \neq r2$. The value of $\gamma = 0.8$.
The rest of the parent population $X_l^i, l = 2, \ldots, \mu$ creates a single offspring $X_l^{i^1}$ by

$$x_l^{i'}(j) = x_l^i(j) + \eta_l^i(j) * N(0, 1) \tag{12}$$

where $x_l^i(j), x_l^{i'}(j)$, and $\eta_l^i(j)$ denote the jth component of the vectors $X_l^i, X_l^{i'}$, and η_l^i, respectively. $N(0, 1)$ denotes a normally distributed 1-D Gaussian random number with zero mean and standard deviation one.
STEP 5: Compute the objective/constraint function values and the overall constraint violation of each offspring $X_l^{i'} \forall l \in \{1, \ldots, \mu\}$. Each offspring retains the objective and constraint function values separately, i.e., each offspring is evaluated only once.
STEP 6: Each parent population $\mathrm{POP_i}, i = 1, \ldots, 4$ is combined with offspring produced by it and the offspring produced by all other populations corresponding to different constraint handling techniques as in **STEP 6** in Fig. 1. The four different groups are:
Group 1: ($\mathrm{POP_1}$,$\mathrm{OFFS_i}, i = 1, \ldots, 4$); **Group 2** ($\mathrm{POP_2}$,$\mathrm{OFFS_i}, i = 1, \ldots, 4$); **Group 3:** ($\mathrm{POP_3}$,$\mathrm{OFFS_i}, i = 1, \ldots, 4$); and **Group 4:** ($\mathrm{POP_4}$,$\mathrm{OFFS_i}, i = 1, \ldots, 4$).
STEP 7: In selection step, parent populations $\mathrm{POP_i}, i = 1, \ldots, 4$ for the next generation are selected from Groups 1, 2, 3, and 4, respectively by applying the tournament selection separately to each Group. The "win" in this step is determined according to individual constraint handling method, i.e., when parent populations $\mathrm{POP_1}$, $\mathrm{POP_2}$, $\mathrm{POP_3}$ and $\mathrm{POP_4}$ are selected from Groups 1, 2, 3 and 4, the "win" is based on SF, SP, EC, and SR, respectively.
STEP 8: Stop if the termination criterion is satisfied. If not, $k = k + 1$ and go to **STEP 3**.

V. ECHT-DE

DE [43] is a population-based stochastic search technique which is a fast and simple technique. Since it was proposed in 1995 [44], it has been successfully applied in diverse fields such as mechanical engineering [45] and signal processing [46]. Depending on the application different variants of DE have been proposed to balance the exploration and exploitation capability [47]. DE has also been recently used to solve constrained optimization problems [7], [10]–[15]. DE starts with an initial population covering the entire search space as much as possible. DE generates a new parameter vector by an operation called "mutation," in which the weighted difference between two population vectors is added to a third vector. The mutated vector's parameters are then mixed with parameters of another predetermined vector, the target vector, to yield the trial vector. Parameter mixing is often referred to as "crossover." If the trial vector yields a lower cost function than the target vector, the trial vector replaces the target vector in the following generation. The replacement of the target vector by the trial vector is called selection. The ECHT-DE algorithm is presented in Table III.

VI. Experimental Results

Each of the four constraint handling methods present in Section II (SF, SP, SR, and EC) are evaluated using both DE (SF-DE, SP-DE, SR-DE, and EC-DE) and adaptive CEP (SF-EP, SP-EP, SR-EP, and EC-EP) as the basic search algorithms. The two instantiations of ECHT with DE (ECHT-DE) and EP (ECHT-EP) are also evaluated.

1) *Experiment 1:* First we evaluate the performance of four single constraint handling methods (SF-EP, SP-EP, SR-EP, and EC-EP) and ECHT-EP on 24 well-defined problems of CEC 2006 [48]. The performance is compared with some of the state-of-the-art methods. The characteristics of the problems are presented in Table IV.
2) *Experiment 2:* From the experimental results presented in [11], [13], [14], it can be observed that most of the problems of CEC2006 are solved by algorithms using DE as basic search algorithm within 500 000 function evaluations. Thus, the effectiveness of ECHT-DE cannot be shown on this problem set. Hence, to differentiate the performance, we designed 13 harder problems. The characteristics of the problems are presented in Table V and the problem definitions are presented in Appendix A.

All the algorithms are run 30 times on each test problem with maximum number of function evaluations (Max_FEs) in beach run set to 240 000 in both experiments. From the experimental results, we observed that the single constraint methods perform better when the population sizes are 200 (maximum generations 1200) with EP and 50 (maximum

TABLE III

ECHT-DE

STEP 1: Each of the four constraint handling techniques (SF, SP, EC, and SR in Section II) has its own population of μ individuals each with dimension n ($\mathrm{POP_i}$, $i = 1, \ldots, 4$) and parameters $\mathrm{PAR_i}$, $i = 1, \ldots, 4$ initialized according to the rules DE and the corresponding constraint handling method $\mathrm{CH_i}$, $i = 1, \ldots, 4$. Set the generation counter $k = 1$.
STEP 2: Evaluate the objective/constraint function values and the overall constraint violation for each individual $X_l^i, \forall l \in \{1, \ldots, \mu\}$ of every population ($\mathrm{POP_i}$, $i = 1, \ldots, 4$) using (1)–(3).
STEP 3: The parameter values of constraint handling methods are updated according to Section II.
STEP 4: Each parent population ($\mathrm{POP_i}$, $i = 1, \ldots, 4$) produces offspring population ($\mathrm{OFFS_i}$, $i = 1, \ldots, 4$) by mutation and crossover. The F and CR values are taken as 0.9 and 0.7, respectively [14].

$$\mathrm{V}^i_{l,G} = \mathrm{X}^{\mathrm{i}}_{\mathrm{r}^l_1,G} + F.(\mathrm{X}^{\mathrm{i}}_{\mathrm{r}^l_2,G} - \mathrm{X}^{\mathrm{i}}_{\mathrm{r}^l_3,G}) \tag{13}$$

$$u^i_{l,G}(j) = \begin{cases} \mathrm{V}^{\mathrm{i}}_{\mathrm{l,G}}(j) & \text{if}(rand_j[0,1) \text{ or } (j = j_{rand}) \\ \mathrm{X}^{\mathrm{i}}_{\mathrm{l,G}}(j) & \end{cases} \qquad j = 1,\ 2,\ \ldots,\ n \tag{14}$$

where $V^{\mathrm{i}}_{\mathrm{l,G}} and u^i_{l,G}$ are the mutant and trial vectors corresponding to lth individual of population POPi, $i = 1, \ldots, 4$ in generation G.
STEP 5: Compute the objective/constraint function values and the overall constraint violation of each offspring $X_l^{i'} \forall l \in \{1, \ldots, \mu\}$. Each offspring retains the objective and constraint function values separately, i.e., each offspring is evaluated only once.
STEP 6: Each parent population $\mathrm{POP_i}$, $i = 1, \ldots, 4$ is combined with offspring produced by it and the offspring produced by all other populations corresponding to different constraint handling techniques as in **STEP 6** in Fig. 1. The four different groups are:
Group 1: ($\mathrm{POP_1}$, $\mathrm{OFFS_i}$, $i = 1, \ldots, 4$); **Group 2:** ($\mathrm{POP_2}$, $\mathrm{OFFS_i}$, $i = 1, \ldots, 4$); **Group 3:** ($\mathrm{POP_3}$, $\mathrm{OFFS_i}$, $i = 1, \ldots, 4$); and **Group 4:** ($\mathrm{POP_4}$,$\mathrm{OFFS_i}$, $i = 1, \ldots, 4$).
STEP 7: In selection step, parent populations $\mathrm{POP_i}$, $i = 1, \ldots, 4$ for the next generation are selected from Groups 1, 2, 3, and 4, respectively. In a Group (say Group 1), since OFF1 is produced by POP1 by mutation and crossover, DE's selection based on competition between parent and its offspring is employed when POP1 competes with OFF1. But when POP1 competes with OFF2 or OFF3 or OFF4, produced by other populations, each member in POP1 competes with a randomly selected offspring from OFF2 or OFF3 or OFF4.
STEP 8: Stop if the termination criterion is satisfied. If not, $k = k + 1$ and go to **STEP 3**.

generations 4800) with DE. In both ECHT-EP and ECHT-DE, the population corresponding to each constraint handling method is set to 50 (maximum generations 1200). In order to observe the effect of ECHT, we make sure that all the components such as initialization of parameters, parent selection, and offspring generation are the same in the four single constraint handling techniques and ECHT. The parameters corresponding to the constraint handling methods are set to: $T_c = 0.2T_{\max}$, $c_p = 5$ and p_f is linearly decreased from an initial value of 0.475–0.025 in the final generation. However, the performance of the ECHT can be improved by tuning the parameters of individual constraint handling methods. To illustrate this, experiments were performed with ECHT-EP using the same parameters as in single constraint handling methods (ECHT-EP1) and with tuned parameters (ECHT-EP2). The tuned parameters in ECHT-EP2 are $T_c = 0.8T_{\max}$ and $c_p = 2$. The tolerance parameter δ for the equality constraints is adapted by using the following expression:

$$\delta(t+1) = \frac{\delta(t)}{\hat{\delta}}. \tag{15}$$

The initial $\delta(0)$ is selected as the median of equality constraint violations over the entire initial population. The value of $\hat{\delta}$ is selected in such a way that it causes δ to reach a value of E−04 at around 600 generations, after which the value of δ is fixed at E−04. The best mean (Mn) values are presented in boldface in Tables VI, VIII, and IX.

A. Experiment 1

1) *Comparison Among Single Constraint Handling Methods:* First we compare the four single constraint handling methods to highlight that their performances vary substantially among the 24 CEC 2006 problems. The results are presented in Table VI. All four constraint handling methods could find optimal values in problems G01, G03, G04, G08, G09, G11, G12, G13, G15, G16, G18 and G24. In G02, the performance of all the algorithms marginally varies. Performance of the SF and SR algorithms are consistent and could find feasible solutions for all problems except G20 and G22, while SP and EC perform well on some problems and fail badly on some problems. SP and EC fail to find feasible solutions in all the runs for G06. The better performance of SP can be observed in G10, G14, while EC perform much better in G05 and G23.

To compare the overall performance, the algorithms in Table VI are ranked based on the mean values and the ranks are indicated in braces for each problem. The overall ranks for SF, SP, EC, and SR are 57, 60, 54, and 60, respectively. The minimum rank that any algorithm could achieve by becoming the best in all problems is 22. The average ranks for SF, SP, EC, and SR are 2.59, 2.73, 2.45, and 3.09, respectively. From the ranking results, it is obvious that the performance of single constraint handling methods vary substantially. From the ranks, it is clear that no single constraint handling method is superior to other single constraint handling methods in every problem as their ranks are low and approximately similar. This is apparently expected due to the NFL theorem.

2) *Comparison of ECHT-EP with Single Constraint Handling Methods:* From Table VI, it can be observed that the ECHT-EP could find feasible solutions in all the problems except G20 and G22, where all single constraint handling methods also fail to find feasible solutions. From ranking shown it can be observed that ensemble (ECHT-EP1, ECHT-EP2) are ranked above all the single constraint handling methods with total and average rankings of (36, 22) and (1.64,

TABLE IV
PROPERTIES OF CEC2006 TEST PROBLEMS

Function	n	Type of f	ρ (%)	LI	NI	LE	NE	a
G01	13	Quadratic	0.0111	9	0	0	0	6
G02	20	Nonlinear	99.8474	0	2	0	0	1
G03	10	Polynomial	0.0000	0	0	0	1	1
G04	5	Quadratic	52.1230	0	6	0	0	2
G05	4	Cubic	0.0000	2	0	0	3	3
G06	2	Cubic	0.0066	0	2	0	0	2
G07	10	Quadratic	0.0003	3	5	0	0	6
G08	2	Nonlinear	0.8560	0	2	0	0	0
G09	7	Polynomial	0.5121	0	4	0	0	2
G10	8	Linear	0.0010	3	3	0	0	6
G11	2	Quadratic	0.0000	0	0	0	1	1
G12	3	Quadratic	4.7713	0	1	0	0	0
G13	5	Nonlinear	0.0000	0	0	0	3	3
G14	10	Nonlinear	0.0000	0	0	3	0	3
G15	3	Quadratic	0.0000	0	0	1	1	2
G16	5	Nonlinear	0.0204	4	34	0	0	4
G17	6	Nonlinear	0.0000	0	0	0	4	4
G18	9	Quadratic	0.0000	0	12	0	0	4
G19	15	Nonlinear	33.4761	0	5	0	0	–
G20	24	Linear	0.0000	0	6	2	12	–
G21	7	Linear	0.0000	0	1	0	5	6
G22	22	Linear	0.0000	0	1	8	11	–
G23	9	Linear	0.0000	0	2	3	1	–
G24	2	Linear	79.6556	0	2	0	0	2

n is the number of decision variables, r=$|F|/|S|$ is the estimated ratio between the feasible region and the search space, LI and NI represent the number of linear and nonlinear inequality constraints, LE and NE represent the number of linear and nonlinear equality constraints, and a is the number of active constraints on the optima. Tolerance δ = 0.0001 for LE and NE.

TABLE V
PROPERTIES OF 13 NEWLY DESIGNED PROBLEMS

Function	n	I	E	Range	ρ
H01	10	1	2	[−50, 50]	0.000000
H02	10	2	1	[−5.12, 5.12]	0.000000
H03	10	1	1	[−100, 100]	0.000000
H04	10	0	2	[−100, 100]	0.000000
H05	10	2	0	[−100, 100]	0.008900
H06	10	1	0	[−100, 100]	0.000000
H07	10	2	1	[−100, 100]	0.000000
H08	10	0	1	[−500, 500]	0.000000
H09	10	3	0	[−500, 500]	0.000001
H10	10	1	0	[−10, 10]	0.500300
H11	10	0	2	[−5, 5]	0.000000
H12	10	0	1	[−50, 50]	0.000000
H13	10	2	0	[−100, 100]	0.259900

n is the number of decision variables, r=$|F|/|S|$ is the estimated ratio between the feasible region and the search space, I and E represent the number of inequality and equality constraints. Tolerance δ = 0.0001 for E.

1), respectively, while the average ranking of SF, SP, EC, and SR are 2.59, 2.73, 2.45 and 3.09, respectively. ECHT-EP2 achieves the best possible rankings of 22 and 1 by becoming the best in all problems. We can also observe that by tuning the parameters of the EC method in the ensemble ECHT-EP2, its performance is improved.

To statistically compare the performance of ensemble with the four single constraint handling methods, t-test results (h values) are presented in Table VI, where each row represents the t-test results for a particular problem for single constraint handling methods against the ensemble (ECHT-EP1). To allow fair comparison, we compare the single cases with only ECHT-EP1, which uses the same parameters as that of the single cases. Numerical values -1, 0, and 1 represent that the ECHT-EP1 is inferior to, equal to, and superior to the SF, SP, EC, and SR constraint handling method, respectively. From the t-test results, we can observe that single constraint handling methods are superior to, equal to and worse than the ECHT-EP1 in 0, 63, and 25 cases, respectively out of the total 88 cases. Thus, the ECHT-EP1 is always either better or equal.

The superior performance of ensemble over the single constraint handling methods within the given maximum function

TABLE VI
COMPARISON OF SINGLE CONSTRAINT HANDLING METHODS AND ECHT

Function and Optimal Value		SF-EP	SP-EP	EC-EP	SR-EP	ECHT-EP1	ECHT-EP2
G01 −15.0000	b	−14.9999	−15.0000	−14.9998	−14.9996	−15.0000	−15.0000
	Md	−14.9999	−14.9999	−14.9996	−14.9991	−15.0000	−15.0000
	Mn	−14.9999(2)	−14.9999(2)	−14.9996(4)	−14.9991(5)	**−15.0000(1)**	**−15.0000(1)**
	w	−14.9998	−14.9999	−14.9995	−14.9985	−15.0000	−15.0000
	sd	1.85E−05	1.26E−05	9.56E−05	2.69E−04	0.00E+00	0.00E+00
	FR	100%	100%	100%	100%	100%	100%
	h	1	0	1	1	–	–
G02 −0.8036191	b	−0.8036085	−0.8035886	−0.8036086	−0.8036068	−0.8036191	−0.8036191
	Md	−0.8035883	−0.7948842	−0.7992277	−0.7936123	−0.8033239	−0.8033239
	Mn	−0.7994479(2)	−0.7960742(4)	−0.7987781(3)	−0.7924832(5)	−0.7998220(1)	**−0.7998220(1)**
	w	−0.7925907	−0.7852561	−0.7925929	−0.7782940	−0.7851820	−0.7851820
	sd	5.38E−03	5.92E−03	5.13E−03	8.16E−03	6.29E−03	6.29E−03
	FR	100%	100%	100%	100%	100%	100%
	h	0	0	0	0	–	–
G03 −1.0005	b	−1.0005	−1.0005	−1.0005	−1.0005	−1.0005	−1.0005
	Md	−1.0005	−1.0005	−1.0005	−1.0005	−1.0005	−1.0005
	Mn	−1.0005(1)	−1.0005(1)	−1.0005(1)	−1.0005(1)	−1.0005(1)	−1.0005(1)
	w	−1.0005	−1.0005	−1.0005	−1.0005	−1.0005	−1.0005
	sd	0.0E+00	0.0E+00	0.0E+00	0.0E+00	0.0E+00	0.0E+00
	FR	100%	100%	100%	100%	100%	100%
	h	0	0	0	0	–	–
G04 −30 665.5387	b	−30 665.5387	−30 665.5387	−30 665.5387	−30 665.5387	−30 665.5387	−30 665.5387
	Md	−30 665.5387	−30 665.5387	−30 665.5387	−30 665.5387	−30 665.5387	−30 665.5387
	Mn	−30 665.5387(1)	−30 665.5387(1)	−30 665.5387(1)	−30 665.5387(1)	−30 665.5387(1)	−30 665.5387(1)
	w	−30 665.5387	−30 665.5387	−30 665.5387	−30 665.5387	−30 665.5387	−30 665.5387
	sd	0.0E+00	0.0E+00	0.0E+00	0.0E+00	0.0E+00	0.0E+00
	FR	100%	100%	100%	100%	100%	100%
	h	0	0	0	0	–	–
G05 5126.4967	b	5126.4969	5126.4967	5126.4967	5126.4969	5126.4967	5126.4967
	Md	5131.7147	5126.5211	5126.4968	5130.0131	5126.4967	5126.4967
	Mn	5161.5388(5)	5127.7182(3)	5126.5058(2)	5158.3317(4)	**5126.4967(1)**	**5126.4967(1)**
	w	5485.1800	5134.6751	5126.6048	5329.3866	5126.4972	5126.4967
	sd	7.8E+01	2.3E+00	2.2E−02	6.0E+01	0.0E+00	0.0E+00
	FR	100%	100%	100%	100%	100%	100%
	h	1	1	1	1	–	
G06 −6961.8139	b	−6961.8139	−6961.8139	−6961.8139	−6961.8139	−6961.8139	−6961.8139
	Md	−6961.8139	−6961.8139	−6961.8139	−6961.8139	−6961.8139	−6961.8139
	Mn	−6961.8139(3)	−6961.8139(4)	−6961.8139(5)	−6961.8139(1)	−6961.8139(1)	−6961.8139(1)
	w	−6961.8139	−6961.8139	−6961.8139	−6961.8139	−6961.8139	−6961.8139
	sd	0.00E+00	0.0E+00	0.0E+00	0.0E+00	0.00E+00	0.00E+00
	FR	100%	60%	27%	100%	100%	100%
	h	1	1	1	1	–	–
G07 24.3062	b	24.3064	24.3062	24.3066	24.3063	24.3063	24.3062
	Md	24.3081	24.3066	24.3082	24.3079	24.3078	24.3063
	Mn	24.3093(4)	24.3071(2)	24.3096(50	24.3097(6)	24.3090(3)	**24.3063(1)**
	w	24.3199	24.3120	24.3290	24.3253	24.3166	24.3063
	sd	3.1E−03	1.4E−03	1.0E−04	4.7E−03	3.0E−03	3.19E−05
	FR	100%	27%	100%	100%	100%	100%
	h	0	0	0	0	–	–
G08 −0.09582504	b	−0.09582504	−0.09582504	−0.09582504	−0.09582504	−0.09582504	−0.09582504
	Md	−0.09582504	−0.09582504	−0.09582504	−0.09582504	−0.09582504	−0.09582504
	Mn	−0.09582504(1)	−0.09582504(1)	−0.09582504(1)	−0.09582504(1)	−0.09582504(1)	−0.09582504(1)
	w	−0.09582504	−0.09582504	−0.09582504	−0.09582504	−0.09582504	−0.09582504
	sd	0.0E+000	2.5E−17	0.0E+000	0.0E+00	0.0E+00	0.0E+00
	FR	100%	100%	100%	100%	100%	100%
	h	0	0	0	0	–	–

TABLE VI
(CONTINUED)

G09 680.630057	b	680.630057	680.630060	680.630057	680.630058	680.6300057	680.630057
	Md	680.630057	680.630614	680.630057	680.630059	680.6300057	680.630057
	Mn	680.630057(1)	680.630614(5)	680.630057(1)	680.630059(4)	680.630057(1)	680.630057(1)
	w	680.630057	680.631032	680.630057	680.630062	680.630060	680.630057
	sd	3.61E−08	1.0E−04	1.32E−08	1.30E−06	2.0E−04	2.61E−08
	FR	100%	100%	100%	100%	100%	100%
	h	0	0	0	0	–	–
G10 7049.2480	b	7054.1548	7049.2566	7051.2552	7055.6236	7049.2487	7049.2483
	Md	7118.6628	7050.4251	7096.0003	7138.4684	7049.3456	7049.2488
	Mn	7123.6326(5)	7051.6584(3)	7094.3033(4)	7155.6031(6)	**7049.4342(2)**	**7049.2490(1)**
	w	7242.9482	7080.8116	7213.3871	7428.5722	7050.3902	7049.2501
	sd	5.3E+01	5.7E+00	3.75E+01	80.E+01	2.00E−01	6.60E−04
	FR	100%	100%	100%	100%	100%	100%
	h	1	1	1	1	–	–
G11 0.7499	b	0.7499	0.7499	0.7499	0.7499	0.7499	0.7499
	Md	0.7499	0.7499	0.7499	0.7499	0.7499	0.7499
	Mn	0.7499(1)	0.7499(1)	0.7499(1)	0.7499(1)	0.7499(1)	0.7499(1)
	w	0.7499	0.7499	0.7499	0.7499	0.7499	0.7499
	sd	0.0E+00	0.0E+00	0.0E+00	0.0E+00	0.0E+00	0.0E+00
	FR	100%	100%	100%	100%	100%	100%
	h	0	0	0	0	–	–
G12 −1.000	b	−1.0000	−1.0000	−1.0000	−1.0000	−1.0000	−1.0000
	Md	−1.0000	−1.0000	−1.0000	−1.0000	−1.0000	−1.0000
	Mn	−1.0000(1)	−1.0000(1)	−1.0000(1)	−1.0000(1)	−1.0000(1)	−1.0000(1)
	w	−1.0000	−1.0000	−1.0000	−1.0000	−1.0000	−1.0000
	sd	0.0E+00	0.0E+00	0.0E+00	0.0E+00	0.0E+00	0.0E+00
	FR	100%	100%	100%	100%	100%	100%
	h	0	0	0	0	–	–
G13 0.053941514	b	0.053941514	0.053941514	0.053941514	0.053941514	0.053941514	0.053941514
	Md	0.053941514	0.053941514	0.053941514	0.053941514	0.053941514	0.053941514
	Mn	0.053941514(1)	0.053941514(1)	0.053941514(1)	0.053941514(1)	0.053941514(1)	0.053941514(1)
	w	0.053941514	0.053941514	0.053941514	0.053941514	0.053941514	0.053941514
	sd	5.61E−13	1.33E−13	5.33E−13	7.50E−12	6.50E−12	1.00E−12
	FR	100%	100%	100%	100%	100%	100%
	h	0	0	0	0	–	–
G14 −47.7649	b	−46.7488	−47.7582	−47.2746	−47.0015	−47.7545	−47.7649
	Md	−45.3623	−47.6731	−46.5184	−45.0535	−47.6710	−47.7648
	Mn	−45.3994(5)	**−47.6078(3)**	−46.5188(4)	−45.1402(6)	**−47.6106(2)**	**−47.7648(1)**
	w	−43.7368	−47.0345	−45.4870	−43.2758	−46.4976	−47.7648
	sd	7.9E−01	1.7E−01	4.7E−01	8.5E−01	2.3E−01	2.72E−05
	FR	100%	100%	100%	100%	100%	100%
	h	1	0	1	1	–	–
G15 961.715022	b	961.715022	961.715022	961.715022	961.715022	961.715022	961.715022
	Md	961.715022	961.715022	961.715022	961.715022	961.715022	961.715022
	Mn	961.715022(1)	961.715022(1)	961.715022(1)	961.715022(1)	961.715022(1)	961.715022(1)
	w	961.715022	961.715022	961.715022	961.715022	961.715022	961.715022
	sd	2.40E−13	2.40E−13	2.40E−13	2.11E−13	2.12E−13	2.01E−13
	FR	100%	100%	100%	100%	100%	100%
	h	0	0	0	0	–	–
G16 −1.905155	b	−1.905155	−1.905155	−1.905155	−1.905155	−1.905155	−1.905155
	Md	−1.905155	−1.905155	−1.905155	−1.905155	−1.905155	−1.905155
	Mn	−1.905155(1)	−1.905155(1)	−1.905155(1)	−1.905155(1)	−1.905155(1)	−1.905155(1)
	w	−1.905155	−1.905155	−1.9051548	−1.905155	−1.905155	−1.905155
	sd	7.64E−13	5.75E−09	1.63E−07	2.13E−09	1.12E−10	1.12E−10
	FR	100%	100%	100%	100%	100%	100%
	h	0	0	0	0	–	–

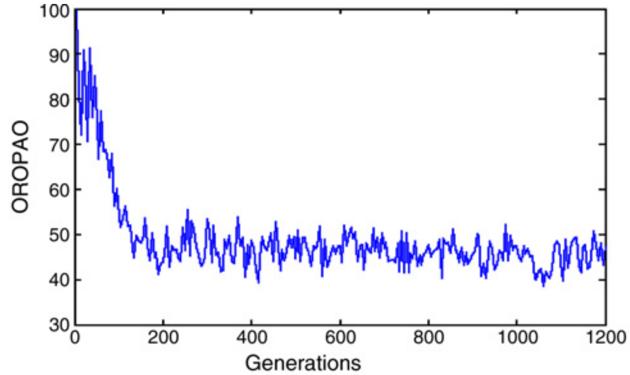
TABLE VI
(CONTINUED)

G17 8853.5397	b	8860.8369	8872.8260	8853.5603	8871.0573	8853.5397	8853.5397
	Md	8944.8365	8918.0986	8853.7422	8944.2003	8853.5479	8853.5397
	Mn	8942.0714(6)	8916.7643(4)	**8853.8194(2)**	8935.0337(5)	**8854.6729(3)**	**8853.5397(1)**
	w	8989.0609	8962.8508	8854.2410	9276.6484	8865.4003	8853.5397
	sd	3.0E+01	2.8E+01	1.9E−01	7.6E+01	2.5E+00	2.13E−08
	FR	100%	100%	100%	100%	100%	100%
	h	1	1	0	1	–	–
G18 −0.86602540	b	−0.86602531	−0.86608306	−0.86602398	−0.86602199	−0.86602541	−0.86602541
	Md	−0.86602514	−0.86561953	−0.86602046	−0.86601599	−0.86602514	−0.86602541
	Mn	−0.86602509(3)	−0.86478625(6)	−0.86601572(4)	−0.86601282(5)	−0.86602510(2)	−0.86602541(1)
	w	−0.86602472	−0.85891161	−0.86597133	−0.86598332	−0.8660252	−0.86602542
	sd	1.71E−07	2.22E−03	1.58E−05	1.10E−05	1.22E−08	1.00E−09
	FR	100%	100%	100%	100%	100%	100%
	h	0	0	0	0	–	–
G19 32.6556	b	32.8928	32.7005	32.8066	32.7902	32.6684	32.6591
	Md	33.2512	32.7981	33.2304	33.3080	32.6846	32.6611
	Mn	33.2849(5)	32.8334(3)	33.2224(4)	33.3392(6)	**32.6969(2)**	**32.6623(1)**
	w	33.9365	33.1570	34.0538	33.7422	32.8139	32.6687
	sd	2.5E−01	1.4E−04	2.7E−01	2.7E−01	4.23E−02	3.4E−03
	FR	100%	100%	100%	100%	100%	100%
	h	1	1	1	1	–	–
G21 193.7245	b	193.7491	195.3702	193.7290	193.7300	193.7245	193.7246
	Md	198.7857	232.4343	240.1518	193.7486	196.3287	193.7399
	Mn	225.4789(4)	241.7966(6)	234.7433(5)	206.1255(2)	215.4050(3)	**193.7438(1)**
	w	294.1277	330.4669	275.8960	261.9906	273.1474	193.7741
	sd	4.10E+01	3.9E+01	2.66E+01	2.63E+01	3.02E+01	1.65E−02
	FR	100%	100%	100%	100%	100%	100%
	h	0	0	0	0	–	–
G23 −400.0551	b	−380.3077	−324.5844	−384.9543	−371.1396	−396.68211	−398.9731
	Md	−348.8464	−253.8178	−348.3227	−345.3168	−331.41637	−385.5657
	Mn	−346.5745(3)	−244.5554(6)	**−347.2257(2)**	−339.6890(4)	−327.01474(5)	**−373.2178(1)**
	w	−306.4070	−137.5764	−289.2883	−312.6974	−217.54773	−335.1145
	sd	2.60E+01	5.55E+01	2.3E+01	1.80E+01	4.5E+01	3.37E+01
	FR	100%	100%	100%	100%	100%	100%
	h	0	0	0	0	–	–
G24 −5.5080	b	−5.5080	−5.5080	−5.5080	−5.5080	−5.5080	−5.5080
	m	−5.5080	−5.5080	−5.5080	−5.5080	−5.5080	−5.5080
	Md	−5.5080(1)	−5.5080(1)	−5.5080(1)	−5.5080(1)	−5.5080(1)	−5.5080(1)
	Mn	−5.5080	−5.5080	−5.5080	−5.5080	−5.5080	−5.5080
	sd	9.36E−16	1.18E−13	9.36E−16	1.73E−15	1.8E−15	1.8E−15
	FR	100%	100%	100%	100%	100%	100%
	h	0	0	0	0	–	–
RANK		57	60	54	68	36	22

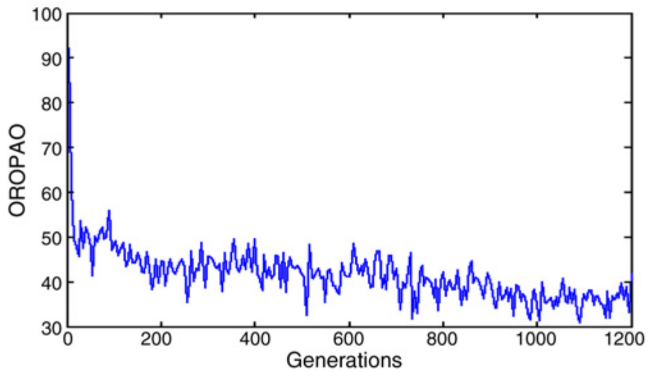
b, Md, Mn, w, sd, and FR represent best, median, mean, worst, standard deviation, and feasibility rate, respectively. Parentheses ( ) contain ranking.

evaluations is due to the efficient use of every function call by all four populations and the search process in the ensemble benefiting from the better performance of different constraint handling methods during different stages of the search process. In every generation every population competes not only with its offspring population but also competes with the offspring solutions produced by the populations of other constraint handling methods. Therefore, if a population generates good offspring at a particular generation, 100% credit cannot be assigned to that particular population because parent of this good offspring could have been produced by some other population in the previous generations. In addition, evolution paths can be different for different populations and cannot be tracked due to the tight coupling between the populations of different constraint handling methods. Thus, it would be difficult to provide evidences such as which constraint handling method is performing better during which stage of the search process. The feasible comparison would be between the final results yielded by the ECHT-EP1/ECHT-EP2 and individual constraint handling methods as done in Table VI.

The evidence for the efficient use of every function call in the ensemble is presented in Fig. 2. In Fig. 2, the horizontal axis represent the generation, while the vertical axis represents the number of offspring population members rejected by their own populations, but are accepted by other populations. From Fig. 2, it is clear that during every generation, a substantial

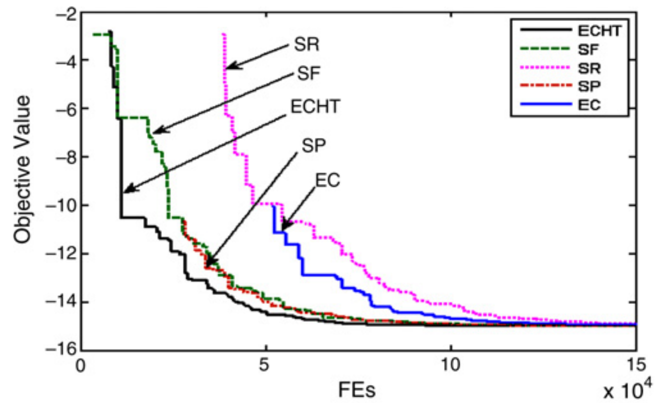


Problem G01

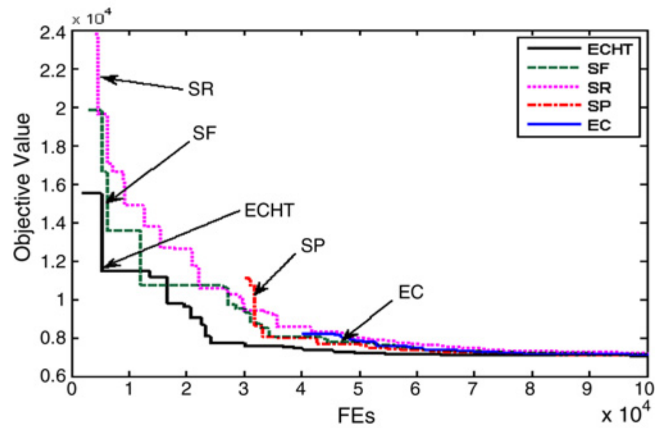


Problem G10

Fig. 2. Efficient use of function calls in the ensemble. OROPAO: offspring rejected by own population and accepted by others.



Problem G01



Problem G10

Fig. 3. Convergence plot of median runs.

portion of the offspring population produced and rejected by one population is accepted by other populations to become parents in the next generation. Due to the diverse nature of the constraint handling methods present in ECHT premature convergence can be avoided and consistency can be improved. Even though there is no learning involved in the ECHT (*lp* is only for adaptive CEP parameters and independent of the constraint handling methods used), due to efficient use of function calls, the convergence properties of the ECHT are better than the single constraint handling methods as shown in Fig. 3.

3) *Comparisons Between Various Related Algorithms:* The characteristics of the state-of-the-art methods are summarized in Table VII. AGFCOGA and SAFF present results only for 11 problems. SAFF uses 1 400 000 FEs and could produce only 17 and 9 feasible runs for G10 and G05, respectively out of the 20 runs conducted. In AGFCOGA, the tolerance δ on the equality constraint satisfaction is much relaxed. The FEs used by (RCGA + SP) is also much larger than 240 000 and the results are inferior to those included in the comparative study. Thus, (RCGA + SP), SAFF, and AGFCOGA are not included in the comparison.

Table VIII compares the results obtained by various state-of-the art algorithms using ES, such as (ES + SR), SMES, ATMES, multiobjective [4], and improved stochastic ranking (ISR) [21]. Evolution strategy with stochastic ranking (ES + SR) [8] had only six feasible runs out of 30 runs performed on the test function G10. The remaining algorithms were able to find feasible solutions in every run in all test functions reported. The ensemble approach (ECHT-EP2) could perform as well as or in most cases better than other algorithms. The algorithms are ranked based on the best (b), median (Md), mean (Mn), and worst (w) values. Ranking is not given for problems G01, G03, G04, G08, G11, and G12, since all algorithms perform equally well. The minimum rank that any algorithm could achieve is 28. The overall rank for each algorithm is presented in the last row of Table VIII. Thus, the average rank for (ES + SR), SMES, ATMES, multiobjective, ISR, and ECHT2 are 4.75, 5.43, 3.36, 2.36, 1.71, and 1.06, respectively. From the ranking, it is obvious that the performance of ECHT-EP2 is superior to the five state-of-the-art algorithms.

B. Experiment 2

The performance of four constraint handling methods (SF, SP, SR, and EC) and their ensemble with two search algorithms DE (SF-DE, SP-DE, SR-DE, EC-DE, and ECHT-DE) and Adaptive CEP (SF-EP, SP-EP, SR-EP, EC-EP, and ECHT-EP) are evaluated using the 13 problems presented in Table VI and in Appendix A. The t-test results are also presented. In t-test SF-DE, SP-DE, SR-DE, and EC-DE are compared with ECHT-DE, while SF-EP, SP-EP, SR-EP, and EC-EP are compared with ECHT-EP.

From the t-test results in Table IX, it can be observed that ECHT-DE and ECHT-EP perform better than the

TABLE VII

CHARACTERISTICS OF STATE-OF-THE-ART CONSTRAINT HANDLING TECHNIQUES

Algorithm	Search Method	Runs	Gen	Max_FEs	Prob.	δ
(ES + SR) [8]	(30 200)-ES	30	1750	350 000	13	0.0001
SAFF [31]	GA with gray coding	20	20 000	1 400 000	11	0.0001
AGFCOGA [26]	Real-coded GA	50	5000	50 000	11	0.001
(RCGA + SP) [9]	Real-coded GA	50	5000	500 000	13	0.0001
SMES [19]	(100 300)-ES	30	800	240 000	13	Decreases from 1.0E−03 to 4E−04
ATMES [22]	(50 300)-ES	30	800	240 000	13	Decreases from 3.0 to 5E−06
Multiobjective[4]	Hybrid EA, NP-250	30	–	240 000	13	1E−07
ECHT-EP, SF-EP, SP-EP, EC-EP, SR-EP	EP, NP-200	30	1200	240 000	37	Decreases from 3.0 to 1.0E−04
ECHT-DE	DE, NP-200	30	1200	240 000	13	Decreases from 3.0 to 1.0E−04
SF-DE, SP-DE, SR-DE, EC-DE	DE, NP-50	30	4800	240 000	13	Decreases from 3.0 to 1.0E−04

Max_FEs, Prob, and δ represent maximum amount of function evaluations, number of problems considered, and relaxation for equality constraint, respectively.

TABLE VIII

COMPARISON BETWEEN VARIOUS STATE-OF-THE-ART METHODS

Fcn & Optimal value		(ES + SR) [8]	SMES [19]	ATMES [22]	Multiobjective [4]	ISR [21]	ECHT-EP2
G01 −15.0000	b	−15.0000	−15.0000	−15.0000	−15.0000	−15.0000	−15.0000
	Md	−15.0000	−15.0000	−15.0000	−15.0000	−15.0000	−15.0000
	Mn	−15.0000	−15.0000	−15.0000	−15.0000	−15.0000	−15.0000
	w	−15.0000	−15.0000	−15.0000	−14.9999	−15.0000	−15.0000
	sd	0.0E+00	0.00E+00	1.6E14	4.297E−07	5.80E−14	0.00E+00
	FR	100%	100%	100%	100%	100%	100%
G02 −0.803619	b	−0.803515(5)	−0.803601(3)	−0.803339(6)	−0.803550(4)	−0.803619(1)	−0.8036191(1)
	Md	−0.785800(6)	−0.792549(4)	−0.792420(5)	−0.794591(2)	−0.793082(3)	−0.8033239(1)
	Mn	−0.781975(6)	−0.785238(4)	−0.790148(3)	−0.792610(2)	−0.782715(5)	**−0.7998220(1)**
	w	−0.726288(5)	−0.751322(4)	−0.756986(3)	−0.756938(2)	−0.723591(6)	−0.7851820(1)
	sd	2.0E−02	1.67E−02	1.3E−02	1.0E−02	2.20E−02	6.29E−03
	FR	100%	100%	100%	100%	100%	100%
G03 −1.0005	b	−1.000	−1.000	−1.000	−1.000	−1.001	−1.0005
	Md	−1.000	−1.000	−1.000	−1.000	−1.001	−1.0005
	Mn	−1.000	−1.000	−1.000	−1.000	−1.001	−1.0005
	w	−1.000	−1.000	−1.000	−1.000	−1.001	−1.0005
	sd	1.9E−04	2.09E−04	5.9E−05	1.304E−12	8.20E−09	0.0E+00
	FR	100%	100%	100%	100%	100%	100%
G04 −30 665.5387	b	−30 665.539	−30 665.539	−30 665.539	−30 665.539	−30 665.539	−30 665.5387
	Md	−30 665.539	−30 665.539	−30 665.539	−30 665.539	−30 665.539	−30 665.5387
	Mn	−30 665.539	−30 665.539	−30 665.539	−30 665.539	−30 665.539	−30 665.5387
	w	−30 665.539	−30 665.539	−30 665.539	−30 665.539	−30 665.539	−30 665.5387
	sd	2.0E−05	0.00E+00	7.4E−12	5.404E−07	1.10E−11	0.0E+00
	FR	100%	100%	100%	100%	100%	100%
G05 5126.4967	b	5126.497(1)	5126.599(6)	5126.4989(4)	5126.4981(4)	5126.497(1)	5126.4967(1)
	Md	5127.372(5)	5160.198(6)	5126.776(4)	5126.4981(3)	5126.497(1)	5126.4967(1)
	Mn	5128.881(5)	5174.492(6)	5127.648(4)	5126.4981(3)	5126.497(1)	**5126.4967(1)**
	w	5142.472(5)	5160.198(6)	5135.256(4)	5126.4984(3)	5126.497(1)	5126.4967(1)
	sd	3.5E+00	5.006E+01	1.8E+00	1.727E−07	7.20E−13	0.0E+00
	FR	100%	100%	100%	100%	100%	100%
G06 −6961.8139	b	−6961.814(1)	−6961.814(1)	−6961.814(1)	−6961.81388(1)	−6961.814(1)	−6961.8139(1)
	Md	−6961.814(1)	−6961.814(1)	−6961.814(1)	−6961.81388(1)	−6961.814(1)	−6961.8139(1)
	Mn	−6875.940(6)	−6961.284(5)	−6961.814(1)	−6961.81388(1)	−6961.814(1)	−6961.8139(1)
	w	−6350.262(6)	−6952.482(5)	−6961.814(1)	−6961.81388(1)	−6961.814(1)	−6961.8139(1)
	sd	1.6E+02	1.85E+00	4.6E−12	8.507E−12	1.90E−12	0.00E+00
	FR	100%	100%	100%	100%	100%	100%
G07 24.3062	b	24.307(5)	24.327(6)	24.306(1)	24.3064582(4)	24.306(1)	24.3062(1)
	Md	24.357(5)	24.426(6)	24.313(4)	24.3073055(3)	24.306(1)	24.3063(1)
	Mn	24.374(5)	24.475(6)	24.316(4)	24.3073989(3)	**24.306(1)**	**24.3063(1)**
	w	24.642(5)	24.843(6)	24.359(4)	24.3092401(3)	24.306(1)	24.3063(1)
	sd	6.6E−02	1.32E−01	1.1E−02	7.118E−04	6.30E−05	3.19E−05
	FR	100%	100%	100%	100%	100%	100%

TABLE VIII
(CONTINUED)

G08 −0.095825	b	−0.095825	−0.095825	−0.095825	−0.095825	−0.095825	−0.09582504
	Md	−0.095825	−0.095825	−0.095825	−0.095825	−0.095825	−0.09582504
	Mn	−0.095825	−0.095825	−0.095825	−0.095825	−0.095825	−0.09582504
	w	−0.095825	−0.095825	−0.095825	−0.095825	−0.095825	−0.09582504
	sd	2.6E−17	0.00E+00	2.8E−17	2.417E−17	2.70E−13	0.0E+00
	FR	100%	100%	100%	100%	100%	100%
G09 680.630057	b	680.630(1)	680.632(6)	680630(1)	680.6300574(1)	680.630(1)	680.630057(1)
	Md	680.641(5)	680.642(6)	680.633(4)	680.6300574(1)	680.630(1)	680.630057(1)
	Mn	680.656(6)	680.643(5)	680.639(4)	**680.6300574(1)**	**680.630(1)**	**680.630057(1)**
	w	680.763(6)	680.719(5)	680.673(4)	680.6300578(1)	680.630(1)	680.630057(1)
	sd	3.4E−02	1.55E−02	1.0E−12	9.411E−08	3.20E−13	2.61E−08
	FR	100%	100%	100%	100%	100%	100%
G10 7049.248	b	7054.316(6)	7051.903(4)	7052.253(5)	7049.286598(3)	7049.248(1)	7049.2483(1)
	Md	7372.613(6)	7253.603(5)	7215.357(4)	7049.486145(3)	7049.248(1)	7049.2488(2)
	Mn	7559.192(6)	7253.047(5)	7250.437(4)	7049.525438(3)	7049.250(2)	**7049.2490(1)**
	w	8835.655(6)	7638.366(5)	7560.224(4)	7049.984208(3)	7049.270(2)	7049.2501(1)
	sd	5.3E+02	1.36E+02	1.2E+02	1.502E−01	3.20E−03	6.60E−04
	FR	20%	100%	100%	100%	100%	100%
G11 0.749900	b	0.750	0.75	0.75	0.75	0.75	0.7499
	Md	0.750	0.75	0.75	0.75	0.75	0.7499
	Mn	0.750	0.75	0.75	0.75	0.75	0.7499
	w	0.750	0.75	0.75	0.75	0.75	0.7499
	sd	8.0E−05	1.52E−04	3.4E+02	1.546E−12	1.10E−16	0.0E+00
	FR	100%	100%	100%	100%	100%	100%
G12 −1.000	b	−1.0000	−1.0000	−1.0000	−1.0000	−1.0000	−1.0000
	Md	−1.0000	−1.0000	−1.0000	−1.0000	−1.0000	−1.0000
	Mn	−1.0000	−1.0000	−1.0000	−1.0000	−1.0000	−1.0000
	w	−1.0000	−1.0000	−1.0000	−1.0000	−1.0000	−1.0000
	sd	0.0E+00	0.00E+00	1.0E−03	0.00E+00	1.20E−09	0.0E+00
	FR	100%	100%	100%	100%	100%	100%
G13 0.0539415	b	0.053957(5)	0.053986(6)	0.053950(4)	0.0539498(3)	0.053942(1)	0.053941514(1)
	Md	0.057006(5)	0.061873(6)	0.053952(4)	0.0539498(3)	0.053942(1)	0.053941514(1)
	Mn	0.067543(5)	0.166385(6)	0.053959(3)	**0.0539498(2)**	0.066770(4)	**0.053941514(1)**
	w	0.216915(4)	0.468294(6)	0.053999(3)	0.0539499(2)	0.438803(5)	0.053941514(1)
	sd	3.1E−02	1.77E−01	1.3E−05	8.678E−08	7.09E−02	1.00E−12
	FR	100%	100%	100%	100%	100%	100%
RANK		133	152	94	66	48	29

b, Md, Mn, w, sd, and FR represent best, median, mean, worst, standard deviation, and feasibility rate, respectively. Parentheses ( ) contain ranking.

TABLE IX
COMPARISON OF SINGLE CONSTRAINT HANDLING METHODS AND ECHT USING DE AND EP

Prob		SF-DE	SP-DE	SR-DE	EC-DE	ECHT-DE	SF-EP	SP-EP	SR-EP	EC-EP	ECHT-EP
H01	b	5.59E−11	1.24E−85	7.57E−81	5.59E−11	8.29E−83	2.57E+02	4.29E−09	2.96E−05	2.57E+02	5.58E−13
	Md	3.33E+03	3.42E−84	3.26E−78	3.33E+03	2.19E−80	1.78E+03	1.48E−07	8.47E−04	1.78E+03	2.18E−11
	Mn	4.58E+03	**2.95E−83**	**1.37E−74**	4.58E+03	**2.66E−78**	1.94E+03	0.8699	1.81E−03	1.94E+03	**3.02E−11**
	w	1.67E+04	3.47E−82	4.04E−37	1.67E+04	7.41E−77	3.70E+03	8.4726	9.43E03	3.70E+03	1.04E−10
	sd	5.63E+03	6.92E−83	7.37E−74	5.63E+03	1.35E−77	1.13E+03	2.6718	2.80E−03	1.13E+03	3.19E−11
	FR	27%	100%	100%	27%	100%	100%	100%	100%	100%	100%
	h	1	0	0	1	–	1	1	1	1	–
H02	b	−1.0123	−2.2720	0.0752	−1.0123	−2.2776	−1.1228	−2.2769	−2.2627	−1.1228	−2.2776
	Md	0.5913	−1.9833	2.1127	0.5913	−2.2604	−0.3177	−2.2762	−2.2168	−0.3177	−2.2773
	Mn	0.7730	−1.4213	2.1672	0.7730	**−2.2516**	−0.2008	**−2.2730**	−2.2065	−0.2008	**−2.2764**
	w	2.9000	1.3637	5.1200	2.9000	−2.2168	1.3043	−2.2604	−2.1588	1.3043	−2.2709
	sd	1.3000	1.0542	1.4263	1.3000	0.0313	0.7182	0.0065	0.0363	0.7182	0.0021
	FR	94%	97%	100%	94%	100%	100%	100%	100%	100%	100%
	h	1	1	1	1	–	1	0	1	1	–

TABLE IX
(CONTINUED)

	b	1.08E−81	2.95E−85	6.25E−80	1.08E−81	1.19E−83	39.7678	6.50E−09	0.0062	39.7678	1.00E−15
	Md	4.59E−80	2.15E−83	3.40E−78	4.59E−80	1.14E−81	93.8244	2.95E−08	0.2205	93.8244	3.30E−11
	Mn	**2.05E−78**	**8.31E−83**	**2.20E−77**	**2.05E−78**	**6.90E−81**	267.1285	**4.01E−08**	0.3108	267.1285	**3.26E−10**
H03	w	3.39E−77	7.15E−82	2.20E−76	3.39E−77	3.68E−80	1038.1376	1.03E−07	0.7313	1038.1376	2.38E−09
	sd	6.53E−78	1.58E−82	5.02E−77	6.53E−78	1.12E−80	373.6603	3.22E−08	0.2749	373.6603	7.43E−10
	FR	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	h	0	0	0	0	–	1	0	1	1	–
	b	3.89E−93	4.17E−94	2.50E−94	3.89E−93	4.91E−95	2.55E+03	0.0487	1.52E−04	2.55E+03	1.54E−13
	Md	4.01E−91	6.93E−93	1.99E−92	4.01E−91	3.24E−93	2.77E+03	14.251	9.25E−03	2.77E+03	4.39E−13
	Mn	1.12E+03	**5.81E−92**	**3.41E−91**	1.12E+03	**1.01E−92**	2.77E+03	28.5655	9.77E−03	2.77E+03	**1.89E−11**
H04	w	1.24E+04	7.01E−91	4.69E−90	1.24E+04	7.98E−92	2.99E+03	112.6388	2.27E−02	2.99E+03	1.72E−10
	Sd	3.21E+03	1.41E−91	9.57E−91	3.21E+03	1.85E−92	3.11E+02	38.5973	8.29E−03	3.11E+02	5.40E−11
	FR	84%	100%	100%	84%	100%	30%	100%	100%	30%	100%
	h	1	0	0	1	–	1	1	1	1	–
	B	−20.0780	−20.0780	−18.9875	−20.0780	−20.0780	−18.9875	−19.9299	−19.9286	−18.9875	−20.0780
	Md	−19.5327	−20.0780	−16.4701	−18.9875	−20.0780	−18.9875	−18.9875	−17.897	−18.2295	−18.9875
	Mn	−19.5371	−19.9198	−16.6614	−19.3877	**−20.0774**	−18.6736	−18.6546	−18.4454	−18.3997	**−19.3877**
H05	W	−18.9387	−18.7566	−14.9096	−18.0109	−20.0599	−17.8970	−17.8970	−17.8970	−17.8970	−18.0109
	sd	0.5563	0.3977	0.9701	0.5997	0.0033	0.4818	0.7706	0.8524	0.5218	0.5997
	FR	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	h	1	1	1	1	–	1	1	1	1	–
	b	−8.3826	−8.3826	−8.3826	−8.3826	−8.3826	−2.5483	−8.8326	−8.3810	−8.8326	−8.8326
	Md	−2.5483	−8.3826	−8.3824	−2.5483	−8.3826	−2.5483	−8.8326	−8.3781	−8.8326	−8.8326
	Mn	−4.1041	−8.1881	−8.1663	−2.9761	**−8.3826**	−2.5483	**−8.8326**	**−8.3779**	**−8.8326**	**−8.8326**
H06	w	−2.5483	−2.5483	−2.5483	−2.5483	−8.3826	−2.5482	−8.8326	−8.3739	−8.8326	−8.8326
	sd	2.6241	1.065	1.1228	1.3524	3.76E−15	1.95E−05	1.14E−05	2.45E−03	1.03E−05	1.77E−05
	FR	100%	100%	90%	94%	100%	100%	100%	100%	100%	100%
	h	1	1	1	1	–	1	0	0	0	–
	b	−7.6159	−7.6155	−7.6159	−6.0200	−7.6159	−7.5326	−7.6159	−7.6198	−7.6159	−7.6159
	Md	−4.4228	−7.5301	−7.6156	−1.4220	−7.6159	−1.3735	−7.5344	−7.6103	−6.8763	−7.6159
	Mn	−4.5019	−6.3253	−6.8220	−1.8649	**−7.6159**	−1.9429	−5.5958	−7.6041	−4.7365	**−7.6159**
H07	w	−1.4220	−1.4206	−1.4211	−1.2526	−7.6159	−1.0732	−0.1399	−7.5667	−0.0492	−7.6159
	sd	3.2469	2.5860	1.9615	1.4609	4.26E−10	1.9675	3.1946	0.018	3.3767	3.18E−09
	FR	100%	100%	100%	100%	100%	100%	100%	0%	100%	100%
	h	1	1	1	1	–	1	1	1	1	–
	b	−161.5611	−481.8698	−471.9798	500.0000	−483.6106	−179.5575	−369.6769	−368.3882	−84.3837	−483.6106
	Md	191.5272	−480.7047	−468.9959	500.0000	−483.6106	79.3192	−355.7125	−358.7575	44.06314	−483.6106
	Mn	180.3723	−480.3973	−467.2375	500.000	**−483.6106**	85.8499	−354.6428	−361.2611	44.06314	**−483.6106**
H08	w	500.0000	−477.9481	−454.5988	500.0000	−483.6106	456.9144	−335.6941	−354.8487	172.5100	−483.6106
	sd	369.5312	1.1284	6.4096	0.00E+00	0.00E+00	216.9676	12.6157	5.0842	181.6512	0.00E+00
	FR	30%	100%	60%	30%	100%	0%	100%	90%	20%	100%
	h	1	1	1	1	–	1	1	1	1	–
	b	−68.4294	−51.5310	−59.4305	−68.4294	−68.4294	−68.4293	−62.2762	−68.4287	−68.4293	−68.4294
	Md	−68.4294	−44.8977	−49.9649	−68.4294	−68.4294	−63.5175	−57.1397	−63.5137	−64.5479	−63.5175
	Mn	−67.4630	−44.6524	−50.3589	−67.6107	**−67.9231**	−64.3957	−57.3526	−64.4546	−65.2790	**−64.9120**
H09	w	−63.5175	−38.6257	−44.2585	−53.5175	−63.5175	−61.6487	−56.0558	−62.2736	−62.2764	−63.5174
	sd	1.7121	2.8442	3.2128	1.8618	1.0938	2.3528	1.8096	2.2796	2.3893	2.0358
	FR	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	h	1	1	1	1	–	1	1	1	1	–
	b	0.0000	0.0000	0.0000	0.0000	0.0000	0.1465	0.0151	0.0987	0.1465	0.0000
	Md	0.0000	0.0000	8.9900	0.0000	0.0000	8.9900	8.9900	5.0077	8.9900	0.0044
	Mn	4.1953	3.8957	5.3940	4.1953	**0.5993**	5.5035	6.3285	4.6927	5.5035	**3.5970**
H10	w	8.9900	8.9900	8.9900	8.9900	8.9900	8.9902	8.9901	8.9904	8.9902	8.9900
	sd	4.5617	4.5310	4.4795	4.5617	2.2808	4.5105	4.2890	4.5373	4.5015	4.6416
	FR	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	h	1	1	1	1	–	1	1	1	1	–
	b	580.7301	580.7301	893.9601	580.7301	580.7301	580.7304	580.7302	580.8066	580.7304	580.7301
	Md	580.7301	580.7301	968.3112	580.7301	580.7301	580.7336	580.7302	581.0729	580.7336	580.7301
	Mn	600.6946	**580.7301**	974.2255	600.6946	**580.7301**	582.1345	**580.7303**	581.223	582.1345	**580.7303**
H11	w	655.2965	580.7301	1108.8864	655.2965	580.7301	594.7405	580.7312	581.5905	594.7405	580.7310
	sd	27.0368	2.11E−14	49.0910	27.0368	1.32E−11	4.4293	0.0003	0.3058	4.4293	0.0003
	FR	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	h	1	0	1	1	–	1	0	1	1	–

TABLE IX

(CONTINUED)

H12	b	1.14E−29	1.25E−30	31.8321	3.20E−29	1.54E−32	0.0239	8.84E−06	0.0027	0.0158	5.00E−07
	Md	50.0000	1.55E−28	50.0000	50.0000	2.41E−31	0.6904	5.36E−05	0.0214	0.8902	1.06E−06
	Mn	39.8874	0.0797	48.0484	42.7604	**4.55E−31**	0.7402	7.67E−05	0.0465	0.2269	**1.95E−06**
	w	50.0000	1.2888	50.0000	50.0000	1.75E−30	0.5497	1.79E−04	0.1409	4.7892	1.06E−05
	sd	20.2878	0.3041	4.8136	17.2038	4.61E−31	1.531	5.53E−05	0.0496	1.3226	3.06E−06
	FR	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	h	1	1	1	1	–	1	0	1	1	–
H13	b	−46.3755	−46.3755	−46.3755	−46.3755	−46.3755	−46.3754	−46.3755	−46.3701	−46.3753	−46.3755
	Md	−46.3755	−46.3755	−46.3755	−46.3755	−46.3755	−43.5685	−46.3753	−43.5667	−43.5685	−46.3755
	Mn	−46.2820	−46.2820	−46.2817	−46.0949	**−46.3755**	−43.6266	−45.8140	−44.4050	−43.6266	**−46.3755**
	w	−43.5686	−43.5686	−43.5686	−43.5686	−46.3755	−43.3436	−43.5686	−43.5632	−41.3436	−46.3755
	sd	0.5125	0.5125	0.5124	0.8565	9.47E−15	1.1924	1.1834	1.3512	1.1923	7.15E−10
	FR	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
	h	1	1	1	1	–	1	1	1	1	–

b, Md, Mn, w, sd, and FR represent best, median, mean, worst, standard deviation, and feasibility rate, respectively.

corresponding single cases. ECHT-EP performs better than ECHT-DE on problem H02 which has a unimodal, linear objective function and multimodal, nonlinear constraints. ECHT-DE performs better than ECHT-EP on problems H05, H09 and H10 which have multimodal, nonlinear objective and constraint functions. In problems H01, H02, and H06, where both equality and inequality constraints are present, the feasibility rate attained by single constraint handling methods using EP as search method is better. In problems H04 and H08, where only equality constraints are present, the feasibility rate attained by single constraint handling methods using DE as search method is better. Thus, the performance varies not only with respect to the constraint handling method but also with the chosen search algorithm.

VII. CONCLUSION

This paper has presented a novel constraint handling procedure called ECHT with four different constraint handling methods where each constraint handling method has its own population. In ECHT every function call is effectively used by all four populations and the offspring population produced by the best suited constraint handling technique dominates the others at a particular stage of the optimization process. Furthermore, an offspring produced by a particular constraint handling method may be rejected by its own population, but could be accepted by the populations associated with other constraint handling methods. NFL theorem implies that irrespective of the exhaustiveness of parameter tuning, no single constraint handling method can be the best for every constrained optimization problem. Hence, according to the NFL, the ECHT has the potential to perform well over diverse problems over any single constraint handling method. We tested the performances of four individual constraint handling methods and ECHT using DE and adaptive CEP algorithm. Experimental results showed that the ECHT outperforms all four single constraint handling methods, as well as the state-of-the-art methods.

APPENDIX A

Definitions of 13 newly designed problems.

H01

$$\text{Min } f(x) = \sum_{i=1}^{D} x_i^2$$

$$g_1(x) = \frac{1}{D}\sum_{i=1}^{D}(-x_i * \cos(\sqrt{|x_i|})) \leq 0$$

$$h_1(x) = \sum_{i=1}^{D/2}(-x_{2i-1} * \cos(\sqrt{|x_{2i-1}|})) = 0$$

$$h_2(x) = \sum_{i=1}^{D/2}(-x_{2i} * \cos(\sqrt{|x_{2i}|})) = 0$$

$$x \in [-50,\ 50]^D.$$

H02

$$\text{Min} f(x) = \max(x)$$

$$\mathrm{g}_1(x) = 10 - \frac{1}{D}\sum_{i=1}^{D}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right] \leq 0$$

$$\mathrm{g}_2(x) = \frac{1}{D}\sum_{i=1}^{D}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right] - 15 \leq 0$$

$$h(x) = \frac{1}{D}\sum_{i=1}^{D}\left[y_i^2 - 10\cos(2\pi y_i) + 10\right] - 20 = 0 \quad y = x - 0.5$$

$$x \in [-5.12,\ 5.12]^{\mathrm{D}}.$$

H03

$$\text{Min } f(x) = \sum_{i=1}^{D} x_i^2$$

$$\mathrm{g}(x) = \sum_{i=1}^{D/2}(x_{2i} * \sin(\sqrt{|x_{2i}|})) \leq 0$$

$$\mathrm{h}(x) = \sum_{i=1}^{D/2}(-x_{2i-1} * \sin(\sqrt{|x_{2i-1}|})) = 0$$

$$x \in [-100,\ 100]^{\mathrm{D}}.$$

H04

$$\text{Min } f(x) = \sum_{i=1}^{D} x_i^2$$

$$\text{h}_1(x) = \sum_{i=1}^{D/2}(-x_{2i-1} * \sin(\sqrt{|x_{2i-1}|})) = 0$$

$$\text{h}_2(x) = \sum_{i=1}^{D/2}(x_{2i} * \sin(\sqrt{|x_{2i}|})) = 0$$

$$x \in [-100,\ 100]^{\text{D}}.$$

H05

$$\text{Min } f(x) = \frac{1}{D}\sum_{i=1}^{D}(-x_i * \sin(\sqrt{|x_i|}))\quad y = x - 100,\ z{=}x + 50$$

$$\text{g}_1(x) = 10 + \frac{1}{D}\sum_{i=1}^{D}(-y_i * \sin(\sqrt{|y_i|})) \le 0$$

$$\text{g}_2(x) = 10 + \frac{1}{D}\sum_{i=1}^{D}(-z_i * \sin(\sqrt{|z_i|})) \le 0$$

$$x \in [-100,\ 100].$$

H06

$$\text{Min } f(x) = \frac{1}{D}\sum_{i=1}^{D}(-x_i * \sin(\sqrt{|x_i|}))$$

$$\text{g}(x) = -100 + \sum_{i}^{D-1}((x_i^2 - x_{i+1})^2 + (x_i - 1)^2) \le 0$$

$$x \in [-100,\ 100].$$

H07

$$\text{Min } f(x) = \frac{1}{D}\sum_{i=1}^{D}(-x_i * \sin(\sqrt{|x_i|}))$$

$$\text{g}_1(x) = -100 + \sum_{i}^{D-1}((x_i^2 - x_{i+1})^2 + (x_i - 1)^2) \le 0$$

$$\text{g}_2(x) = -20 + \prod_{i=1}^{D}\frac{x_i^2}{4000} - \prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}) + 1 \le 0$$

$$\text{h}(x) = \sum_{i=1}^{D-1}(x_i^2 - x_{i+1})^2 = 0$$

$$x \in [-100,\ 100].$$

H08

$$\text{Min } f(x) = \max(x)$$

$$\text{h}(x) = \frac{1}{D}\sum_{i=1}^{D}(-x_i * \sin(\sqrt{|x_i|})) = 0$$

$$x \in [-500,\ 500].$$

H09

$$\text{Min } f(x) = \frac{1}{D}\sum_{i=1}^{D}(-x_i * \sin(\sqrt{|x_i|}))$$

$$\text{g}_1(x) = -50 + \frac{1}{100 * D}\sum_{i}^{D} x_i^2 \le 0$$

$$\text{g}_2(x) = 50 * \sum_{i=1}^{D}\sin(\frac{1}{50} * \pi * x) \le 0$$

$$\text{g}_3(x) = 75 - 50 * (\sum_{i=1}^{D}\frac{x_i^2}{4000} - \prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}) + 1) \le 0$$

$$x \in [-500,\ 500].$$

H10

$$\text{Min } f(x) = \sum_{i=1}^{D-1}(100 * (x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$$

$$\text{g}(x) = \sum_{i=1}^{D}(x * \cos(2 * \sqrt{|x_i|})) \le 0$$

$$x \in [-10,\ 10].$$

H11

$$\text{Min } f(x) = \sum_{i=1}^{D}(z_i^2 - 10 * \cos(2 * \pi * z) + 10)\quad z = \text{x} - 10$$

$$\text{h}_1(x) = \frac{1}{D}\sum_{\text{i}=1}^{\text{D}}(-x * \sin(2 * \sqrt{|x_i|})) \le 0$$

$$\text{h}_2(x) = \frac{1}{D}\sum_{\text{i}=1}^{\text{D}}(x * \sin(2 * \sqrt{|x_i|})) \le 0$$

$$x \in [-5,\ 5].$$

H12

$$\text{Min } f(x) = \max(|x|)$$

$$\text{h}(x) = \frac{1}{D}\sum_{i=1}^{D}(-x_i * \sin(\sqrt{|x_i|})) = 0$$

$$x \in [-50,\ 50].$$

H13

$$\text{Min } f(x) = \max(x)$$

$$\text{g}_1(x) = \frac{1}{D}\sum_{i=1}^{D}(-x_i * \sin(\sqrt{|x_i|})) \le 0$$

$$\text{g}_2(x) = \frac{1}{D}\sum_{i=1}^{D}(x_i * \cos(\sqrt{|x_i|})) \le 0$$

$$x \in [-100,\ 100].$$

REFERENCES

[1] C. A. C. Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state-of-the-art," *Comput. Methods Appl. Mech. Eng.*, vol. 191, nos. 11–12, pp. 1245–1287, Jan. 2002.

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

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

[4] Y. Wang, Z. Cai, G. Guo, and Y. Zhou, "Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems," *IEEE Trans. Syst., Man, Cybern.*, vol. 37, no. 3, pp. 560–575, Jun. 2007.

[5] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[6] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Comput.*, vol. 9, no. 1, pp. 3–12, Jan. 2005.

[7] V. L. Huang, A. K. Qin, and P. N. Suganthan, "Self-adaptive differential evolution algorithm for constrained real-parameter optimization," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 215–222.

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

[9] B. Tessema and G. G. Yen, "A self-adaptive penalty function based algorithm for constrained optimization," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 246–253.

[10] J. Brest, V. Zumer, and M. S. Maucec, "Self-adaptive differential evolution algorithm in constrained real-parameter optimization," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 215–222.

[11] M. F. Tasgetiren and P. N. Suganthan, "A multipopulated differential evolution algorithm for solving constrained optimization problems," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 33–40.

[12] S. Kukkonen and J. Lampinen, "Constrained real-parameter optimization with generalized differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 207–214.

[13] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. C. Coello, "Modified differential evolution for constrained optimization," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 25–32.

[14] T. Takahama and S. Sakai, "Constrained optimization by the constrained differential evolution with gradient-based mutation and feasible elites," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 1–8.

[15] K. Zielinski and R. Laur, "Constrained single-objective optimization using differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 223–230.

[16] J. J. Liang and P. N. Suganthan, "Dynamic multiswarm particle swarm optimizer with a novel constraint-handling mechanism," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 9–16.

[17] K. Zielinski and R. Laur, "Constrained single-objective optimization using particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 443–450.

[18] E. Mezura-Montes and C. A. C. Coello, "Adding diversity mechanism to a simple evolution strategy to solve constrained optimization problems," in *Proc. Congr. Evol. Comput.*, 2003, pp. 6–13.

[19] E. Mezura-Montes and C. A. C. Coello, "A simple multimembered evolution strategy to solve constrained optimization problems," *IEEE Trans. Evol. Comput.*, vol. 9, no. 1, pp. 1–17, Feb. 2005.

[20] T. P. Runarsson, "Approximate evolution strategy using stochastic ranking," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 745–752.

[21] T. P. Runarsson and X. Yao, "Search biases in constrained evolutionary optimization," *IEEE Trans. Syst., Man, Cybern.*, vol. 35, no. 2, pp. 233–243, May 2005.

[22] Y. Wang, Z. Cai, Y. Zhou, and W. Zeng, "An adaptive tradeoff model for constrained evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 80–92, Feb. 2008.

[23] J. H. Kim and H. Myung, "Evolutionary programming techniques for constrained optimization problems," *IEEE Trans. Evol. Comput.*, vol. 1, no. 2, pp. 129–140, Jul. 1997.

[24] R. Mallipeddi and P. N. Suganthan, "Evaluation of novel adaptive evolutionary programming on four constraint handling techniques," in *Proc. IEEE Congr. Evol. Comput.*, Hong Kong, China, 2008, pp. 4045–4052.

[25] A. Sinha, A. Srinivasan, and K. Deb, "A population-based, parent centric procedure for constrained real-parameter optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2006, pp. 239–245.

[26] S. Venkatraman and G. G. Yen, "A generic framework for constrained optimization using genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 9, no. 4, pp. 424–435, Aug. 2005.

[27] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.

[28] S. B. Hamida and M. Schoenauer, "ASCHEA: New results using adaptive segregational constraint handling," in *Proc. Congr. Evol. Comput.*, vol. 1. Honolulu, HI, May 2002, pp. 82–87.

[29] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comp. Methods Appl. Mech. Eng.*, vol. 186, nos. 2–4, pp. 311–338, Jun. 2000.

[30] D. Powell and M. Skolnick, "Using genetic algorithms in engineering design optimization with nonlinear constraints," in *Proc. 5th Int. Conf. Genetic Algorithms*, San Mateo, CA: Morgan Kaufmann, 1993, pp. 424–431.

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

[32] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 204–223, Apr. 2003.

[33] Y.-S. Ong, M.-H. Lim, N. Zhu, and K.-W. Wong, "Classification of adaptive memetic algorithms: A comparative study," *IEEE Trans. Syst., Man, Cybern.*, vol. 36, no. 1, pp. 141–152, Feb. 2006.

[34] Y. S. Ong and A. J. Keane, "Meta-Lamarckian learning in memetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 99–110, Apr. 2004.

[35] Z. Skolicki and K. De Jong, "The importance of a two-level perspective for island model design," in *Proc. IEEE Congr. Evol. Comput.*, Singapore, Sep. 2007, pp. 4623–4630.

[36] D. B. Fogel, *System Identification Through Simulated Evolution: A Machine Learning Approach to Modeling*. Needham Heights, MA: Ginn Press, 1991.

[37] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence Through Simulated Evolution*. New York: Wiley, 1966.

[38] N. Saravanan, D. B. Fogel, and K. M. Nelson, "A comparison of methods for self-adaptation in evolutionary algorithms," *BioSystems*, vol. 36, no. 2, pp. 157–166, 1995.

[39] K. H. Liang, X. Yao, and C. S. Newton, "Adapting self-adaptive parameters in evolutionary algorithms," *Appl. Intell.*, vol. 15, no. 3, pp. 171–180, 2001.

[40] C. Y. Lee and X. Yao, "Evolutionary programming using mutations based on the Levy probability distribution," *IEEE Trans. Evol. Comput.*, vol. 8, no. 1, pp. 1–13, Feb. 2004.

[41] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

[42] X. Yao and Y. Liu, "Scaling up evolutionary programming algorithms," in *Proc. 7th Annu. Conf. Evol. Program. (EP98)*, Lecture Notes in Computer Science, Berlin, Germany: Springer-Verlag, 1998, pp. 103–112.

[43] R. Storn and K. Price, "Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, Dec. 1997.

[44] R. Storn and K. Price, "Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces," Int. Comput. Sci. Inst., Berkeley, CA, Tech. Rep. TR-95-012, Mar. 1995.

[45] R. Joshi and A. C. Sanderson, "Minimal representation multisensor fusion using differential evolution," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 29, no. 1, pp. 63–76, Jan. 1999.

[46] R. Storn, "Differential evolution design of an IIR-filter," in *Proc. IEEE Int. Conf. Evol. Comput.*, Nagoya, Japan, May 1996, pp. 268–273.

[47] S. Das, A. Abraham, and U. K. Chakraborty, "Differential evolution using a neighborhood-based mutation operator," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526–553, Jun. 2009.

[48] 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 CEC 2006 special session on

constrained real-parameter optimization," Nanyang Technol. Univ., Singapore, Tech. Rep. #2 006 005, Dec. 2005 [Online]. Available: http://www3.ntu.edu.sg/home/EPNSugan/

Rammohan Mallipeddi received the B.Tech. degree in electrical and electronic engineering from Acharya Nagarjuna University, Guntur, Andhra Pradesh, India, in 2005, and the Master's degree in computer control and automation from the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore, in 2007. Since 2008, he has been working toward the Ph.D. degree at the School of Electrical and Electronics Engineering, Nanyang Technological University.

He was a Research Associate at the Department of Electrical and Electronic Engineering, Nanyang Technological University, Singapore from 2007 to 2009. His research interests include evolutionary computing, artificial intelligence, image processing, digital signal processing, robotics, and control engineering.

Ponnuthurai N. Suganthan (SM'00) received the B.A. degree, the Postgraduate Certificate, and the M.A. degree in electrical and information engineering from the University of Cambridge, Cambridge, U.K., in 1990, 1992, and 1994, respectively. He received the Ph.D. degree in electrical and electronic engineering from the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore, in 1996.

He was a Predoctoral Research Assistant in the Department of Electrical Engineering, University of Sydney, Sydney, New South Wales, Australia, from 1995 to 1996, and a Lecturer in the Department of Computer Science and Electrical Engineering, University of Queensland, Brisbane St. Lucia, Queensland, Australia, from 1996 to 1999. From 1999 to 2003, he was an Assistant Professor in the School of Electrical and Electronics Engineering, Nanyang Technological University, where he is currently an Associate Professor. His research interests include evolutionary computation, applications of evolutionary computation, neural networks, pattern recognition, and bioinformatics.

He is an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and *J. Pattern Recognition*.