

A novel differential evolution algorithm for solving constrained engineering optimization problems

Ali Wagdy Mohamed¹

Received: 20 July 2016 / Accepted: 8 January 2017 / Published online: 24 January 2017
© Springer Science+Business Media New York 2017

Abstract This paper introduces a novel differential evolution (DE) algorithm for solving constrained engineering optimization problems called (NDE). The key idea of the proposed NDE is the use of new triangular mutation rule. It is based on the convex combination vector of the triplet defined by the three randomly chosen vectors and the difference vectors between the best, better and the worst individuals among the three randomly selected vectors. The main purpose of the new approach to triangular mutation operator is the search for better balance between the global exploration ability and the local exploitation tendency as well as enhancing the convergence rate of the algorithm through the optimization process. In order to evaluate and analyze the performance of NDE, numerical experiments on three sets of test problems with different features, including a comparison with thirty state-of-the-art evolutionary algorithms, are executed where 24 well-known benchmark test functions presented in CEC'2006, five widely used constrained engineering design problems and five constrained mechanical design problems from the literature are utilized. The results show that the proposed algorithm is competitive with, and in some cases superior to, the compared ones in terms of the quality, efficiency and robustness of the obtained final solutions.

Keywords Engineering optimization · Differential evolution · Triangular mutation · Constrained optimization

Introduction

Generally, optimization is the process of finding the best results for a given problem under certain conditions. In real world problems, applications and different fields of science and engineering, most of the optimization problems are subject to different types of constraints. Thus, these problems are called constrained optimization problems (COPs) and are considered challenging and complex due to their variations in mathematical properties and structures (Mohamed and Sabry 2012; de Melo and Carosio 2012; Dong and Wang 2014; Elsayed et al. 2014; Akay and Karaboga 2012; Brajevic and Tuba 2013; Yi et al. 2016a, b; Yu et al. 2016; Mohamed 2015b). COPs can be classified in several ways depending on their mathematical properties. They can be classified based on the nature of variables, such as real, integer and discrete, and may have equality and/or inequality constraints. Another important alternative classification depends on the type of expression of objective and constraints functions. It can be classified as linear or non-linear, continuous or discontinuous and unimodal or multimodal. Furthermore, the feasible region of such problems can be either a tiny or significant portion of the search space. It can be either one single-bounded region or a collection of multiple disjoint regions.

Due to the above mentioned difficulties characteristics, over the past decade, many attempts have been made to solve COPs using various types of computational intelligence algorithms together with different constraint-handling techniques. A complete study on the most constraint-handling techniques used with evolutionary algorithms (EAs) has been discussed in details (Coello 2002). In this research, most recent studies and well-known algorithms done over the past decade are discussed. As a matter of fact, Genetic Algorithms (GAs) are considered the pioneer algorithm of

✉ Ali Wagdy Mohamed
aliwagdy@gmail.com

¹ Operations Research Department, Institute of Statistical Studies and Research, Cairo University, Giza 12613, Egypt

EAs and have been successfully applied to solve constrained optimization problems. [Tessema and Yen \(2006\)](#) introduced a GA based on an adaptive penalty function. The introduced method aimed to exploit infeasible individuals with low objective function values and low constraint violations. The approach was tested on a set of 13 benchmark problems with the results showing that the approach was able to find feasible solutions in all runs. However, its performance was not robust. A multi-objective optimization based on a hybrid evolutionary algorithm to solve constrained optimization problems, named (HCOEA), was proposed by [Wang et al. \(2007\)](#). This method consists of two main parts: the global search model and the local search model. In the global search model, a niching genetic algorithm is proposed. The local search model performs a parallel local search operation based on clustering partition of the population. In this method, the comparison of individuals is based on multi-objective optimization technique. Recently, [Elsayed et al. \(2011\)](#) proposed a self-adaptive multi-operator genetic algorithm (SAMO-GA). In SAMO-GA, the population is divided into four sub-populations with individual crossover and mutation where the feasibility rules proposed by [Deb \(2000\)](#) were used to handle the constraints. SAMO-GA has shown significant improvement in comparison to other variants of GA with a single operator. Many optimization problems have been solved by using other branches of EAs which are evolutionary strategies (ES) and evolutionary programming (EP). [Mezura-Montes and Coello \(2005\)](#) proposed a simple multi membered evolution strategy to solve constrained optimization problems (SMES). Instead of using the penalty function approach, the proposed approach uses a simple diversity mechanism which allows infeasible solutions to remain in the population. The approach tested on the benchmark problems and the results obtained were very competitive when compared with other techniques in terms of the number of function evaluations used. On the other hand and as motivated by the ensemble operators technique in the literature, an ensemble of four constraint handling techniques combined with evolutionary programming algorithm, named (ECHT-EP2), is proposed by [Mallipeddi and Suganthan \(2010\)](#). The performance of ECHT-EP2 is competitive to the state-of-the-art algorithms. Nonetheless, since most real life problems can be mathematically formulated as optimization problems, developing another various types of efficient optimization techniques to deal with unconstrained and constrained problems has attracted much research in recent years, such as swarm intelligence which is nature-inspired algorithms that include the well-known algorithm, particle swarm optimization (PSO). Besides, latest swarm algorithms, the Firefly Algorithm (FA) was first proposed ([Yang 2009](#)), the artificial bee colony (ABC) was originally proposed ([Karaboga 2005](#)) and Seeker Optimization Algorithm (SOA) was recently proposed ([Dai et al. 2006](#)). [Elsayed et al. \(2014\)](#) proposed a

self-adaptive mix of particle swarm methodologies (SAM-PSO). In the proposed ensemble algorithm, a set of different PSO variants are used and each of which evolves with different number of individuals from the current population. Besides, a new PSO variant is developed to balance both local and global PSO versions. SAM-PSO outperforms other PSO variants and other state-of-the-art algorithms. A modified Artificial Bee Colony algorithm (M-ABC) is proposed ([Mezura-Montes and Cetina-Domínguez](#)). Four modifications are made to improve its performance in a constrained search space. In order to improve the performance of both FA and SOA in solving constrained optimization problems, [Tuba and Bacanin \(2014\)](#) proposed a modified algorithm that hybridizes FA and SOA, named SOA-FS. It outperforms other state-of-the-art swarm intelligence algorithms. Similar to other Evolutionary algorithms (EAs), differential evolution (DE) is a stochastic population based search method, proposed by [Storn and Price \(1995\)](#). The DE algorithm coupling with the constraint-handling technique has been successfully used for solving the COPs. [Becerra and Coello \(2006\)](#) proposed a cultural algorithm with a differential evolution population for the COPs. In this method, this culture algorithm has used different knowledge sources to influence the variation operator of the differential evolution algorithm. [Zhang et al. \(2008\)](#) introduced multi-member DE with dynamic stochastic ranking to keep the feasible solutions as well as the promising infeasible solution during the evolution process. After extensive experiments studying the performance of several DE variants for solving the COPs, [Mezura-Montes et al. \(2010\)](#) combined two variants: DE/rand/1/bin and DE/best/1/bin into one single approach, called (DECV). Recently, a novel constrained optimization based on a modified DE algorithm (COMDE) is proposed by [Mohamed and Sabry \(2012\)](#), where a new directed mutation strategy and a new dynamic tolerance technique have been developed to handle equality constraints. Taking advantage of multi-objective optimization techniques, [Wang and Cai \(2012\)](#) presented a multi-objective optimization based on differential evolution algorithm, named (CMODE), with novel infeasible solution replacement mechanism to guide the population toward feasible region. [Gong et al. \(2014\)](#), presented the rank-iMDDE method, in which DE is combined with a ranking-based mutation operator to enhance the convergence speed and an improved dynamic diversity mechanism to maintain either infeasible or feasible solutions in the population. [Elsayed et al. \(2011\)](#) proposed self-adaptive DE algorithm with multi-operator strategy (SAMO-DE). In this algorithm, two DE variants rand-to-best and current/2/best are used for mutation and Gaussian numbers are used to generate F and CR values. Motivated by the recent success of diverse memetic approaches to solve many optimization problems, [Dong and Wang \(2014\)](#) proposed a new memetic DE algorithm with dynamic preference (MDEDP).

In this memetic algorithm, DE is the global search and it is guided by a novel achievement scalarizing function. Simplex crossover (SPX) is used as a local search procedure to guide the search approaching the boundary of the feasible region. After performing many experimental analyses using eight different variants of the DE algorithm to solve COPs, a self-adaptive multi-strategy DE (SAMSDE) is exhibited by [Elsayed et al. \(2012\)](#). The proposed algorithm divides the population into a number of subpopulations, where each subpopulation evolves with its own mutation and crossover. The population size, crossover rate and scaling factor are adaptive to the search progresses. To speed up the convergence of SAMSDE, the SQP based local search is used to a randomly selected individual from the entire population. [Elsayed et al. \(2013\)](#) introduced an improved self-adaptive multi-operator DE algorithm (ISAMODE-CMA) that adopts a mix of different DE mutation operator. To improve the local search ability of DE, CMA-ES proposed by [Hansen et al. \(2003\)](#) is periodically applied. To tackle constraints of the problem, the dynamic penalty constraints-handling technique is used. Along the same lines, [Asafuddoula et al. \(2014\)](#), proposed an adaptive hybrid DE algorithm (AH-DEa). In this algorithm, the basic mutation strategy DE/rand/1/bin is combined with binomial and exponential crossover. Additionally, the crossover rate is adaptively controlled based on the success of the trial solutions generated. The Sequential Quadratic Programming (SQP) based local search is invoked from the best individual found to explore possibilities of further improvement. Following the same concept of parameter adaptation with different mechanisms, [Sarker et al. \(2014\)](#) presented a DE algorithm with dynamic parameters selection (DE-DPS), where three sets of DE parameters, amplification factor (F), crossover rate (CR) and population size (NP) are considered. Each individual is assigned a random combination of F and CR. For each combination, the success rate is recorded for a certain number of generations and the one with better performance is applied for a number of following generations. This process is repeated for many cycles. From the above discussion and analysis of the recent studies to solve COPs, it can be seen that all previous and current attempts have focused on applying ensemble operator techniques or hybridizing one or more algorithms and/or methods coupling with developing self-adaptive mechanism which lead to complicated algorithms with extra parameters. However, with the exception of [Mohamed and Sabry \(2012\)](#), few attempts have been made to develop a new mutation scheme to further improve the performance of DE. Consequently, motivated by this analysis and discussion, a novel triangular mutation rule without any extra parameters is proposed; the aim being to balance both the exploration capability and the exploitation tendency and enhance the convergence speed of the algorithm. The proposed algorithm is tested and analyzed by solving a set of well-known benchmark functions and

engineering and mechanical design problems. The proposed algorithm shows a superior and competitive performance to other recent constrained optimization algorithms. It is worth noting that although this work is an extension and modification of our previous work in [Mohamed \(2015a\)](#), there are significant differences and these are as follows: (1) Previous work in [Mohamed \(2015a\)](#) is proposed for unconstrained problems, whereas this work is proposed for constrained problems. (2) The crossover rate in [Mohamed \(2015a\)](#) is given by a dynamic non-linear increased probability scheme, but in this work, the crossover rate is fixed 0.95. (3) In [Mohamed \(2015a\)](#), only one difference vector between the best and the worst individuals among the three randomly selected vectors with one scaling factor, uniformly random number in (0, 1), is used in the mutation, but in this work, three difference vectors between the tournament best, better and worst of the selected vectors with corresponding three scaling factors, which are independently generated according to uniform distribution in (0, 1), are used in the mutation scheme. (4) the triangular mutation rule is only used in this work, but in the previous work ([Mohamed 2015a](#)), the triangular mutation strategy is embedded into the DE algorithm and combined with the basic mutation strategy DE/rand/1/bin through a non-linear decreasing probability rule. (5) In the previous work ([Mohamed 2015a](#)) a restart mechanism based on Random Mutation and modified BGA mutation is used to avoid stagnation or premature convergence, whereas this work does not. The rest of the paper is organized as follows: “Problem formulation and constraint handling” section presents the formulation of the COPs. “Differential evolution” section introduces the DE algorithm. The fourth section proposed NDE is presented in details. “Experimental results and analysis” section reports on the computational results of testing benchmark functions and some engineering and mechanical optimization problems. Besides, the comparison with other techniques is discussed. Finally, in “Conclusion” section, the paper is concluded and some possible future research is suggested.

Problem formulation and constraint handling

In general, constrained optimization problem can be expressed as follows (without loss of generality minimization is considered here) ([Mohamed and Sabry 2012](#)):

$$\text{Minimize } f(\vec{x}), \vec{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n \quad (1)$$

Subject to:

$$g_j(\vec{x}) \leq 0, \quad j = 1, \dots, q \quad (2)$$

$$h_j(\vec{x}) = 0, \quad j = q + 1, \dots, m \quad (3)$$

where $\vec{x} \in \Omega \subseteq S$, Ω is the feasible region, and S is an n -dimensional rectangular space in \mathfrak{R}^n defined by the parametric constraints $l_i \leq x_i \leq u_i, 1 \leq i \leq n$ where l_i and u_i are lower and upper bounds for a decision variable x_i , respectively. For an inequality constraint that satisfies $g_j(\vec{x}) = 0 (j \in 1, \dots, q)$ at any point $\vec{x} \in \Omega$, we say it is active at \vec{x} . All equality constraints $h_j(\vec{x}) (j = q + 1, \dots, m)$ are considered active at all points of Ω . Most constraint-handling approaches used with EAs tend to deal only with inequality constraints. Therefore equality constraints are transformed into inequality constraints of the form $|h_j(\vec{x})| - \varepsilon \leq 0$ where ε is the tolerance allowed (a very small value). In order to handle constraints, we use Deb’s constraint handling procedure. Deb (2000) proposed a new efficient feasibility-based rule to handle constraint for genetic algorithm where pairwise solutions are compared using the following criteria:

- Between two feasible solutions, the one with the highest fitness values wins.
- If one solution is feasible and the other one is infeasible, the feasible solution wins.
- If both solutions are infeasible, the one with the lowest sum of constraint violation is preferred.

As a result, Deb (2000) has introduced the superiority of feasible solutions selection procedure based on the idea that any individual in a constrained search space must first comply with the constraints and then with the objective function. Practically, the Deb’s selection criterion does not need to fine-tune any parameter. Typically, from the problem formulation above, there are m constraints and hence the amount of constraint violation for an individual is represented by a vector of m dimensions. Using a tolerance (ε) allowed for equality constraints, the constraint violation of a decision vector or an individual \vec{x} on the j th constraint is calculated by

$$cv_j(\vec{x}) = \begin{cases} \max(0, g_j(\vec{x})), & j = 1, \dots, q \\ \max(0, |h_j(\vec{x})| - \varepsilon), & j = q + 1, \dots, m \end{cases} \quad (4)$$

If a decision vector or an individual \vec{x} satisfies the j th constraint, $cv_j(\vec{x})$ is set to zero, otherwise it is greater than zero. Thus, in order to consider all the constraints at the same time or to treat each constraint equally, each constraint violation is then normalized by dividing it by the largest violation of that constraint in the population (Venkatraman and Yen 2005). Thus, the maximum violation of each constraint in the population is given by

$$cv_{\max}^j = \max_{\vec{x} \in S} cv_j(\vec{x}) \quad (5)$$

These maximum constraint violation values are used to normalize each constraint violation. The normalized con-

straint violations are added together to produce a scalar constraint violation $cv(\vec{x})$ for that individual which takes a value between 0 and 1

$$cv(\vec{x}) = \frac{1}{m} \sum_{j=1}^m \frac{cv_j(\vec{x})}{cv_{\max}^j} \quad (6)$$

Differential evolution

This section provides a brief summary of the basic Differential Evolution (DE) algorithm. In simple DE, generally known as DE/rand/1/bin (Storn and Price 1997; Fan and Lampinen 2003), an initial random population, denoted by P , consists of NP individual. Each individual is represented by the vector $x_i = (x_{1,i}, x_{2,i}, \dots, x_{D,i})$, where D is the number of dimensions in solution space. Since the population will be varied with the running of evolutionary process, the generation times in DE are expressed by $G = 0, 1, \dots, G_{\max}$, where G_{\max} is the maximal time of generations. For the i th individual of P at the G generation, it is denoted by $x_i^G = (x_{1,i}^G, x_{2,i}^G, \dots, x_{D,i}^G)$. The lower and upper bounds in each dimension of search space are respectively recorded by $x_L = (x_{1,L}, x_{2,L}, \dots, x_{D,L})$ and $x_U = (x_{1,U}, x_{2,U}, \dots, x_{D,U})$. The initial population P^0 is randomly generated according to a uniform distribution within the lower and upper boundaries (x_L, x_U) . After initialization, these individuals are evolved by DE operators (mutation and crossover) to generate a trial vector. A comparison between the parent and its trial vector is then made to select the vector which should survive to the next generation (Das and Suganthan 2011). DE steps are discussed below:

Initialization

In order to establish a starting point for the optimization process, an initial population P^0 must be created. Typically, each j th component ($j = 1, 2, \dots, D$) of the i th individuals ($i = 1, 2, \dots, NP$) in the P^0 is obtained as follows:

$$x_{j,i}^0 = x_{j,L} + \text{rand}(0, 1) \cdot (x_{j,U} - x_{j,L}) \quad (7)$$

where $\text{rand}(0, 1)$ returns a uniformly distributed random number in $[0, 1]$.

Mutation

At generation G , for each target vector x_i^G , a mutant vector v_i^G is generated according to the following:

$$v_i^G = x_{r_1}^G + F \cdot (x_{r_2}^G - x_{r_3}^G), r_1 \neq r_2 \neq r_3 \neq i \quad (8)$$

With randomly chosen indices $r_1, r_2, r_3 \in \{1, 2, \dots, NP\}$. F is a real number to control the amplification of the difference vector $(x_{r_2}^G - x_{r_3}^G)$. According to (Storn and Price 1997), the range of F is in $[0, 2]$. In this work, If a component of a mutant vector violates search space, the value of this component is generated a new using (7).

Crossover

There are two main crossover types, binomial and exponential. In the binomial crossover, the target vector is mixed with the mutated vector, using the following scheme to yield the trial vector u_i^G .

$$u_{j,i}^G = \begin{cases} v_{j,i}^G, & \text{if } (rand_{j,i} \leq CR \text{ or } j = j_{rand}) \\ x_{j,i}^G, & \text{otherwise} \end{cases} \quad (9)$$

where $rand_{j,i}$ ($i \in [1, NP]$ and $j \in [1, D]$) is a uniformly distributed random number in $[0, 1]$, $CR \in [0, 1]$ called the crossover rate that controls how many components are inherited from the mutant vector, j_{rand} is a uniformly distributed random integer in $[1, D]$ that makes sure at least one component of trial vector is inherited from the mutant vector.

Selection

DE adapts a greedy selection strategy. If and only if the trial vector u_i^G yields as good as or a better fitness function value than x_i^G , then u_i^G is set to x_i^{G+1} . Otherwise, the old vector x_i^G is retained. The selection scheme is as follows (for a minimization problem):

$$x_i^{G+1} = \begin{cases} u_i^G, & f(u_i^G) \leq f(x_i^G) \\ x_i^G, & \text{otherwise} \end{cases} \quad (10)$$

A detailed description of standard DE algorithm is given in Fig. 1.

The proposal: NDE

In this section, a novel DE algorithm NDE and the constraint-handling technique used in this research are described and they explain the pseudo-code of the algorithm in details.

Triangular mutation scheme

DE/rand/1 is the fundamental mutation strategy developed by (Storn and Price 1997; Price et al. 2005), and it is reported to be the most successful and widely used scheme in the literature (Brest et al. 2006). Obviously, in this strategy,

the three vectors are chosen from the population at random for mutation and the base vector is then selected at random among the three. The other two vectors form the difference vector that is added to the base vector. Consequently, it is able to maintain population diversity and global search capability with no bias to any specific search direction, but it slows down the convergence speed of DE algorithms (Qin et al. 2009). DE/rand/2 strategy is the same as the former scheme with extra two vectors that forms another difference vector, which might lead to better perturbation than one-difference-vector-based strategies (Qin et al. 2009). Furthermore, it can provide more various differential trial vectors than the DE/rand/1/bin strategy which increases its exploration ability of the search space. On the other hand, greedy strategies like DE/best/1, DE/best/2 and DE/current-to-best/1 incorporate the information of the best solution found so far in the evolutionary process to increase the local search tendency that leading to fast convergence speed of the algorithm (Qin et al. 2009). However, the diversity of the population and exploration capability of the algorithm may deteriorate or may completely diminish through a very small number of generations i.e. at the beginning of the optimization process that causes problems such as stagnation and/or premature convergence. Consequently, in order to overcome the shortcomings of both types of mutation strategies, most of the recent successful algorithms utilize the strategy candidate pool that combines different trail vector generation strategies that have diverse characteristics and distinct optimization capabilities, with different control parameter settings to deal with a variety of problems with different features at different stages of evolution (Qin et al. 2009; Mallipeddi et al. 2011; Wang et al. 2011). Contrarily, taking into consideration the weakness of existing greedy strategies, Zhang and Sanderson (2009) introduced a new differential evolution (DE) algorithm, named JADE, to improve the optimization performance by implementing a new mutation strategy called “DE/current-to- p best” with optional external archive and updating control parameters in an adaptive manner. Consequently, proposing new mutations strategies that can considerably improve the search capability of DE algorithms and increase the possibility of achieving promising and successful results in complex and large scale optimization problems is still an open challenge for evolutionary computation research. Therefore, this research uses a new triangular mutation rule with a view of balancing the global exploration ability and the local exploitation tendency and enhancing the convergence rate of the algorithm. The proposed mutation strategy is based on the convex combination vector of the triplet defined by the three randomly chosen vectors and three difference vectors between the tournament best, better and worst of the selected vectors. The proposed mutation vector is generated in the following manner:

Fig. 1 Description of standard DE algorithm. rand [0, 1) is a function that returns a real number between 0 and 1. randint (min, max) is a function that returns an integer number between min and max. NP, G_{max} , CR and F are user-defined parameters. D is the dimensionality of the problem

```

01. Begin
02. G=0
03. Create a random initial population  $\vec{x}_i^G \forall i, i = 1, \dots, NP$ 
04. Evaluate  $f(\vec{x}_i^G) \forall i, i = 1, \dots, NP$ 
05. For G=1 to Gmax Do
06.   For i=1 to NP Do
07.     Select randomly  $r1 \neq r2 \neq r3 \neq i \in [1, NP]$ 
08.      $j_{rand} = \text{randint}(1, D)$ 
09.     For j=1 to D Do
10.       If (rand $_{j,i}$ [0,1) < CR or  $j = j_{rand}$ ) Then
11.          $u_{i,j}^G = x_{r1,j}^G + F \cdot (x_{r2,j}^G - x_{r3,j}^G)$ 
12.       Else
13.          $u_{i,j}^G = x_{i,j}^G$ 
14.       End If
15.     End For
16.     If ( $f(\vec{u}_i^G) \leq f(\vec{x}_i^G)$ ) Then
17.        $\vec{x}_i^{G+1} = \vec{u}_i^G$ 
18.     Else
19.        $\vec{x}_i^{G+1} = \vec{x}_i^G$ 
20.     End If
21.   End For
22.   G=G+1
23. End For
24. End

```

$$v_i^{G+1} = \vec{x}_c^G + F1 \cdot (x_{best}^G - x_{better}^G) + F2 \cdot (x_{best}^G - x_{worst}^G) + F3 \cdot (x_{better}^G - x_{worst}^G) \quad (11)$$

where \vec{x}_c^G is a convex combination vector of the triangle and F1, F2 and F3 are the mutation factors that are associated with x_i and are independently generated according to uniform distribution in (0, 1) and x_{best}^G , x_{better}^G and x_{worst}^G are the tournament best, better and worst three randomly selected vectors, respectively. The convex combination vector \vec{x}_c^G of the triangle is a linear combination of the three randomly selected vectors and is defined as follows:

$$\vec{x}_c^G = w_1 \cdot x_{best} + w_2 \cdot x_{better} + w_3 \cdot x_{worst} \quad (12)$$

where the real weights w_i satisfy $w_i \geq 0$ and $\sum_{i=1}^3 w_i = 1$. Where the real weights w_i are given by $w_i = p_i / \sum_{i=1}^3 p_i$, $i = 1, 2, 3$. Where $p_1 = 1$, $p_2 = \text{rand}(0.75, 1)$ and $p_3 = \text{rand}(0.5, p(2))$, $\text{rand}(a, b)$ is a function that returns a real number between a and b. where a and b are not included. For

constrained optimization problems at any generation $g > 1$, the tournament selection of the three randomly selected vectors and assigning weights follows one of the following three scenarios that may exist through generations. Without loss of generality, we only consider minimization problems:

Scenario 1: If the three randomly selected vectors are feasible, then sort them in ascending order according to their objective function values and assign w_1, w_2, w_3 to x_{best}^G, x_{better}^G and x_{worst}^G , respectively.

Scenario 2: If the three randomly selected vectors are infeasible, then sort them in ascending order according to their constraint violations (CV) values and assign w_1, w_2, w_3 to x_{best}^G, x_{better}^G and x_{worst}^G , respectively.

Scenario 3: If the three randomly selected vectors are mixed (feasible and infeasible), then the vectors are sorted by using the three criteria: (a) Sort feasible vectors in front of infeasible solutions (b) Sort feasible solutions according to their objective function values (c) Sort infeasible solutions according to their constraint violations. Accordingly, assign

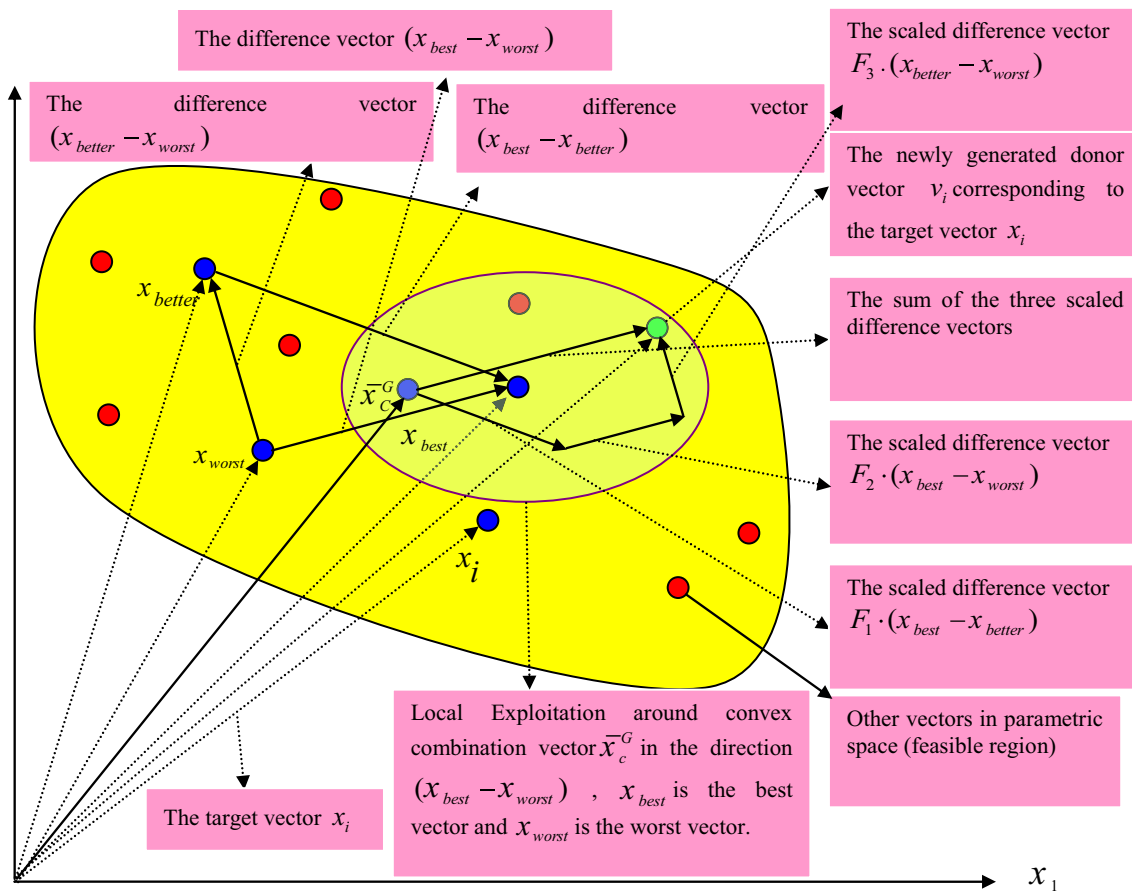


Fig. 2 An illustration of the new triangular mutation scheme in two-dimensional parametric space (local exploitation)

w_1, w_2, w_3 to x_{best}^G, x_{better}^G and x_{worst}^G , respectively. Obviously, from mutation Eqs. (11) and (12), it can be observed that the incorporation of the objective function value and constraints violation in the mutation scheme has two benefits. Firstly, the perturbation part of the mutation is formed by the three sides of the triangle in the direction of the best vector among the three randomly selected vectors. Therefore, the directed perturbation in the proposed mutation resembles the concept of gradient as the difference vector is oriented from the worst to the best vectors (Feoktistov 2006). Thus, it is considerably used to explore the landscape of the objective function being optimized in different sub-region around the best vectors within constrained search-space through the optimization process. Secondly, the convex combination vector \bar{x}_c^G is a weighted sum of the three randomly selected vectors where the best vector has the highest contribution. Therefore, \bar{x}_c^G is extremely affected and biased by the best vector more than the remaining two vectors. Consequently, the global solution can be easily reached if all vectors follow the direction of the best vectors together with the fact that they also follow the opposite direction of the worst vectors among the randomly selected vectors. Indeed, the new mutation process exploits the nearby region of each \bar{x}_c^G in

the direction of $(x_{best}^G - x_{worst}^G)$ for each mutated vector. In a nutshell, it concentrates on the exploitation of some sub-regions of the search space. Thus, it has better local search tendency so it accelerates the convergence speed of the proposed algorithm. Besides, the global exploration ability of the algorithm is significantly enhanced by forming many different sizes and shapes of triangles in the feasible region through the optimization process. Thus, the proposed directed mutation balances both the global exploration capability and the local exploitation tendency. Illustrations of the local exploitation and global exploration capabilities of the new triangular mutation are illustrated in Figs. 2 and 3, respectively.

Constraint handling

In this paper, Deb’s feasibility rules (Deb 2000) are used to handle the constraints. As aforementioned in “Problem formulation and constraint handling” section, in the constraint-handling technique, the equality constraints are transformed to inequality constraints by a tolerance value ϵ . In the experiments, the tolerance value ϵ is adopted like (Mezura-Montes and Coello 2005) and is used in Yang

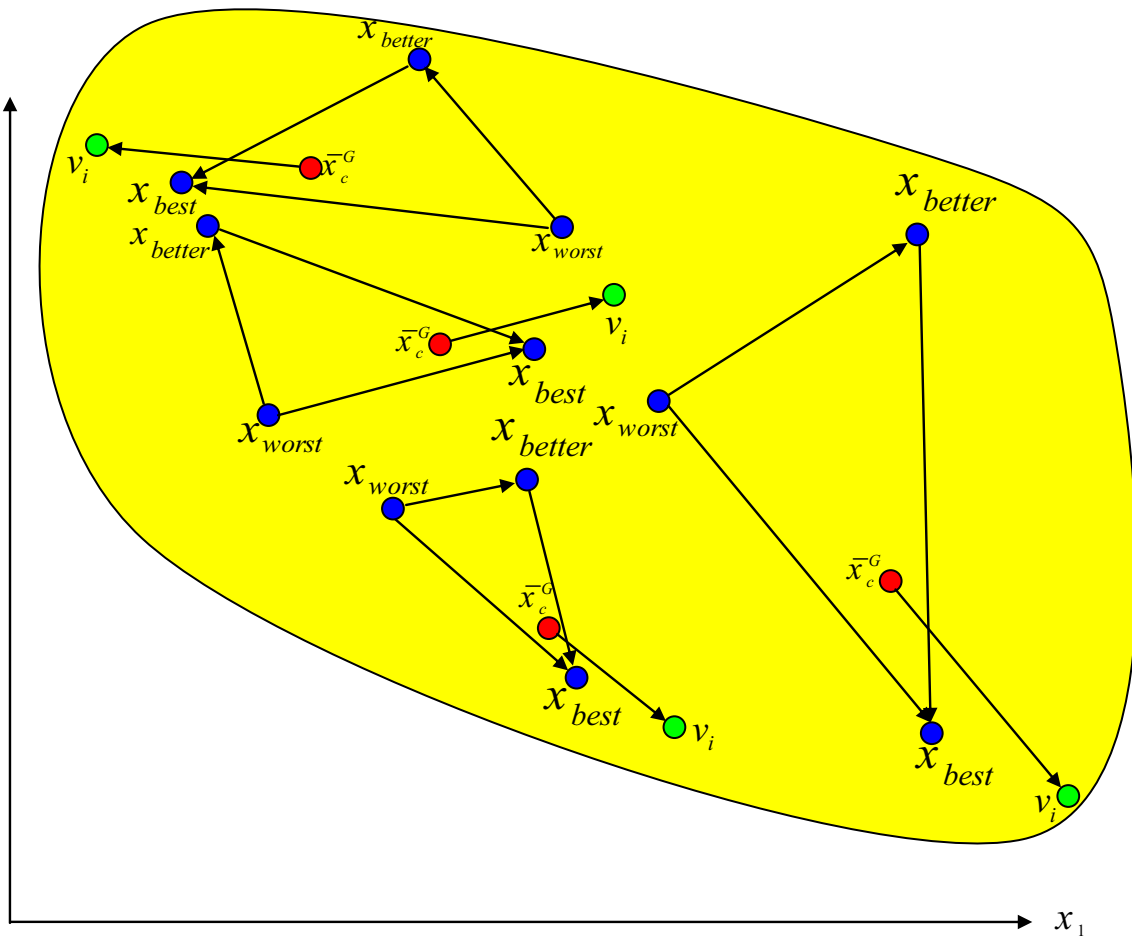


Fig. 3 An illustration of the new triangular mutation scheme with collection of convex combinations vectors and the newly generated donor vectors v_i corresponding to the target vectors x_i in two-dimensional parametric space (global exploration)

(2009). The parameter ε decreases from generation to generation using the following equation:

$$\varepsilon(g + 1) = \begin{cases} \frac{\varepsilon(g)}{1.0168}, & \text{if } \varepsilon > 10^{-4} \\ 10^{-4}, & \text{otherwise.} \end{cases} \quad (13)$$

where g is the generation number and the initial $\varepsilon(0)$ is set to 5 with all problems with exception to $g17$ and $g23$, it is set to 10. Regarding the boundary-handling method, the re-initialization method is used (see Eq. 7), i.e. when one of the decision variables violates its boundary constraint, it is generated within the uniform distribution within the boundary. The pseudo-code of NDE is presented in Fig. 4.

Experimental results and analysis

In this section, the performance of NDE is evaluated by performing comprehensive experiments using benchmark functions, engineering and mechanical design problems.

Benchmark test functions and parameter settings

In this paper, 24 well-known benchmark test functions from CEC 2006 mentioned in (Liang et al. 2006) are used. The features of 24 benchmark functions are listed in Table 1. Obviously, from Table 1, n is the number of decision variables. These test functions contain several types of objective function shapes (e.g. linear, non-linear, and quadratic) as well as different types of constraints [e.g. linear inequality (LI), non-linear equality (NE), and non-linear-inequality (NI)], LI denotes the number of linear inequalities, NE denotes the number of non-linear equation, NI denotes the number of non-linear inequalities, a represents the number of active constraints at the global optimum solution, and $f(x^*)$ is the objective function value of the best known solution x^* . $\rho = |F|/|S|$ is the feasibility ratio between the feasible region and whole search space. Note that 11 test functions ($g03, g05, g11, g13, g14, g15, g17, g20, g21, g22,$ and $g23$) contain equality constraints. The $f(x^*)$ of $g17$ in Table 1 is different from that in (Liang et al. 2006)


```

01. Begin
02. G=0 ,
03. Create a random initial population  $\vec{x}_i^G \forall i, i = 1, \dots, NP$ , CR= 0.95 .
04. Evaluate  $f(\vec{x}_i^G)$ ,  $cv(\vec{x}_i^G)$ ,  $\forall i, i = 1, \dots, NP$ 
05. For G=1 to Gmax Do
06. Compute the Factor of the tolerance value ( $\varepsilon$ ) using equation (13).
07.   For i=1 to NP Do
08.      $F_1 = \text{rand}[0,1]$ ,  $F_2 = \text{rand}[0,1]$ ,  $F_3 = \text{rand}[0,1]$ 
09.      $j_{\text{rand}} = \text{randint}(1,D)$ 
10.     Select randomly  $r1 \neq r2 \neq r3 \neq i \in [1, NP]$ 
11.     For j=1 to D Do
12.       If ( $\text{rand}_{i,j} \in [0,1] < \text{CR}$  or  $j = j_{\text{rand}}$ ) Then
13.         Determine the tournament  $\mathcal{X}_{\text{best}}^G$ ,  $\mathcal{X}_{\text{better}}^G$  and  $\mathcal{X}_{\text{worst}}^G$  based on  $f(\vec{x}_i^G)$ ,  $cv(\vec{x}_i^G)$ ,  $i = 1, 2, 3$ ,
14.         (three randomly selected vectors) using one of three scenarios and compute  $\vec{x}_c^G$  according to eq.(11).
15.          $u_{i,j}^G = \vec{x}_{c,j}^G + F_1(\mathcal{X}_{\text{best},j}^G - \mathcal{X}_{\text{better},j}^G) + F_2(\mathcal{X}_{\text{best},j}^G - \mathcal{X}_{\text{worst},j}^G) + F_3(\mathcal{X}_{\text{better},j}^G - \mathcal{X}_{\text{worst},j}^G)$ 
16.         Else
17.            $u_{ij}^G = \mathcal{X}_{ij}^G$ 
18.         End If
19.       End For
20.       If ( $\vec{u}_i^G$  is better than  $\vec{x}_i^G$  (based on Deb's selection criteria)) Then
21.          $\vec{x}_i^{G+1} = \vec{u}_i^G$ 
22.       Else
23.          $\vec{x}_i^{G+1} = \vec{x}_i^G$ 
24.       End If
25.     Evaluate  $f(\vec{x}_i^{G+1})$  and  $cv(\vec{x}_i^{G+1})$ ,  $\forall i, i = 1, \dots, NP$ .
26.   End For
27.   G=G+1
28. End For
End

```

Fig. 4 Description of NDE algorithm

because NDE finds a better solution for test function g17 than the best known solutions from (Liang et al. 2006) where $x^* = (201.7844624935499400, 99.9999999999995310, 383.0710348527716700, 419.9999999999993200, -10.90-769795937907600000, 0.073148231208428699)$ with $f(x^*) = 8853.5338748064842000$. The three main control parameters of NDE algorithm, population size NP is 200, crossover rate Cr is fixed 0.95 with exception to g02 where Cr is 0.25 and the scale factors $F1$, $F2$ and $F3$ are uniformly random numbers in (0, 1) as mentioned before. The maximum number of function evaluations (FEs) for all benchmark problems are set to 240,000 (Gong et al. 2014; Elsayed et al. 2013). The experiments were executed on an Intel Pentium core i7 processor 1.6 GHz and 4 GB-RAM. NDE algorithm is coded and realized in MATLAB. For each problem, 30 independent runs are performed and statistical results are provided includ-

ing the best, median, mean, worst results and the standard deviation. Note that two functions (g20 and g22) are highly constrained problems since they contain 14 and 19 and contain 20 and 22 decision variables, respectively. NDE cannot obtain a feasible solution over all 30 runs for both functions. Therefore, in the following experiments, the results of these two functions are not reported and the analysis is based on 22 test problems as, to the best of our knowledge, no feasible solutions have been found so far for this test function. On the other hand, to compare the solution quality from a statistical angle of different algorithms and to check the behavior of the stochastic algorithms (García et al. 2009), the results are compared using three non-parametric statistical hypothesis tests: (i) the Friedman test (to obtain the final rankings of different algorithms for all functions); (ii) Iman-Davenport test (to check the differences between all algorithms for all

Table 1 Main characteristics of 24 benchmark problems

Problem	n	Type of f	ρ (%)	LI	NI	LE	NE	a	$f(x^*)$
g01	13	Quadratic	0.0003	9	0	0	0	6	-15.0000000000
g02	20	Nonlinear	99.9971	0	2	0	0	1	-0.8036191042
g03	10	Polynomial	0.0000	0	0	0	1	1	-1.0005001000
g04	5	Quadratic	52.1230	0	6	0	0	2	-30665.53867178
g05	4	Cubic	0.0000	2	0	0	3	3	5126.4967140071
g06	2	Cubic	0.0066	0	2	0	0	2	-6961.813875580
g07	10	Quadratic	0.0003	3	5	0	0	6	24.3062090681
g08	2	Nonlinear	0.8560	0	2	0	0	0	-0.0958250415
g09	7	Polynomial	0.5121	0	4	0	0	2	680.630057374402
g10	8	Linear	0.0010	3	3	0	0	6	7049.2480205286
g11	2	Quadratic	0.0000	0	0	0	1	1	0.7499000000000
g12	3	Quadratic	4.7713	0	1	0	0	0	-1.000000000000
g13	5	Nonlinear	0.0000	0	0	0	3	3	0.0539415140
g14	10	Nonlinear	0.0000	0	0	3	0	3	-47.7648884595
g15	3	Quadratic	0.0000	0	0	1	1	2	961.7150222899
g16	5	Nonlinear	0.0204	4	34	0	0	4	-1.9051552586
g17	6	Nonlinear	0.0000	0	0	0	4	4	8853.5396748064
g18	9	Quadratic	0.0000	0	13	0	0	6	-0.8660254038
g19	15	Nonlinear	33.4761	0	5	0	0	0	32.6555929502
g20	24	Linear	0.0000	0	6	2	12	16	0.2049794002
g21	7	Linear	0.0000	0	1	0	15	6	193.7245100700
g22	22	Linear	0.0000	0	1	8	11	19	236.4309755040
g23	9	Linear	0.0000	0	2	3	1	6	-400.0551000000
g24	2	Linear	79.6556	0	2	0	0	2	-5.5080132716

functions); and (iii) multi-problem Wilcoxon signed-rank test at a 0.05 significance level (Gong et al. 2013). Given k algorithms and N data sets, the Friedman test ranks the performance of algorithms for each data set (in case of equal performance, average ranks are assigned) and tests if the measured average ranks $R_j = (1/N) \sum_{i=1}^N r_i^j$ (r_i^j as the rank of the j th algorithm on the i th data set) are significantly different from the mean rank. Under the null hypothesis, which states that all the algorithms behave similarly and thus their rank should be equal, the Friedman statistic used is $\chi_F^2 = \frac{12N}{K(K+1)} \left[\sum_{j=1}^k R_j^2 - \frac{k(k+1)^2}{4} \right]$ which follows a χ^2 distribution with $k - 1$ degrees of freedom, and the Iman-Davenport statistic is $F_F = \frac{(N-1)\chi_F^2}{N(k-1) - \chi_F^2}$ which is distributed according to Fischer's F -distribution with $k - 1$ and $(k - 1)(N - 1)$ degrees of freedom (Demšar 2006). Regarding multi-problem Wilcoxon signed-rank test, R^+ denotes the sum of ranks for the test problems in which the first algorithm performs better than the second algorithm (in the first column), and R^- represents the sum of ranks for the test problems in which the first algorithm performs worse than the second algorithm (in the first column). Larger ranks indicate larger performance discrepancy. The numbers in Better,

Equal and worse columns denote the number of problems in which the first algorithm is better than, equal or worse than the second algorithm. As a null hypothesis, it is assumed that there is no significant difference between the mean results of the two samples. Whereas the alternative hypothesis is that there is significance in the mean results of the two samples, the number of test problems $N = 22$ and 5% significance level. use the p value and compare it with the significance level. Reject the null hypothesis if the p -value is less than or equal to the significance level (5%). Based on the result of the test, one of three signs (+, -, and \approx) is assigned for the comparison of any two algorithms (shown in the last column), where (+) sign means the first algorithm is significantly better than the second, (-) sign means the first algorithm is significantly worse than the second, and (\approx) sign means that there is no significant difference between the two algorithms. All the p values in this paper were computed using SPSS (the version is 20.00).

Results of the proposed approach (NDE)

The statistical results of the NDE on the benchmarks are summarized in Table 2. It includes the known optimal solution for each test problem and the obtained best, median,

Table 2 The results of NDE on the benchmarks, where “–” means no feasible solution is found

Problem	Optimal	Best	Median	Mean	Worst	SD
g01	−15.000000	−15.000000	−15.000000	−15.000000	−15.000000	0.00E+00
g02	−0.803619	−0.803480	−0.801962	−0.801809	−0.800495	5.10E−04
g03	−1.000000	−1.0005001	−1.0005001	−1.0005001	−1.0005001	0.00E+00
g04	−30665.539	−30665.539	−30665.539	−30665.539	−30665.539	0.00E+00
g05	5126.49671	5126.49671	5126.49671	5126.49671	5126.49671	0.00E+00
g06	−6961.8138	−6961.813875	−6961.813875	−6961.813875	−6961.813875	0.00E+00
g07	24.306209	24.306209	24.306209	24.306209	24.306209	1.35E−014
g08	−0.095825	−0.095825	−0.095825	−0.095825	−0.095825	0.00E+00
g09	680.630057	680.630057	680.630057	680.630057	680.630057	0.00E+00
g10	7049.248020	7049.248020	7049.248020	7049.248020	7049.248020	3.41E−09
g11	0.750000	0.749999	0.749999	0.749999	0.749999	0.00E+00
g12	−1.000000	−1.000000	−1.000000	−1.000000	−1.000000	0.00E+00
g13	0.0539415	0.0539415	0.0539415	0.0539415	0.0539415	0.00E+00
g14	−47.7648885	−47.7648885	−47.7648885	−47.7648885	−47.7648885	5.14E−15
g15	961.7150223	961.7150223	961.7150223	961.7150223	961.7150223	0.00E+00
g16	−1.90515525	−1.90515525	−1.90515525	−1.90515525	−1.90515525	0.00E+00
g17	8853.533874806	8853.533874806	8853.533874806	8853.533874806	8853.533874806	0.00E+00
g18	−0.8660254	−0.8660254	−0.8660254	−0.8660254	−0.8660254	0.00E+00
g19	32.655593	32.65559377	32.65560741	32.65562603	32.65570220	3.73E−05
g20	–	–	–	–	–	–
g21	193.7245101	193.7245101	193.7245101	193.7245101	193.7245101	6.26E−011
g22	–	–	–	–	–	–
g23	−400.05510000	−400.05510000	−400.05510000	−400.05510000	−400.05510000	3.45E−09
g24	−5.50801327	−5.50801327	−5.50801327	−5.50801327	−5.50801327	0.00E+00

mean, worst values and the standard deviations. Additionally, Figs. 5, 6, 7, 8, 9 and 10 show the convergence graphs of $\log_{10}(f(x) - f(x^*))$ over FEs at the median run for each test problem with 240,000 FEs, where x^* is the best solution found. As shown in Table 2, NDE is able to find the global optimal solution consistently in 20 out of 22 test functions over 30 runs with the exception of test functions (g02 and g19). With respect to these test functions, although the optimal solutions are not consistently found, the best result achieved is very close to the global optimal solution which can be verified by the very small standard deviation. Additionally, it is able to provide the competitive results. NDE is also able to find new solution to test functions g17 which is better than the best known solutions from (Liang et al. 2006). From Figs. 5, 6, 7, 8, 9 and 10, it can be obviously seen that a fast convergence (less than 50,000 FES) can be achieved for the test functions (g04, g06, g08, g09, g12, g16, g18, g24). Moreover, for test functions (g01, g03, g05, g07, g10, g11, g13, g14, g15, g17), less than 150,000 FES are needed. In addition, by using about 200,000 FES, the convergence is achieved for (g19, g21, and g23). However, among those problems, g02 is multi-modal, that contains many local

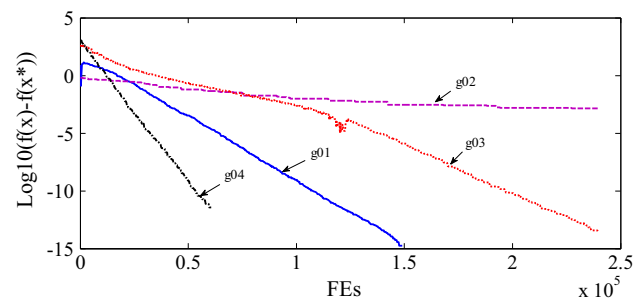


Fig. 5 Convergence graph for g01–g04

optima near the global minimum (Asafuddoula et al. 2014). Thus, all 240,000 FES are needed. From the above analysis, it can be concluded that NDE has a fast convergence speed for most of these 22 test functions. Accordingly, the main benefits of the new triangular mutation are the fast convergence speed and the extreme robustness which are the weak points of all evolutionary algorithms. Therefore, the proposed NDE algorithm is proven to be an effective and powerful approach to solve constrained global optimization problems.

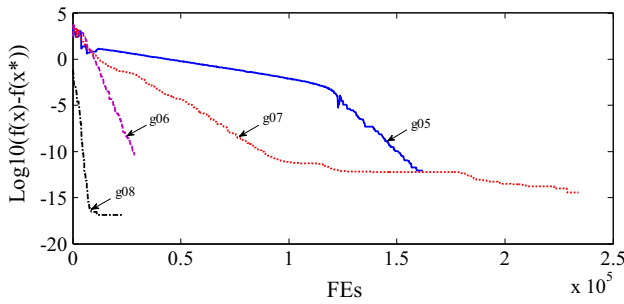


Fig. 6 Convergence graph for g05–g08

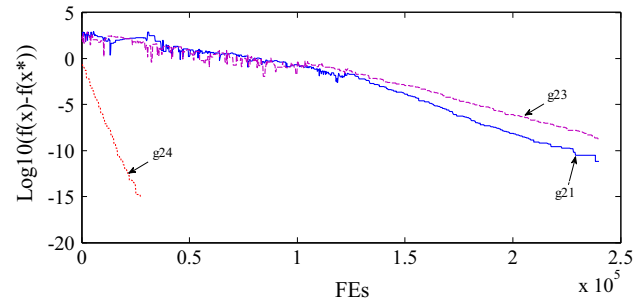


Fig. 10 Convergence graph for g21, g23 and g24

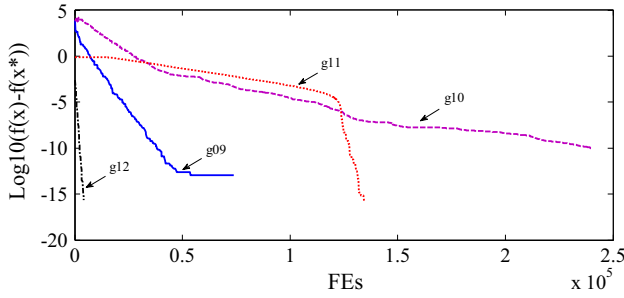


Fig. 7 Convergence graph for g09–g12

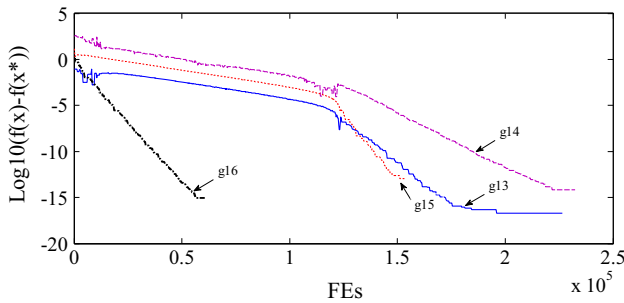


Fig. 8 Convergence graph for g13–g16

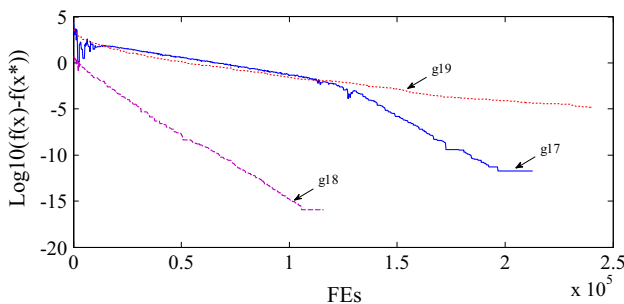


Fig. 9 Convergence graph for g17–g19

Comparisons with current state-of-the-art competitive DE approaches

The NDE was compared to seven state-of-the-art DE approaches for COPs that were all tested on benchmark functions presented in CEC’2006 mentioned in (Liang et al.

2006). These algorithms are rank-iMDDE (Gong et al. 2014), ISAMODE-CMA (Elsayed et al. 2013), SAMO-DE (Elsayed et al. 2011), MDEDP (Dong and Wang 2014), AH-DEa (Asafuddoula et al. 2014), DE-DPS (Sarker et al. 2014), and DECV (Mezura-Montes et al. 2010). The selected algorithms had the appropriate performances available to make this comparison possible and used the same maximum number of function evaluations (FEs) (240,000). The best, the median, the mean, the worst, and the standard deviation of the objective function values of the final solutions for each algorithm are listed in Table 3. A result in boldface indicates the best result or the global optimum. NA means not available. The results provided by these approaches were directly taken from the original references for each approach. According to the results shown in Table 3, it can be obviously seen that NDE consistently exhibits high quality results in all benchmark problems as compared with other seven DE algorithms. For g02, only DE-DPS was able to find the optimal solution consistently. Besides, NDE is slightly worse than ISAMODE-CMA, MDEDP and DE-DPS in function g19. However, MDEDP, AH-DEa, DE-DPS and DECV were unable to find the global optimal solution for g17 consistently.

In order to perform the Friedman test, we first rank all the algorithms according to the mean values in Table 4. From Table 4, NDE and DE-DPS get the first ranking among the eight algorithms on 22 test functions. Figure 11 shows the average ranking of the considered DE algorithm based on the Friedman test. For $N = 22$ and $k = 8$, the Friedman statistic is $\chi_F^2 = \frac{12 \times 22}{8 \times 9} [(3.79545^2 + 3.90909^2 + 4.04545^2 + 5.06818^2 + 4.72727^2 + 4.63636^2 + 3.84091^2 + 5.97727^2) - \frac{8 \times 9^2}{4}] = 14.89462$.

And the Iman-Davenport statistic is $F_F = \frac{21 \times 14.89462}{22 \times 7 - 14.89462} = 2.2486$.

F -distribution with $(k - 1)$ and $(k - 1)(N - 1)$ degrees of freedom. The critical value of $F(7, 147)$ for $\alpha = 0.05$ is 2.07. Therefore, the null-hypothesis that ranks do not significantly differ is rejected. Thus, it can be concluded that there is a significant difference between the performances of the algorithms. Additionally, due to the importance of the multiple-problem statistical analysis, Table 5 summarizes

Table 3 Statistical features of the results obtained by of NDE, rank-iMDDE, ISAMODE-CMA, SAMO-DE, MDEDP, AH-DEa, DE-DPS and DECV

Problem	Features	NDE	rank-iMDDE	ISAMODE-CMA	SAMO-DE	MDEDP	AH-DEa	DE-DPS	DECV
g01	Best	-15.000000	-15.000000	-15.000000	-15.000000	-15.000000	-15.000000	-15.000000	-15.0000
	Median	-15.000000	-15.000000	NA	NA	-15.000000	NA	NA	NA
	Mean	-15.000000	-15.000000	-15.000000	-15.000000	-15.000000	-15.000000	-15.000000	-15.000000
	Worst	-15.000000	-15.000000	NA	NA	-15.000000	NA	NA	-14.855
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	-13.000
	Best	-0.803480	-0.80361905	-0.8036191	-0.8036191	-0.803616	-0.8036191	-0.8036191	4.59E-01
g02	Median	-0.801962	-0.80361882	NA	NA	-0.763716	NA	NA	NA
	Mean	-0.801809	-0.80202118	-0.79244	-0.79873521	-0.757713	-0.8007	-0.8036191	-0.569458
	Worst	-0.800495	-0.77174939	NA	NA	-0.616138	NA	NA	-0.238203
	SD	5.10E-04	4.57E-03	2.80E-02	8.80005E-03	1.90E-02	3.912E-03	9.06255E-08	9.51E-02
	Best	-1.0005001	-1.0005001	-1.0005	-1.0005	-1.0005	-1.0005	-1.0005	-0.461
	Median	-1.0005001	-1.0005001	NA	NA	-1.0005	NA	NA	NA
g03	Mean	-1.0005001	-1.0005001	-1.0005	-1.0005	-1.0005	-1.0005	-1.0005	-0.134
	Worst	-1.0005001	-1.0005001	NA	NA	-1.0005	NA	NA	-0.002
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.90E-09	0.00E+00	0.00E+00	1.17E-01
	Best	-30665.539	-30665.539	-30665.539	-30665.5386	-30665.53867	-30665.5386	-30665.539	-30.665.539
	Median	-30665.539	-30665.539	NA	NA	-30665.53867	NA	NA	NA
	Mean	-30665.539	-30665.539	-30665.539	-30665.5386	-30665.53867	-30665.5386	-30665.539	-30.665.539
g04	Worst	-30665.539	-30665.539	NA	NA	-30665.53867	NA	NA	-30.665.539
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.40E-11	0.00E+00	0.00E+00	1.56E-06
	Best	5126.49671	5126.49671	5126.49671	5126.497	5126.49671	5126.497	5126.497	5126.497
	Median	5126.49671	5126.49671	NA	NA	5126.49671	NA	NA	NA
	Mean	5126.49671	5126.49671	5126.49671	5126.497	5126.49671	5126.497	5126.497	5126.497
	Worst	5126.49671	5126.49671	NA	NA	5126.49671	NA	NA	5126.497
g05	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.00E-12	0.00E+00	0.00E+00	0.00E+00
	Best	-6961.81388	-6961.81388	-6961.81388	-6961.813875	-6961.814	-6961.813875	-6961.814	-6961.814
	Median	-6961.81388	-6961.81388	NA	NA	-6961.814	NA	NA	NA
	Mean	-6961.81388	-6961.81388	-6961.81388	-6961.813875	-6961.814	-6961.813875	-6961.814	-6961.814
	Worst	-6961.81388	-6961.81388	NA	NA	-6961.814	NA	NA	-6961.814
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.80E-12	0.00E+00	0.00E+00	0.00E+00

Table 3 continued

Problem	Features	NDE	rank-iMDDE	ISAMODE-CMA	SAMO-DE	MDEDP	AH-DEa	DE-DPS	DECV
g07	Best	24.306209	24.306209	24.3062	24.3062	24.306	24.3062	24.3062	24.306
	Median	24.306209	24.306209	NA	NA	24.306	NA	NA	NA
	Mean	24.306209	24.306209	24.3062	24.3096	24.306	24.3062	24.3062	24.794
	Worst	24.306209	24.306209	NA	NA	24.306	NA	NA	29.511
	SD	1.35E-014	0.00E+00	0.00E+00	1.5888E-03	4.60E-06	0.00E+00	0.00E+00	1.37E+00
g08	Best	-0.095825	-0.095825	-0.095825	-0.09582504	-0.095825	-0.09582504	-0.09582504	-0.095825
	Median	-0.095825	-0.095825	NA	NA	-0.095825	NA	NA	NA
	Mean	-0.095825	-0.095825	-0.095825	-0.09582504	-0.095825	-0.09582504	-0.09582504	-0.095825
	Worst	-0.095825	-0.095825	NA	NA	-0.095825	NA	NA	-0.095825
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.40E-17	0.00E+00	0.00E+00	4.23E-17
g09	Best	680.630057	680.630057	680.630	680.630	680.630	680.630	680.630	680.630
	Median	680.630057	680.630057	NA	NA	680.630	NA	NA	NA
	Mean	680.630057	680.630057	680.630	680.630	680.630	680.630	680.630	680.630
	Worst	680.630057	680.630057	NA	NA	680.630	NA	NA	680.630
	SD	0.00E+00	0.00E+00	0.00E+00	1.1567E-05	3.60E-10	0.00E+00	0.00E+00	3.45E-07
g10	Best	7049.248020	7049.248020	7049.24802	7049.24802	7049.249	7049.248020	7049.248020	7049.248
	Median	7049.248020	7049.248020	NA	NA	7049.257	NA	NA	NA
	Mean	7049.248020	7049.248020	7049.248020	7059.81345	7049.258	7049.248020	7049.248020	7103.548
	Worst	7049.248020	7049.248020	NA	NA	7049.311	NA	NA	7808.980
	SD	3.41E-09	0.00E+00	5.42E-06	7.856E+00	1.1E-02	1.688E-09	0.00E+00	1.48E+02
g11	Best	0.749999	0.749999	0.749999	0.749999	0.749999	0.749999	0.749999	0.75
	Median	0.749999	0.749999	NA	NA	0.749999	NA	NA	NA
	Mean	0.749999	0.749999	0.749999	0.749999	0.749999	0.749999	0.749999	0.75
	Worst	0.749999	0.749999	NA	NA	0.749999	NA	NA	0.75
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.40E-16	0.00E+00	0.00E+00	1.12E-16
g12	Best	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000
	Median	-1.000000	-1.000000	NA	NA	-1.000000	NA	NA	NA
	Mean	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000
	Worst	-1.000000	-1.000000	NA	NA	-1.000000	NA	NA	-1.000
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
g13	Best	0.0539415	0.0539415	0.053942	0.053942	0.0539415	0.053942	0.053942	0.059798
	Median	0.0539415	0.0539415	NA	NA	0.0539415	NA	NA	NA
	Mean	0.0539415	0.0539415	0.053942	0.053942	0.0539415	0.053942	0.053942	0.382401
	Worst	0.0539415	0.0539415	NA	NA	0.0539415	NA	NA	0.999094
	SD	0.00E+00	0.00E+00	0.00E+00	1.7541E-08	1.30E-12	0.00E+00	0.00E+00	2.68E-01

Table 3 continued

Problem	Features	NDE	rank-iMDDE	ISAMODE-CMA	SAMO-DE	MDEDP	AH-DEa	DE-DPS	DECV
g14	Best	-47.7648885	-47.7648885	-47.764888	-47.76489	-47.76489	-47.76	-47.7648885	-47.764888
	Median	-47.7648885	-47.7648885	NA	NA	-47.76489	NA	NA	NA
	Mean	-47.7648885	-47.7648885	-47.764888	-47.68115	-47.76487	-47.76	-47.7648885	-47.722542
	Worst	-47.7648885	-47.7648885	NA	NA	-47.76478	NA	NA	-47.036510
	SD	0.00E+00	0.00E+00	0.00E+00	4.043E-02	2.90E-05	3.894E-05	0.00E+00	1.52E-01
g15	Best	961.7150223	961.7150223	961.71502	961.71502	961.715022	961.71502	961.71502	961.7150223
	Median	961.7150223	961.7150223	NA	NA	961.715022	NA	NA	NA
	Mean	961.7150223	961.7150223	961.71502	961.71502	961.715022	961.71502	961.71502	961.7150223
	Worst	961.7150223	961.7150223	NA	NA	961.715022	NA	NA	961.7150223
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.90E-13	0.00E+00	0.00E+00	2.31E-13
g16	Best	-1.90515525	-1.90515525	-1.905155	-1.905155	-1.9051553	-1.905155	-1.905155	-1.905155
	Median	-1.90515525	-1.90515525	NA	NA	-1.9051553	NA	NA	NA
	Mean	-1.90515525	-1.90515525	-1.905155	-1.905155	-1.9051553	-1.905155	-1.905155	-1.905155
	Worst	-1.90515525	-1.90515525	NA	NA	-1.9051553	NA	NA	-1.905149
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.60E-16	0.00E+00	0.00E+00	1.10E-06
g17	Best	8853.533874	8853.539675	8853.5397	8853.5397	8853.5339	8853.5339	8853.5397	8853.541289
	Median	8853.533874	8853.539675	NA	NA	8853.5339	NA	NA	NA
	Mean	8853.533874	8853.539675	8853.5397	8853.5398	8858.4711	8.858E+03	8883.77467	8919.936362
	Worst	8853.533874	8853.539675	NA	NA	8927.5917	NA	NA	8938.571060
	SD	0.00E+00	0.00E+00	0.00E+00	1.15E-05	1.90E+01	1.847E+01	3.56207E+01	2.59E+01
g18	Best	-0.8660254	-0.8660254	-0.86602	-0.866025	-0.8660254	-0.8660254	-0.866025	-0.866025
	Median	-0.8660254	-0.8660254	NA	NA	-0.8660254	NA	NA	NA
	Mean	-0.8660254	-0.8660254	-0.86602	-0.866024	-0.8405529	-0.8660254	-0.866025	-0.859657
	Worst	-0.8660254	-0.8660254	NA	NA	-0.6749814	NA	NA	-0.674981
	SD	0.00E+00	0.00E+00	0.00E+00	7.0436E-07	6.60E-02	0.00E+00	0.00E+00	3.48E-02
g19	Best	32.65559377	32.65559313	32.655593	32.655593	32.65559	32.65559	32.655593	32.655593
	Median	32.65560741	32.65569552	NA	NA	32.65559	NA	NA	NA
	Mean	32.65562603	32.65561099	32.655593	32.757340	32.65559	3.475E+01	32.655593	32.660587
	Worst	32.65570220	32.65640159	NA	NA	32.65567	NA	NA	32.785360
	SD	3.73E-05	8.30E-05	6.46E-07	6.145E-02	1.6E-05	2.524E+00	2.52789E-07	2.37E-02
g20	Best	-	-	-	-	-	-	-	-
	Median	-	-	-	-	-	-	-	-
	Mean	-	-	-	-	-	-	-	-
	Worst	-	-	-	-	-	-	-	-
	SD	-	-	-	-	-	-	-	-

Table 3 continued

Problem	Features	NDE	rank-iMDDE	ISAMODE-CMA	SAMO-DE	MDEDP	AH-DEa	DE-DPS	DECV
g21	Best	193.7245101	193.7245101	193.72451	193.724510	193.72451	1.937E+02	193.72451	193.724510
	Median	193.7245101	193.7245101	NA	NA	193.72451	NA	NA	NA
	Mean	193.7245101	193.7245101	193.72451	193.771375	220.35875	1.939E+02	193.72451	198.090578
	Worst	193.7245101	193.7245101	NA	NA	330.18271	NA	NA	324.702842
	SD	6.26E-011	0.00E+00	0.00E+00	1.9643E-02	5.40E+01	6.977E-01	0.00E+00	2.39E+01
	Best	-	-	-	-	-	-	-	-
g22	Median	-	-	-	-	-	-	-	-
	Mean	-	-	-	-	-	-	-	-
	Worst	-	-	-	-	-	-	-	-
	SD	-	-	-	-	-	-	-	-
	Best	-400.0551000	-400.0551000	-400.0551	-396.165732	-400.05510	-4.001E+02	-400.0551	-400.055093
	Median	-400.0551000	-400.0549541	NA	NA	-400.05510	NA	NA	NA
g23	Mean	-400.0551000	-398.1808652	-395.62403	-360.817656	-400.05509	-3.444E+02	-400.0551	-392.029610
	Worst	-400.0551000	-378.0715141	NA	NA	-400.05505	NA	NA	-342.524522
	SD	0.00E+00	4.51E+00	7.79E+00	1.9623E+01	1.30E-05	7.782E+01	0.00E+00	4.51E+00
	Best	-5.50801327	-5.50801327	-5.508013	-5.508013	-5.5080133	-5.50801327	-5.508013	-5.5080132
	Median	-5.50801327	-5.50801327	NA	NA	-5.5080133	NA	NA	-5.5080132
	Mean	-5.50801327	-5.50801327	-5.508013	-5.508013	-5.5080133	-5.50801327	-5.508013	-5.5080132
g24	Worst	-5.50801327	-5.50801327	NA	NA	-5.5080133	NA	NA	-5.5080132
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.80E-15	0.00E+00	0.00E+00	2.71E-15

Table 4 Algorithm ranking by the objective function values for all test functions

	NDE	rank-iMDDE	ISAMODE-CMA	SAMO-DE	MDEDP	AH-DEa	DE-DPS	DECV
g01	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
g02	3.0	2.0	6.0	5.0	7.0	4.0	1.0	8.0
g03	4.0	4.0	4.0	4.0	4.0	4.0	4.0	8.0
g04	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
g05	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
g06	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
g07	3.5	3.5	3.5	7.0	3.5	3.5	3.5	8.0
g08	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
g09	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
g10	3.0	3.0	3.0	7.0	6.0	3.0	3.0	8.0
g11	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
g12	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
g13	4.0	4.0	4.0	4.0	4.0	4.0	4.0	8.0
g14	3.5	3.5	3.5	7.0	3.5	3.5	3.5	8.0
g15	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
g16	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
g17	2.0	2.0	2.0	4.0	5.5	5.5	7.0	8.0
g18	3.0	3.0	3.0	6.0	8.0	3.0	3.0	7.0
g19	5.0	4.0	2.0	7.0	2.0	8.0	2.0	6.0
g20	–	–	–	–	–	–	–	–
g21	2.5	2.5	2.5	5.0	8.0	6.0	2.5	7.0
g22	–	–	–	–	–	–	–	–
g23	1.5	4	5	7	3	8	1.5	6
g24	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
Average rank	3.84091	3.86364	4.0	5.11364	4.72727	4.63636	3.84091	5.97727

the statistical analysis results of applying Wilcoxon’s test between NDE and other seven compared algorithms. From the results shown in Table 5, we can see that NDE provides higher R^+ values than R^- in all the cases with exception to DE-DPS case. Precisely, we can draw the following conclusions: NDE outperforms SAMO-DE, AH-DEA and DECV significantly, and it is comparable with other algorithms.

Comparisons with other state-of-the-art EAs

To further verify the performance of NDE, comparisons are carried out with seven competitive state-of-the-art evolutionary algorithms. These algorithms are SAM-PSO (Elsayed et al. 2014), SAMO-GA (Elsayed et al. 2011), ECHT-EP2 (Mallipeddi and Suganthan 2010), M-ABC (Mezura-Montes and Cetina-Domínguez 2012), SOA-FS (Tuba and Bacanin 2014), HCOEA (Wang et al. 2007) and SMES (Mezura-Montes and Coello 2005). The selected algorithms had the appropriate performances available to make this comparison possible and used the same maximum number of function evaluations (FEs) (240,000). The best, the median, the mean, the worst, and the standard deviation of the objective func-

tion values of the final solutions for each algorithm are listed in Table 6. A result in boldface indicates the best result or the global optimum. NA means not available. The results provided by these approaches were directly taken from the original references for each approach. Note that SOA-FS, HCOEA and SMES algorithms solved only the first 13 benchmark problems. According to the results shown in Table 6, it can be obviously seen that almost all the eight algorithms can find the optimal solution consistently for six test functions (g01, g03, g04, g08, g11, g12). With respect to the mean results, only NDE was able to obtain the best result or global optimal in functions (g02, g19, g21, g23). Moreover, NDE had almost better performance with 0 standard deviations in most cases. All in all, NDE is superior to all compared EAs algorithms.

In order to perform the Friedman test, we rank all the algorithms according to the mean values in Table 7. From Table 7, NDE gets the first ranking among the five algorithms on 22 test functions. Figure 12 shows the average ranking of the considered DE algorithm based on the Friedman test. For $N = 22$ and $k = 5$, the Friedman statistic is

Fig. 11 Average ranking of NDE, rank-Imdde, ISAMODE-CMA, SAMO-DE, MDEDP, AH-DEa, DE-DPS and DECV by the Friedman test for the 22 functions in terms of the mean value

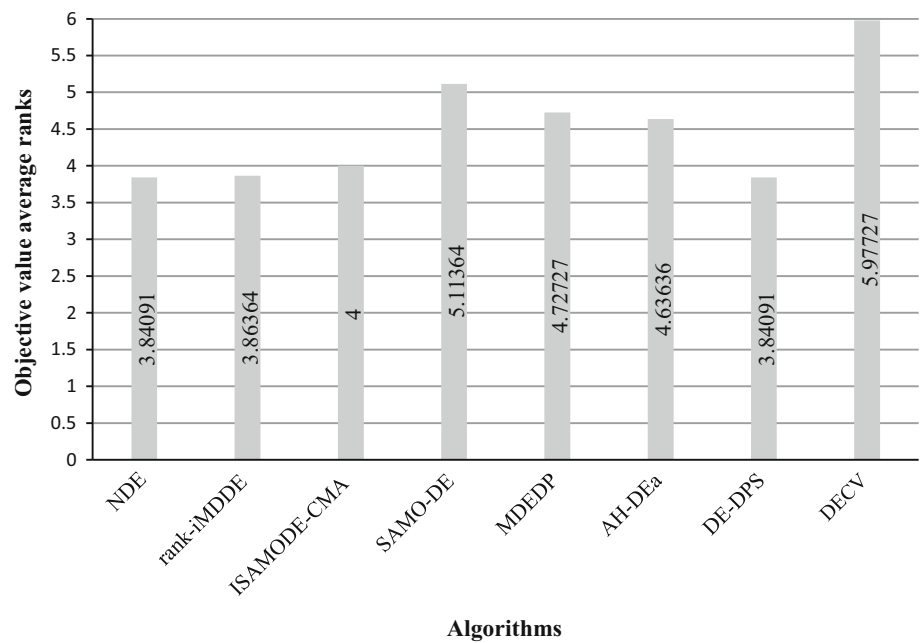


Table 5 Results of multiple-problem Wilcoxon’s test for NDE versus rank-iMDDE, ISAMODE-CMA, SAMO-DE, MDE-DP, AH-DEa, DE-DPS and DECV over all functions at a 0.05 significance level

Algorithm	R ⁺	R ⁻	p value	Better	Equal	Worse	Dec.
NDE versus rank-iMDDE	7	3	0.465	2	18	2	≈
NDE versus ISAMODE-CMA	67	38	0.363	9	8	5	≈
NDE versus SAMO-DE	173	17	0.002	16	3	3	+
NDE versus MDE-DP	95	41	0.163	9	6	7	≈
NDE versus AH-DEa	115	21	0.015	12	6	4	+
NDE versus DE-DPS	53	67	0.691	7	7	8	≈
NDE versus DECV	162	9	0.001	16	4	2	+

$$\chi^2_F = \frac{12 \times 22}{5 \times 6} \left[(2.02273^2 + 2.63636^2 + 3.61364^2 + 2.79545^2 + 3.93182^2) - \frac{5 \times 6^2}{4} \right] = 20.89097.$$

$$F_F = \frac{21 \times 20.89097}{22 \times 4 - 20.89097} = 6.537278.$$

F-distribution with (k - 1) and (k - 1)(N - 1) degrees of freedom.

The critical value of F(4, 84) for α = 0.05 is 2.4803. Therefore, the null-hypothesis that ranks do not significantly differ is rejected. Thus, it can be concluded that there is a significant difference between the performances of the algorithms. Additionally, due to the importance of the multiple-problem statistical analysis, Table 8 summarizes the statistical analysis results of applying Wilcoxon’s test between NDE and other four compared algorithms. From the results shown in Table 8, we can see that NDE provides higher R⁺ values than R⁻ in all the cases. Precisely, we can draw the following conclusions: NDE outperforms SAMO-GA,

ECHT-EP2 and M-ABC significantly, and it is comparable with SAM-PSO.

In order to perform the Friedman test, we rank all the algorithms according to the mean values in Table 9. From Table 9, NDE gets the first ranking among the four algorithms on 13 test functions. Figure 13 shows the average ranking of the considered DE algorithm based on the Friedman test. For N = 13 and k = 4, the Friedman statistic is

$$\chi^2_F = \frac{12 \times 13}{4 \times 5} \left[(1.84^2 + 2.69^2 + 2.16^2 + 3.31^2) - \frac{4 \times 5^2}{4} \right] = 9.69852.$$

$$F_F = \frac{12 \times 9.69852}{13 \times 3 - 9.69852} = 3.97189.$$

F-distribution with (k - 1) and (k - 1)(N - 1) degrees of freedom.

The critical value of F(3, 36) for α = 0.05 is 2.866. Therefore, the null-hypothesis that ranks do not significantly differ is rejected. Thus, it can be concluded that there is a significant difference between the performances of the algorithms. Additionally, due to the importance of the multiple-problem statistical analysis, Table 10 summarizes the statistical anal-

Table 6 Statistical features of the results obtained by of NDE, SAM-PSO, SAMO-GA, ECHT-EP2, M-ABC, SOA-FS, HCOEA and SMES

Problem	Features	NDE	SAM-PSO	SAMO-GA	ECHT-EP2	M-ABC	SOA-FS	HCOEA	SMES
g01	Best	-15.000000	-15.000000	-15.000000	-15.000000	-15.000000	-15.000000	-15.000	-15.000
	Median	-15.000000	NA	NA	-15.000000	NA	NA	-15.000	-15.000
	Mean	-15.000000	-15.000000	-15.000000	-15.000000	-15.000000	-15.000000	-15.000000	-15.000
	Worst	-15.000000	NA	NA	-15.000000	-15.000000	NA	-14.999998	-15.000
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
g02	Best	-0.803480	-0.8036191	-0.80359052	-0.8036191	-0.803615	-0.803618	-0.803241	-0.803601
	Median	-0.801962	NA	NA	-0.8033239	NA	NA	-0.802556	-0.792549
	Mean	-0.801809	-0.79606	-0.79604769	-0.7998220	-0.799336	-0.796049	-0.801258	-0.785238
	Worst	-0.800495	NA	NA	-0.7851820	-0.777438	NA	-0.792363	-0.751322
	SD	5.10E-04	5.3420E-03	5.8025E-03	6.29E-03	6.84E-03	9.00E-03	3.832E-03	1.67E-02
g03	Best	-1.0005001	-1.0005001	-1.0005	-1.0005	-1.000	-1.0000	-1.00000000	-1.000
	Median	-1.0005001	NA	NA	-1.0005	NA	NA	-1.00000000	-1.000
	Mean	-1.0005001	-1.0005001	-1.0005	-1.0005	-1.000	-1.0000	-1.00000000	-1.000
	Worst	-1.0005001	NA	NA	-1.0005	-1.000	NA	-1.00000000	-1.000
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.68E-05	0.00E+00	0.00E+00	2.09E-04
g04	Best	-30665.539	-30665.539	-30665.5386	-30665.53867	-30665.539	-30665.539	-30665.539	-30665.539
	Median	-30665.539	NA	NA	-30665.53867	NA	NA	-30665.539	-30665.539
	Mean	-30665.539	-30665.539	-30665.5386	-30665.53867	-30665.539	-30665.539	-30665.539	-30665.539
	Worst	-30665.539	NA	NA	-30665.53867	-30665.539	NA	-30665.539	-30665.539
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.22E-11	0.00E+00	0.00E+00	0.00E+00
g05	Best	5126.49671	5126.49671	5126.497	5126.49671	5126.736	5126.497	5126.4981	5126.599
	Median	5126.49671	NA	NA	5126.49671	NA	NA	5126.4981	5160.198
	Mean	5126.49671	5126.49671	5127.976432	5126.49671	5178.139	5134.715	5126.4981	5147.492
	Worst	5126.49671	NA	NA	5126.49671	5317.196	NA	5126.4981	5304.167
	SD	0.00E+00	1.3169E-10	1.1166E+00	0.00E+00	5.60E+01	6.796E+00	1.727E-07	50.06E+0
g06	Best	-6961.81388	-6961.81388	-6961.81388	-6961.81388	-6961.814	-6961.813875	-6961.8138	-6961.814
	Median	-6961.81388	NA	NA	-6961.81388	NA	NA	-6961.8138	-6961.814
	Mean	-6961.81388	-6961.81388	-6961.81388	-6961.81388	-6961.814	-6961.813875	-6961.8138	-6961.284
	Worst	-6961.81388	NA	NA	-6961.81388	-6961.814	NA	-6961.8138	-6952.482
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.507E-12	1.85E+0
g07	Best	24.306209	24.306209	24.3062	24.3062	24.315	24.314	24.3064582	24.327
	Median	24.306209	NA	NA	24.3063	NA	NA	24.3073055	24.426
	Mean	24.306209	24.306209	24.4113	24.3063	24.415	24.331	24.3073989	24.475
	Worst	24.306209	NA	NA	24.3063	24.854	NA	24.3092401	24.843
	SD	1.35E-014	1.9289E-08	4.5905E-02	3.19E-05	1.24E-01	1.50E-02	7.118E-04	1.32E-01

Table 6 continued

Problem	Features	NDE	SAM-PSO	SAMO-GA	ECHT-EP2	M-ABC	SOA -FS	HCOEA	SMES
g08	Best	-0.095825	-0.095825	-0.09582504	-0.09582504	-0.095825	-0.09582504	-0.095825	-0.095825
	Median	-0.095825	NA	NA	-0.09582504	NA	NA	-0.095825	-0.095825
	Mean	-0.095825	-0.095825	-0.09582504	-0.09582504	-0.095825	-0.09582504	-0.095825	-0.095825
	Worst	-0.095825	NA	NA	-0.09582504	-0.095825	NA	-0.095825	-0.095825
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.23E-17	0.00E+00	0.00E+00	0.00E+00
g09	Best	680.630057	680.630057	680.630	680.630057	680.632	680.630	680.6300574	680.632
	Median	680.630057	NA	NA	680.630057	NA	NA	680.6300574	680.642
	Mean	680.630057	680.630057	680.634	680.630057	680.647	680.634	680.6300574	680.643
	Worst	680.630057	NA	NA	680.630057	680.691	NA	680.6300578	680.719
	SD	0.00E+00	0.00E+00	1.4573E-03	2.61E-08	1.55E-2	2.00E-03	9.411E-8	1.55E-02
g10	Best	7049.248020	7049.248	7049.2481	7049.2483	7051.706	7049.329	7049.286598	7051.903
	Median	7049.248020	NA	NA	7049.2488	NA	NA	7049.486145	7253.603
	Mean	7049.248020	7049.248	7144.40311	7049.2490	7233.882	7139.531	7049.525438	7253.047
	Worst	7049.248020	NA	NA	7049.2501	7473.109	NA	7049.984208	7638.366
	SD	3.41E-09	1.5064E-05	6.7860E+01	6.60E-04	1.10E+02	6.696E+01	1.502E-01	136.02E+0
g11	Best	0.749999	0.749999	0.749999	0.749999	0.749999	0.749999	0.75	0.75
	Median	0.749999	NA	NA	0.749999	NA	NA	0.75	0.75
	Mean	0.749999	0.749999	0.749999	0.749999	0.749999	0.749999	0.75	0.75
	Worst	0.749999	NA	NA	0.749999	0.749999	NA	0.75	0.75
	SD	0.00E+00	0.00E+00	0.00E+00	3.40E-16	2.30E-05	0.00E+00	1.546E-12	1.52E-04
g12	Best	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
	Median	-1.000000	NA	NA	-1.000000	NA	NA	-1.000000	-1.000000
	Mean	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000	-1.000000
	Worst	-1.000000	NA	NA	-1.000000	-1.000000	NA	-1.000000	-1.000000
	SD	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
g13	Best	0.0539415	0.0539415	0.053942	0.0539415	0.053985	0.053942	0.0539498	0.053986
	Median	0.0539415	NA	NA	0.0539415	NA	NA	0.0539498	0.061873
	Mean	0.0539415	0.0539415	0.054028	0.0539415	0.158552	0.138	0.0539498	0.166385
	Worst	0.0539415	NA	NA	0.0539415	0.442905	NA	0.0539499	0.468294
	SD	0.00E+00	0.00E+00	5.9414E-05	1.30E-12	1.72E-01	9.60E-02	8.678E-08	1.77E-01
g14	Best	-47.7648885	-47.76489	-47.1883	-47.7649	-47.641	-	-	-
	Median	-47.7648885	NA	NA	-47.7648	NA	-	-	-
	Mean	-47.7648885	-47.76489	-46.47317932	-47.7648	-47.271	-	-	-
	Worst	-47.7648885	NA	NA	-47.7648	-46.537	-	-	-
	SD	0.00E+00	2.8342E-07	3.1590E-01	2.72E-05	2.46E-01	-	-	-

Table 6 continued

Problem	Features	NDE	SAM-PSO	SAMO-GA	ECHEP2	M-ABC	SOA-FS	HCOEA	SMES
g15	Best	961.7150223	961.7150223	961.71502	961.715022	961.715	-	-	-
	Median	961.7150223	NA	NA	961.715022	NA	-	-	-
	Mean	961.7150223	961.7150223	961.715087	961.715022	961.719	-	-	-
	Worst	961.7150223	NA	NA	961.715022	961.793	-	-	-
	SD	0.00E+00	0.00E+00	5.5236E-05	2.01E-13	1.42E-02	-	-	-
g16	Best	-1.90515525	-1.90515525	-1.905155	-1.9051553	-1.905	-	-	-
	Median	-1.90515525	NA	NA	-1.9051553	NA	-	-	-
	Mean	-1.90515525	-1.90515525	-1.905154	-1.9051553	-1.905	-	-	-
	Worst	-1.90515525	NA	NA	-1.9051553	-1.905	-	-	-
	SD	0.00E+00	0.00E+00	6.9520E-07	1.12E-10	4.52E-16	-	-	-
g17	Best	8853.533874	8853.539675	8853.5397	8853.5397	8866.618	-	-	-
	Median	8853.533874	NA	NA	8853.5397	NA	-	-	-
	Mean	8853.533874	8853.539675	8853.8871	8853.5397	8987.459	-	-	-
	Worst	8853.533874	NA	NA	8853.5397	9165.219	-	-	-
	SD	0.00E+00	2.1125E-07	1.7399E-01	2.13E-08	9.57E+01	-	-	-
g18	Best	-0.8660254	-0.8660254	-0.866025	-0.8660254	-0.866006	-	-	-
	Median	-0.8660254	NA	NA	-0.8660254	NA	-	-	-
	Mean	-0.8660254	-0.8660254	-0.865545	-0.8660254	-0.7950187	-	-	-
	Worst	-0.8660254	NA	NA	-0.8660254	-0.672216	-	-	-
	SD	0.00E+00	7.5801E-08	4.0800E-04	1.00E-09	9.39E-02	-	-	-
g19	Best	32.65559377	32.656235	32.655592	32.6591	33.285	-	-	-
	Median	32.65560741	NA	NA	32.6611	NA	-	-	-
	Mean	32.65562603	32.666236	36.427463	32.6623	34.267	-	-	-
	Worst	32.65570220	NA	NA	32.6687	35.746	-	-	-
	SD	3.73E-05	1.5762E-02	1.0372E+00	3.4E-03	6.31E-01	-	-	-
g20	Best	-	-	-	-	-	-	-	-
	Median	-	-	-	-	-	-	-	-
	Mean	-	-	-	-	-	-	-	-
	Worst	-	-	-	-	-	-	-	-
	SD	-	-	-	-	-	-	-	-
g21	Best	193.7245101	193.72451	193.724510	193.7246	266.500	-	-	-
	Median	193.7245101	NA	NA	193.7399	NA	-	-	-
	Mean	193.7245101	193.72474	246.091539	193.7438	306.609	-	-	-
	Worst	193.7245101	NA	NA	193.7741	329.960	-	-	-
	SD	6.26E-011	8.9521E-05	1.4917E+01	1.65E-02	1.98E+01	-	-	-

Table 6 continued

Problem	Features	NDE	SAM-PSO	SAMO-GA	ECHT-EP2	M-ABC	SOA-FS	HCOEA	SMES
g22	Best	-	-	-	-	-	-	-	-
	Median	-	-	-	-	-	-	-	-
	Mean	-	-	-	-	-	-	-	-
	Worst	-	-	-	-	-	-	-	-
	SD	-	-	-	-	-	-	-	-
g23	Best	-400.0551000	-400.0551	-355.661000	-398.9731	-159.739	-	-	-
	Median	-400.0551000	NA	NA	-385.5657	NA	-	-	-
	Mean	-400.0551000	-400.0512	-194.760335	-373.2178	-35.272	-	-	-
	Worst	-400.0551000	NA	NA	-335.1145	109.010	-	-	-
	SD	0.00E+00	1.2014E-03	5.3278E+01	3.37E+01	8.28E+01	-	-	-
g24	Best	-5.50801327	-5.508013	-5.508013	-5.5080	-5.508	-	-	-
	Median	-5.50801327	NA	NA	-5.5080	NA	-	-	-
	Mean	-5.50801327	-5.508013	-5.508013	-5.5080	-5.508	-	-	-
	Worst	-5.50801327	NA	NA	-5.5080	-5.508	-	-	-
	SD	0.00E+00	0.00E+00	0.00E+00	1.80E-15	2.71E-15	-	-	-

Table 7 Algorithms ranking by the objective function values for all test functions

	NDE	SAM-PSO	SAMO-GA	ECHT-EP2	M-ABC
g01	3.0	3.0	3.0	3.0	3.0
g02	1.0	4.0	5.0	2.0	3.0
g03	3.0	3.0	3.0	3.0	3.0
g04	3.0	3.0	3.0	3.0	3.0
g05	2.0	2.0	4.0	2.0	5.0
g06	3.0	3.0	3.0	3.0	3.0
g07	2.0	2.0	4.0	2.0	5.0
g08	3.0	3.0	3.0	3.0	3.0
g09	1.5	3.5	1.5	5.0	3.5
g10	1.0	2.0	4.0	3.0	5.0
g11	3.0	3.0	3.0	3.0	3.0
g12	3.0	3.0	3.0	3.0	3.0
g13	2.0	2.0	4.0	2.0	5.0
g14	1.0	2.0	5.0	3.0	4.0
g15	1.0	2.0	4.0	3.0	5.0
g16	3.0	3.0	3.0	3.0	3.0
g17	1.0	2.5	4.0	2.5	5.0
g18	2.0	2.0	4.0	2.0	5.0
g19	1.0	3.0	5.0	2.0	4.0
g20	–	–	–	–	–
g21	1.0	2.0	4.0	3.0	5.0
g22	–	–	–	–	–
g23	1.0	2.0	4.0	3.0	5.0
g24	3.0	3.0	3.0	3.0	3.0
Average rank	2.02273	2.63636	3.61364	2.79545	3.93182

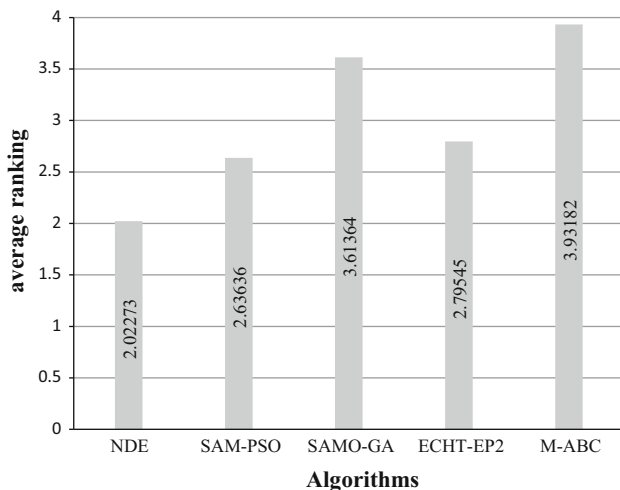


Fig. 12 Average ranking of NDE, SAM-PSO, SAMO-GA, ECHT-EP2 and M-ABC by the Friedman test for the 22 functions in terms of the mean value

ysis results of applying Wilcoxon’s test between NDE and other four compared algorithms. From the results shown in Table 10, we can see that NDE provides higher R^+ values

than R^- in all the cases. Precisely, we can draw the following conclusions: NDE outperforms SOA-FS, HCOEA and SMES.

In the final analysis, the performance of the NDE algorithm is superior and competitive to the state-of-the-art well-known DE approaches and EAs in terms of final solution quality and robustness. Moreover, it is easily implemented and is considered a reliable approach to constrained optimization problem. In all the rankings the NDE is a powerful algorithm showing fast convergence, providing good final results and significantly outperforming most of the compared algorithms.

NDE for constrained engineering design problems

In order to study the performance of the proposed algorithm NDE on real-world constrained optimization problems, five widely used constrained engineering design problems have been solved. The five problems are: (i) welded beam design (Huang et al. 2007), (ii) tension/compression spring design (Ray and Liew 2003), (iii) speed reducer design (Ray and Liew 2003), (iv) three-bar truss design (Ray and Liew 2003), and (v) pressure vessel design (Huang et al. 2007). For

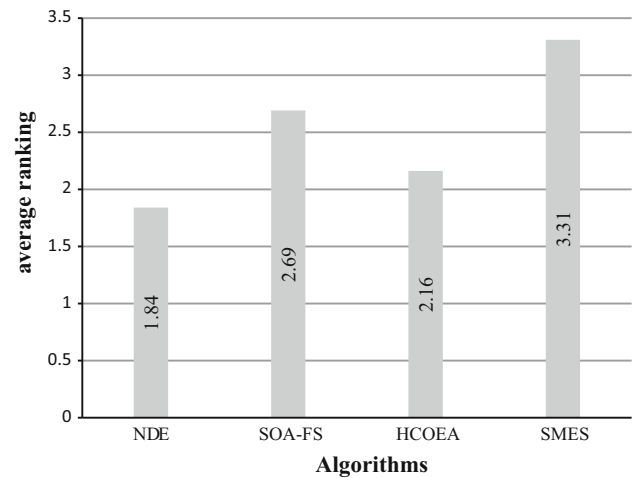
Table 8 Results of multiple-problem Wilcoxon's test for NDE versus SAM-PSO, SAMO-GA, ECHT-EP2 and M-ABC over all functions at a 0.05 significance level

Algorithm	R ⁺	R ⁻	<i>p</i> value	Better	Equal	Worse	Dec.
NDE versus SAM-PSO	31	5	0.069	6	14	2	≈
NDE versus SAMO-GA	152	1	0.000	16	5	1	+
NDE versus ECHT-EP2	85	6	0.006	10	9	3	+
NDE versus M-ABC	134	2	0.001	15	6	1	+

Table 9 Algorithm ranking by the objective function values for first 13 test functions

	NDE	SOA-FS	HCOEA	SMES
g01	2.5	2.5	2.5	2.5
g02	1.0	3.0	2.0	4.0
g03	2.5	2.5	2.5	2.5
g04	2.5	2.5	2.5	2.5
g05	1.0	3.0	2.0	4.0
g06	2.0	2.0	2.0	4.0
g07	1.0	3.0	2.0	4.0
g08	2.5	2.5	2.5	2.5
g09	1.5	3.0	1.5	4.0
g10	1.0	3.0	2.0	4.0
g11	2.5	2.5	2.5	2.5
g12	2.5	2.5	2.5	2.5
g13	1.5	3.0	1.5	4.0
Average rank	1.84	2.69	2.16	3.31

reasons of (limited) space, interested readers can find the formulation of these problems in their corresponding literature. For each problem, 30 independent runs are performed and statistical results are provided including the best, median, mean and worst results and the standard deviation. Moreover, the statistical results provided by NDE were compared with some algorithms used in literature. The same settings of parameters are used. However, due to the different characteristics of different problems, NDE utilizes different population size, maximum number of generation (G_{max}), and maximum number of function evaluations (FEs) for each test function and are listed in Table 11. The best values of objective function, design variables and constraints obtained by NDE in each problem are presented in Table 12. The statistical results of all engineering design problems that measure the quality of results (best, median, mean, worst, and standard deviation) are summarized in Table 13. From Table 13, it can be concluded that NDE is able to consistently find the global optimal in all engineering problems with a very small standard deviation and with a very small (FEs). This indicates that the proposed NDE has a remarkable ability to solve constrained engineering design problems with a perfect performance in terms of high quality solution, rapid convergence speed, efficiency and robustness.

**Fig. 13** Average ranking of NDE, SOA-FS, HCOEA and SMES by the Friedman test for the 13 functions in terms of the mean value

Welded beam design problem

This problem was previously solved by several approaches in the literature. NDE is compared with twelve optimizers: (COMDE) (Mohamed and Sabry 2012), differential evolution with level comparison (DELIC) (Wang and Li 2010), multiple trial vector based DE (MDDE) (Mezura-Montes et al. 2007), particle swarm optimization (CVI-PSO) (Mazhoud et al. 2013), genetic algorithm with automatic dynamic penalization (BIANCA) (Montemurro et al. 2013), multi-view differential evolution (MVDE) (de Melo and Carosio 2013), (Rank-iMDDE) (Gong et al. 2014), hybrid particle swarm optimization (HPSO) (He and Wang 2007), co-evolutionary DE (Co-DE) (Huang et al. 2007), mine blast algorithm (MBA) (Sadollah et al. 2013), water cycle algorithm (WCA) (Eskandar et al. 2012), and social-spider algorithm (SSO-C) (Cuevas and Cienfuegos 2014). The results of these algorithms are shown in Table 14. A result in boldface means a better (or best) solution obtained. It can be obviously seen from Table 14 that NDE is able to find the optimal solution consistently and its statistical result is similar to COMDE, DELIC, MDDE, Rank-iMDDE and MBA algorithms and is superior to the remaining algorithms. However, It is to be noted that the improvement percentage of NDE in terms of FEs in comparison to COMDE, DELIC, MDDE, Rank-iMDDE and MBA is 40,

Table 10 Results of multiple-problem Wilcoxon’s test for NDE versus SOA-FS, HCOEA and SMES over 13 functions at a 0.05 significance level

Algorithm	R ⁺	R ⁻	p value	Better	Equal	Worse	Dec.
NDE versus SOA-FS	35	1	0.017	7	5	1	+
NDE versus HCOEA	28	0	0.018	7	6	0	+
NDE versus SMES	28	0	0.018	7	6	0	+

Table 11 The required population size, maximum number of generation and number of function evaluations for engineering design problems

Problem	n	NP	G _{max}	FES
Welded beam	4	40	200	8000
Spring design	3	60	400	24,000
Speed reducer	7	40	450	18,000
Three-bar truss	2	40	100	4000
Pressure vessel	4	125	160	20,000

40, 33.33, 46.67 and 82.77%, respectively. Therefore, NDE is considered the most efficient with the smallest (FES) among all compared algorithms. The convergence graph of NDE for the welded beam design problem is plotted in Fig. 14.

Tension/compression spring design problem

The approaches applied to this problem for the comparison include (COMDE), (DELIC), (MDDE), (CVI-PSO), (BIANCA), (MVDE), (Rank-iMDDE), (HPSO), (Co-DE), (MBA), (WCA), (SSO-C), dynamic stochastic ranking based DE (DSS-MDE) (Zhang et al. 2008), and accelerated adaptive trade-off model based EA (AATM) (Wang et al. 2009). The results are tabulated in Table 15. It can be observed that there are seven algorithms (NDE, COMDE, DELIC, DSS-MDE, Rank-iMDDE, HPSO, and SSO-C) that are able to find the optimal solution in this problem but DELIC has the lower standard deviation. The mean value obtained by NDE is slightly worse than the mean values obtained by DELIC and Rank-iMDDE. However, NDE performs better than the remaining algorithms in terms of the mean value. The convergence graph of NDE for the tension/compression spring design problem is plotted in Fig. 15.

Table 12 Best values of objective functions, design variables and constraints for engineering design problems

	Welded beam	Spring design	Speed reducer	Three-bar truss	Pressure vessel
x ₁	0.2057296389468447	0.0516890589523321	3.5000000000000000	0.78867531961558757	0.8125000000000000
x ₂	3.4704887037902257	0.3567176885468677	0.7000000000000000	0.40824776714655764	0.4375000000000000
x ₃	9.0366239107577062	11.288968756392725	17.0000000000000000	–	42.098445595800
x ₄	0.2057296397864752	–	7.3000000000000000	–	176.63659584330
x ₅	–	–	7.7153199114882929	–	–
x ₆	–	–	3.3502146660972438	–	–
x ₇	–	–	5.2866544649806402	–	–
x ₈	–	–	–	–	–
x ₉	–	–	–	–	–
x ₁₀	–	–	–	–	–
f(x)	1.7248523125432824	0.0126652327883194	2994.4710661479908	263.8958433765157	6059.714335
g ₁	–0.000063341994974	0	–0.073915280398147	–0.070525402833398	–0.000000000000014
g ₂	–0.000002714066795	0	–0.197998527142186	–1.467936135628140	–0.000358808290162
g ₃	–0.00000000839631	–4.053785529111818	–0.499172248101439	–0.602589267205258	–0.00000016703270
g ₄	–3.432983781912939	–0.727728835000534	–0.904643904555729	–	–0.633634041563814
g ₅	–0.080729638946845	–	–0.000000000000702	–	–
g ₆	–0.235540322586703	–	–0.000000000000235	–	–
g ₇	–0.000000209275640	–	–0.702499999999991	–	–
g ₈	–	–	–0.000000000000209	–	–
g ₉	–	–	–0.795833333333279	–	–
g ₁₀	–	–	–0.051325753542591	–	–
g ₁₁	–	–	–0.00000000001243	–	–

Table 13 Statistical results obtained by NDE with 30 independent runs on five engineering design problems

Problem	Optimal	Best	Median	Mean	Worst	SD
Welded beam	1.724852309	1.724852309	1.724852309	1.724852309	1.724852309	3.73E–12
Spring design	0.012665233	0.012665232	0.012665423	0.012668899	0.012687092	5.38E–06
Speed reducer	2994.4710661	2994.4710661	2994.4710661	2994.4710661	2994.4710661	4.17E–12
Three-bar truss	263.89584338	263.8958434	263.8958434	263.8958434	263.8958434	0.00E+00
Pressure vessel	6059.714355	6059.714335	6059.714335	6059.714335	6059.714335	4.56E–07

Table 14 Results of Welded beam design problem

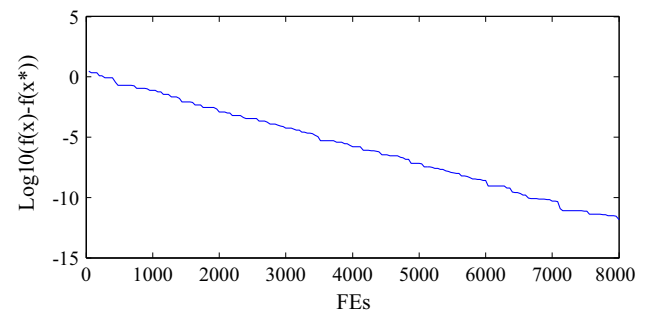
Algorithm	Best	Median	Mean	Worst	SD	FEs
NDE	1.724852309	1.724852309	1.724852309	1.724852309	3.73E–12	8000
COMDE	1.724852	1.724852	1.724852	1.724852	1.60E–12	20,000
DELDC	1.724852	1.724852	1.724852	1.724852	4.10E–13	20,000
MDDE	1.725	NA	1.725	1.725	1.00E–15	24,000
CVI-PSO	1.724852	NA	1.725124	1.727665	6.12E–04	25,000
BIANCA	1.724852	NA	1.752201	1.793233	2.30E–02	80,000
MVDE	1.7248527	NA	1.7248621	1.7249215	7.88E–06	15,000
Rank-iMDDE	1.724852309	NA	1.724852309	1.724852309	7.71E–11	15,000
HPSO	1.724852	NA	1.749040	1.814295	4.0E–02	81,000
Co-DE	1.733461	NA	1.768158	1.824105	2.22E–02	240,000
MBA	1.724853	NA	1.724853	1.724853	6.94E–19	47,340
WCA	1.724856	NA	1.726427	1.744697	4.29E–03	46,450
SSO-C	1.7248523085	NA	1.746461619	1.799331766	2.57E–02	25,000

Speed reducer design problem

This problem has been solved by (COMDE), (DELDC), (MDDE), (MVDE), (rank-iMDDE), (MBA), (WCA), (DSS-MDE), and (AATM), and the results are shown in Table 16. It is clear that NDE, COMDE, DELDC, DSS-MDE and Rank-iMDDE are able to find the optimal solution consistently in all runs but NDE has the smallest FEs among them. Note that the best result provided by MBA is slightly worse than the optimal solution although it used the least FEs. Thus, NDE is considered the most efficient with the smallest (FEs) among superior compared algorithms. The convergence graph of NDE for the speed reducer design problem is plotted in Fig. 16.

Three-bar truss design problem

This problem has been solved by (COMDE), (DELDC), (MVDE), (Rank-iMDDE), (MBA), (WCA), (DSS-MDE), and (AATM), and the results are shown in Table 17. It is clear that NDE, COMDE, DELDC, and Rank-iMDDE are able to find the optimal solution consistently in all runs. Besides, the standard deviation of NDE and rank-iMDDE are equal to zero. Moreover, NDE is considered the most efficient with the smallest (FEs) among all compared algorithms. The conver-

**Fig. 14** Convergence graph of NDE for welded beam design problem

gence graph of NDE for the three-bar truss design problem is plotted in Fig. 17.

Pressure vessel design problem

This problem has been studied by several approaches in the literature, including (COMDE), (DELDC), (MDDE), (CVI-PSO), (BIANCA), (MVDE), (rank-iMDDE), (HPSO), (Co-DE), (MBA), and (WCA). The statistical results of all algorithms are listed in Table 18. From the results only five algorithms (NDE, COMDE, DELDC, MDDE, and rank-

Table 15 Results of tension/compression spring design problem

Algorithm	Best	Median	Mean	Worst	SD	FES
NDE	0.012665232	0.012665423	0.012668899	0.012687092	5.38E−06	24,000
COMDE	0.012665233	0.012665423	0.012667168	0.012676809	3.09E−06	24,000
DELDC	0.012665233	0.012665233	0.012665267	0.012665575	1.30E−07	20,000
DSS-MDE	0.012665233	0.012665304	0.012669366	0.012738262	1.25E−05	24,000
MDDE	0.012665	NA	0.012666	0.012674	2.00E−06	24,000
AATM	0.012668262	NA	0.012708075	0.012861375	4.50E−05	25,000
CVI-PSO	0.0126655	NA	0.012731	0.0128426	5.58E−05	25,000
BIANCA	0.012671	NA	0.012681	0.012913	5.12E−05	80,000
MVDE	0.012665272	NA	0.012667324	0.012719055	2.45E−06	10,000
Rank-iMDDE	0.012665233	NA	0.012665264	0.01266765	2.45E−07	19,565
HPSO	0.0126652	NA	0.0127072	0.0127191	1.6E−05	81,000
Co-DE	0.0126702	NA	0.012703	0.012790	2.7E−05	240,000
MBA	0.012665	NA	0.012713	0.012900	6.30E−05	7650
WCA	0.012665	NA	0.012746	0.012952	8.06E−05	11,750
SSO-C	0.012665232	NA	0.012764888	0.012867916	9.28E−06	25,000

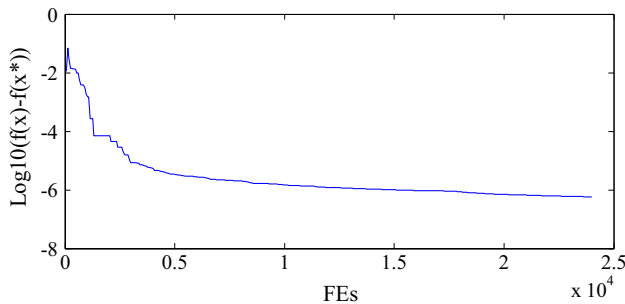


Fig. 15 Convergence graph of NDE for tension/compression spring design problem

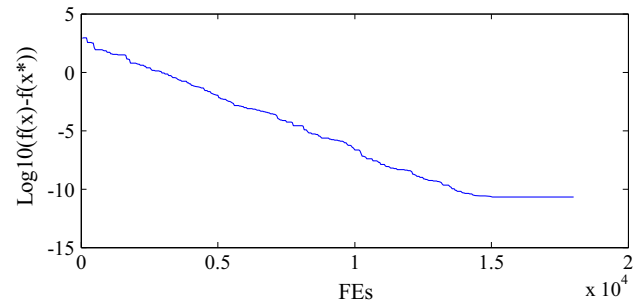


Fig. 16 Convergence graph of NDE for speed reducer design problem

iMDDE) are able to solve this problem consistently in all runs but MDDE has the lower standard deviation. Moreover, NDE is considered the most efficient with the smallest (FES)

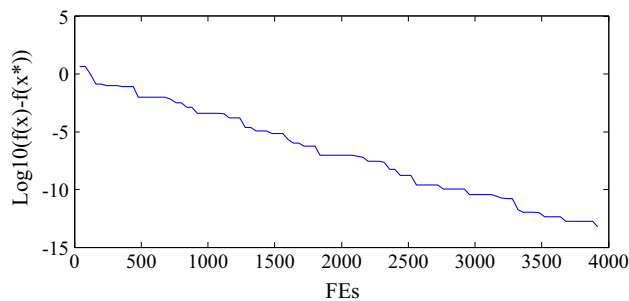
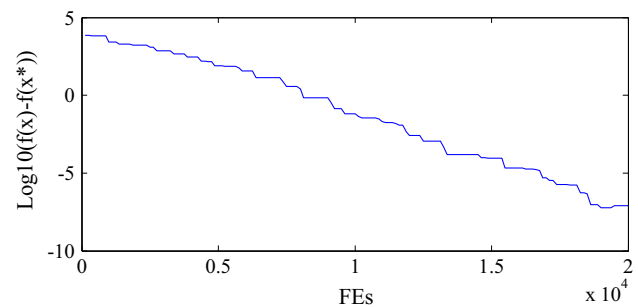
among all compared algorithms. The convergence graph of NDE for the pressure vessel design problem is plotted in Fig. 18.

Table 16 Results of speed reducer design problem

Algorithm	Best	Median	Mean	Worst	SD	FES
NDE	2994.4710661	2994.4710661	2994.4710661	2994.4710661	4.17E−12	18,000
COMDE	2994.4710661	2994.4710661	2994.4710661	2994.4710661	1.54E−12	21,000
DELDC	2994.471066	2994.471066	2994.471066	2994.471066	1.90E−12	30,000
DSS-MDE	2994.471066	2994.471066	2994.471066	2994.471066	3.60E−12	30,000
MDDE	2996.357	NA	2996.367	2996.39	8.20E−03	24,000
AATM	2994.516778	NA	2994.585417	2994.659797	3.30E−02	40,000
MVDE	2994.471066	NA	2994.471066	2994.471069	2.82E−07	30,000
rank-iMDDE	2994.471066	NA	2994.471066	2994.471066	7.93E−13	19,920
MBA	2994.482453	NA	2996.769019	2999.652444	1.56E+00	6300
WCA	2994.471066	NA	2994.474392	2994.505578	7.40E−03	15,150

Table 17 Results of three-bar truss design problem

Algorithm	Best	Median	Mean	Worst	SD	TNFE
NDE	263.8958434	263.8958434	263.8958434	263.8958434	0.00E+00	4000
COMDE	263.8958434	263.8958434	263.8958434	263.8958434	5.34E−13	7000
DELC	263.8958434	263.8958434	263.8958434	263.8958434	4.34E−14	10,000
DSS-MDE	263.8958433	263.8958433	263.8958436	263.8958498	9.7E−07	15,000
AATM	263.8958435	NA	263.90041	263.8966	1.10E−03	17,000
MVDE	263.8958434	NA	263.8958434	263.8958548	2.58E−07	7000
rank-iMDDE	263.8958434	NA	263.8958434	263.8958434	0.00E+00	4920
MBA	263.895852	NA	263.897996	263.915983	3.93E−03	13,280
WCA	263.895843	NA	263.895903	263.896201	8.71E−05	5250

**Fig. 17** Convergence graph of NDE for three-bar truss design problem**Fig. 18** Convergence graph of NDE for pressure vessel design problem

Based on the above analysis, results and comparisons, the proposed NDE algorithm is of better searching quality, efficiency and robustness for solving constrained engineering design problems and besides the fact that its performance is superior and competitive with all compared algorithms.

NDE for constrained mechanical design problems

To further verify the performance of NDE on solving real-world constrained optimization problems, five widely used

constrained mechanical engineering design problems have been solved. The five problems are: (i) step-cone pulley (Rao et al. 2011), (ii) hydrostatic thrust bearing (Rao et al. 2011), (iii) Rolling element bearing (Rao et al. 2011), (iv) Belleville spring (Rao et al. 2011), and (v) gear train (Sadollah et al. 2013). All these problems have different natures of objective functions, constraints and design variables (Rao et al. 2011). The objective functions of all problems are minimized herein. For reasons of space, interested readers can find the

Table 18 Results of pressure vessel design problem

Algorithm	Best	Median	Mean	Worst	SD	TNFE
NDE	6059.714335	6059.714335	6059.714335	6059.714335	4.56E−07	20,000
COMDE	6059.714335	6059.714335	6059.714335	6059.714335	3.62E−10	30,000
DELC	6059.7143	6059.7143	6059.7143	6059.7143	2.1E−11	30,000
MDDE	6059.702	NA	6059.702	6059.702	1.00E−12	24,000
CVI-PSO	6059.7143	NA	6292.1231	6820.4101	2.88E+02	25,000
BIANCA	6059.9384	NA	6182.0022	6447.3251	1.22E+02	80,000
MVDE	6059.714387	NA	6059.997236	6090.533528	2.91E+00	15,000
rank-iMDDE	6059.714335	NA	6059.714335	6059.714335	1.95E−12	23,465
HPSO	6059.7143	NA	6099.9323	6288.6770	86.2	81,000
Co-DE	6061.0777	NA	6085.2303	6371.0455	43.0	240,000
MBA	5889.3216	NA	6200.64765	6392.5062	160.34	70,650
WCA	5885.3327	NA	6198.6172	6590.2129	213.0490	27,500

Table 19 the required population size, maximum number of generation and number of function evaluations for mechanical design problems

Problem	n	NP	G_{max}	FES
Step-cone pulley	5	50	300	15,000
Hydrostatic thrust bearing	4	80	300	24,000
Rolling element bearing	10	50	300	15,000
Belleville spring	4	50	200	10,000
Gear train	4	25	40	1000

formulation of these problems in their corresponding literature. For each problem, 30 independent runs are performed and statistical results are provided including the best, median, mean and worst results and the standard deviation. Moreover,

the statistical results provided by NDE were compared with some algorithms used in the literature. The same settings of parameters are used. However, due to different characteristics of different problems, NDE utilizes different population size, maximum number of generation (G_{max}), and maximum number of function evaluations (FES) for each test function that are listed in Table 19. The best values of objective function, design variables and constraints obtained by NDE in each problem are presented in Table 20. The statistical results of all mechanical design problems that measure the quality of results (best, median, mean, worst, and standard deviation) are summarized in Table 21. From Table 21, it can be concluded that NDE is able to consistently find the global optimal in rolling element bearing, Belleville spring and gear train problems with a very small standard deviation

Table 20 Statistical results obtained by NDE with 30 independent runs on five mechanical design problems

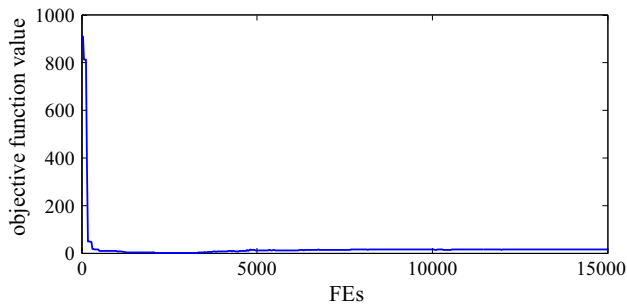
Problem	Best	Median	Mean	Worst	SD
Step-cone pulley	14.4919913809279	14.492006805051977	14.522307912133043	14.823926095157306	1.00E–01
Hydrostatic thrust bearing	1625.4427649676115	1674.4064507478774	1692.672623443781	1789.2330193627886	53.7568525
Rolling element bearing	–85549.239142260223	–85549.239092671458	–85549.23290360783	–85549.16997388922	1.81E–2
Belleville spring	1.9796747595734621	1.9796747729266873	1.9796747713912306	1.9796747801380858	9.41E–09
Gear Train	270085715E–12	9.7456E–10	1.317E–09	6.512E–09	1.753E–09

Table 21 Values of objective functions, design variables and constraints for mechanical design problems

	Step-cone pulley	Hydrostatic thrust bearing	Rolling element bearing	Belleville spring	Gear train
x_1	99.999998938907020	5.9557804954072431	125.7190556146683	12.00999999516319	43
x_2	34.587680024467569	5.3890130457499099	21.42559024077250	10.03047328200870	16
x_3	47.584054369346779	0.0000053586972684	11	0.204143354252978	19
x_4	63.441402209442454	2.2696559656861695	0.515000000000388	0.200000000186008	49
x_5	76.086394586806078	–	0.515000016599447	–	–
x_6	–	–	0.459856414789225	–	–
x_7	–	–	0.619398026392338	–	–
x_8	–	–	0.30000000001946	–	–
x_9	–	–	0.044951766218101	–	–
x_{10}	–	–	0.654902937732652	–	–
$f(x)$	14.4919913809279	1625.4427649676115	–85549.239142260223	1.979674759573462	270085715E–12
g_1	–0.0000000003235	–0.000000063708285	–0.00000000020725	–0.000029842136428	–
g_2	–0.0000000001318	–0.000000009546739	–10.661231446299219	–0.000014270942302	–
g_3	–0.0000000003731	–0.000000043195541	–0.506681365918652	–0.779703706879640	–
g_4	–0.9903991779813	–0.000324362507723	–1.778502108792939	–1.595856644178049	–
g_5	–0.9985532013866	–0.566767449657333	–0.719055614668292	–0.00000007042010	–
g_6	–1.10085573350032	–0.000996361387052	–10.518885939856887	–1.979526724875687	–
g_7	–1.10165766981873	–0.000001006240382	–0.000000000006155	–0.198965748370059	–
g_8	–706.394806606583	–	–0.000000000000388	–	–
g_9	–486.900698162025	–	–0.000000016599447	–	–
g_{10}	–216.910383706140	–	0.000000000000000	–	–
g_{11}	–0.0000002726796	–	–	–	–

Table 22 Results of step-cone pulley problem

Algorithm	Best	Median	Mean	Worst	SD	FEs
NDE	14.49199138	14.49200681	14.52230791	17.74707691	8.72E-01	15,000
Rank-iMDDE	14.487968	–	15.472300	17.833931	–	15,000
TLBO	16.634510	–	24.011358	74.022951	–	15,000
ABC	16.634655	–	36.099500	145.470500	–	15,000

**Fig. 19** Convergence graph of NDE for step-cone pulley problem

and with small (FEs). For the step-cone pulley problem and hydrostatic thrust bearing problem, although the best solution are not consistently found, the results achieved in all runs are very close to the best solution which can be verified by the small standard deviation. Clearly, it indicates that the proposed NDE algorithm is proven to be an effective and powerful approach to solve constrained mechanical design problems.

Step-cone pulley problem

This problem was previously solved using (rank-iMDDE), teaching–learning-based optimization (TLBO) (Rao et al. 2011), Artificial Bee Colony (ABC) (Rao et al. 2011). The comparison of the obtained statistical results for different algorithms is given in Table 22. From Table 22, almost NDE and rank-iMDDE obtain the same best solution. However, NDE obtains the best results compared with other optimizers in terms of mean and worst solution with the same used FEs. Thus, NDE is more robust than other compared algorithms. The convergence graph of NDE for the step-cone pulley problem is shown in Fig. 19. It can be easily seen from Fig. 19. That NDE converged to near best solutions in early generations.

Hydrostatic thrust bearing problem

This problem was previously solved using (rank-iMDDE), teaching–learning-based optimization (TLBO) (Rao et al. 2011), Artificial Bee Colony (ABC) (Rao et al. 2011). The comparison of the obtained statistical results for different

algorithms is given in Table 23. From Table 23, NDE consistently obtains the best results compared with other algorithms in terms of best, mean and worst solutions with least FEs and smallest standard deviation. The convergence graph of NDE for the hydrostatic thrust bearing problem is plotted in Fig. 20. Clearly, NDE has a fast convergence speed to achieve best solutions in early generations.

Rolling element bearing problem

This problem was previously solved using (rank-iMDDE), teaching–learning-based optimization (TLBO) (Rao et al. 2011), Artificial Bee Colony (ABC) (Rao et al. 2011), MBA (Sadollah et al. 2013), WCA (Eskandar et al. 2012) and Genetic Algorithm (GA) (Gupta and Tiwari 2007). The statistical results of all algorithms are listed in Table 24. Surprisingly, NDE is also able to find new solutions to this test functions which is better than the best known solutions from (Eskandar et al. 2012). Moreover, it almost obtains the best solution consistently in all runs with very small standard deviation. In fact, NDE detected the new best solution with significant difference over the previous solutions from (Sadollah et al. 2013; Eskandar et al. 2012) during the early generations as can be seen from the convergence graph of NDE for the Rolling element bearing problem in Fig. 21.

Belleville spring problem

This problem was previously solved using (rank-iMDDE), (TLBO), (ABC), and MBA. The statistical results of all algorithms are listed in Table 25. Almost all optimizers obtain the same best solution. NDE is slightly worse than Rank-iMDDE in terms of mean and worst results. However, NDE obtains the best solution consistently in all runs and it is considered the most efficient with the smallest FEs with fast convergence speed as depicted in the convergence graph of NDE for the Belleville spring problem in Fig. 22.

Gear train problem

This problem was previously solved using Unified Particle Swarm Optimization (UPSO) (Parsopoulos and Vrahatis 2005), (ABC), and MBA. The statistical results of all algo-

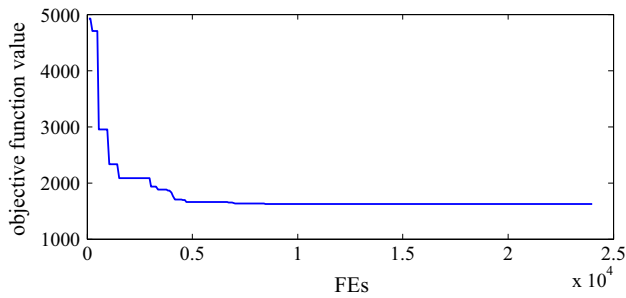


Fig. 20 Convergence graph of NDE for hydrostatic thrust bearing problem

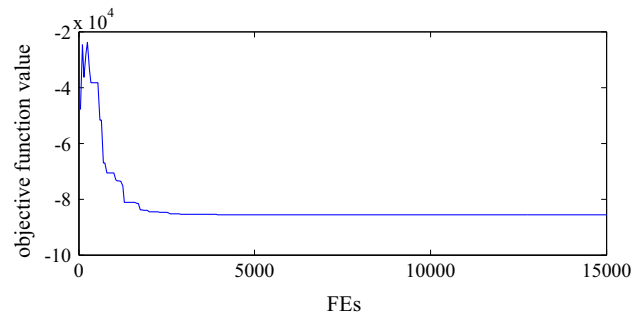


Fig. 21 Convergence graph of NDE for rolling element bearing problem

rithms are listed in Table 26. Obviously, all optimizers obtain the same best solution. NDE is slightly worse than ABC in terms of mean result. NDE is better than UPSO and MBA in terms of mean and worst results. However, ABC is considered the most efficient with the smallest FEs followed by NDE, MBA and UPSO. Nonetheless, it is noteworthy that there is a remarkable difference between the FEs used by NDE and MBA algorithms and FEs used by UPSO, the ratio is (1:100) which supports the claim that NDE is highly efficient. The convergence graph of NDE for the Gear train problem is plotted in Fig. 23.

Conclusion

Over the last decade, many EAs have been proposed to solve constrained optimization problems. However, all these algorithms including DE have the same shortcomings in solving optimization problems. These shortcoming are as follows:

- (i) premature convergence and stagnation due to the imbalance between the two main contradictory aspects exploration and exploitation capabilities during the evolution process,
- (ii) sensitivity to the choice of the control parameters which are difficult to adjust for different problems with different features,
- (iii) their performances decrease as search space dimensionality increases (Das et al. 2009; Wagdy Mohamed et al. 2011). Moreover, most real world optimization problems are limited by a set of constraints i.e. imposing constraints on the optimization problems. To optimize these constrained optimization problems, the optimal solution does not only taken into consideration the objective function value but it has also to satisfy the added constraints and this increases the difficulty to all EAs including DE of finding the global solution through the optimization process. Consequently, to overcome these obstacles, a considerable number of research studies have been proposed and developed to enhance the performance of DE, and they can be classified into three categories.

Table 23 Results of hydrostatic thrust bearing problem

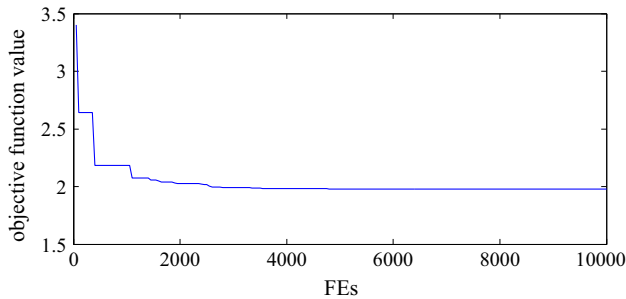
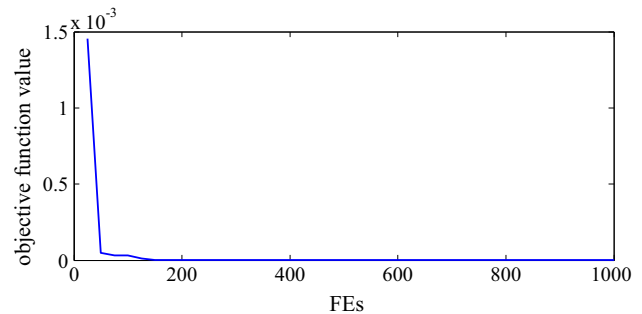
Algorithm	Best	Median	Mean	Worst	SD	FEs
NDE	1625.44276496	1674.40645074	1692.67262344	1789.233019362	53.7568525	24,000
Rank-iMDDE	1625.460142	–	1724.727935	1894.734127	–	25,000
TLBO	1625.443000	–	1797.707980	2096.801270	–	25,000
ABC	1625.442760	–	1861.554000	2144.836000	–	25,000

Table 24 Results of Rolling element bearing problem

Algorithm	Best	Median	Mean	Worst	SD	FEs
NDE	–85549.239142260	–85549.239092671	–85549.23290360	–85549.16997388	1.81E–2	15,000
Rank-iMDDE	–81859.732421	NA	–81859.010377	–81838.757577	NA	10,000
TLBO	–81859.740000	NA	–81438.987000	–80807.855100	NA	10,000
ABC	–81859.741600	NA	–81496.000000	–78897.810000	NA	10,000
MBA	–85535.9611	NA	–85321.4030	–84440.1948	211.52	15,100
WCA	–85538.48	NA	–83847.16	–83942.71	488.30	3950
GA	–81843.3	NA	NA	NA	NA	225,000

Table 25 Results of Belleville spring problem

Algorithm	Best	Median	Mean	Worst	SD	FEs
NDE	1.97967477	1.97967501	1.97967661	1.97969110	4.82E-06	10,000
Rank-iMDDE	1.979675	NA	1.979675	1.979683	NA	15,000
TLBO	1.979675	NA	1.97968745	1.979757	0.45	15,000
ABC	1.979675	NA	1.995475	2.104297	0.07	15,000
MBA	1.9796747	NA	1.984698	2.005431	7.78E-03	10,600

**Fig. 22** Convergence graph of NDE for Belleville spring problem**Fig. 23** Convergence graph of NDE for gear train problem

1. Using adaptive or self-adaptive mechanisms to adapt strategies and parameters of DE combined with single search operator or multiple search operators.
2. Controlling the population diversity of DE by introducing new parameters to measure the diversity during the evolution process.
3. Hybridizing DE with another evolutionary algorithm or classical method or incorporating local search techniques.

Although all of the above method can probably enhance the performance of Basic DE, they definitely increase the complexity of the algorithm by introducing extra parameters and/or complicated mechanisms. Nonetheless, so far, there have been a few attempts in the literature to introduce novel mutations that can balance the general trade-off between the global exploration and local exploitation with fast convergence speed (Das and Suganthan 2011). Thus, in this paper, a new triangular mutation rule is proposed. It is based on the convex combination vector of the triplet defined by the three randomly chosen vectors and the difference vectors between the best, better and the worst individuals among

the three randomly selected vectors. The main benefit of the proposed algorithm is that it neither adds extra parameters nor hybrids with another algorithm. Besides, no ensemble operators or complicated self-adaptive mechanism has been added. Thus, it can be obviously proven that the proposed triangular mutation represents a unique and significant step and a novel trend in developing the existing DE approaches. It also opens new research horizons in the field of global optimization. The performance of the proposed algorithm is evaluated on three well-known benchmark problem sets. The three sets are 24 well-known benchmark test functions presented in CEC'2006; five widely used constrained engineering design problems and five constrained mechanical design problems, respectively. The proposed NDE algorithm has been compared with 30 recent Evolutionary Algorithms. The experimental results and comparisons have shown both a remarkable performance of NDE on solving all three test functions with different characteristics and superiority when compared with other recent EAs. There are five possible directions of future work. Firstly, to study the effect of adaptation techniques on the performance of NDE, NDE will combine with self-adaptive controlling parameters. Secondly, the

Table 26 Results of gear train problem

Algorithm	Best	Median	Mean	Worst	SD	FEs
NDE	2.70085715E-12	9.7456E-10	1.317E-09	6.512E-09	1.753E-09	1000
UPSO	2.700857E-12	NA	3.80562E-08	NA	NA	100,000
ABC	2.700857E-12	NA	3.641339E-10	NA	NA	60
MBA	2.700857E-12	NA	2.471635E-09	2.062904E-08	NA	1120

performance of the NDE algorithm in solving unconstrained and constrained multi-objective benchmark problems will be investigated. Thirdly, the proposed triangle mutation operator will be joined with other compared DE based algorithms to study its impact on the optimization process. Moreover, Future research studies may focus on applying the algorithm to solve other complex real-world and engineering optimization problems. Finally, this would be greatly beneficial for theoretical analysis and the study of the stochastic behavior and properties of the proposed triangular mutation. The Matlab source code of NDE can be downloaded from <https://sites.google.com/view/optimization-project/files>.

References

- Akay, B., & Karaboga, D. (2012). Artificial bee colony algorithm for large-scale problems and engineering design optimization. *Journal of Intelligent Manufacturing*, 23, 1001–1014.
- Asafuddoula, M. D., Ray, T., & Sarker, R. A. (2014). An adaptive hybrid differential evolution algorithm for single objective optimization. *Applied Mathematics and Computation*, 231, 601–618.
- Becerra, R. L., & Coello, C. A. C. (2006). Cultured differential evolution for constrained optimization. *Computer Methods in applied Mechanics and Engineering*, 195, 4303–4322.
- Brajevic, I., & Tuba, M. (2013). An Updated artificial bee colony (ABC) algorithm for constrained optimization problems. *Journal of Intelligent Manufacturing*, 24(4), 729–740.
- Brest, J., Greiner, S., Boskovic, B., Mernik, M., & Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation*, 10, 646–657.
- Coello, C. A. C. (2002). 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*, 191, 1245–1287.
- Cuevas, E., & Cienfuegos, M. (2014). A new algorithm inspired in the behavior of the social-spider for constrained optimization. *Expert Systems with Applications*, 41, 412–425.
- Dai, C., Chen, W., & Zhu, Y. (2006). Seeker optimization algorithm. In 2006 International Conference on Computational Intelligence and Security, Guangzhou, China (pp. 225–229).
- Das, S., & Suganthan, P. N. (2011). Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1), 4–31.
- Das, S., Abraham, A., Chakraborty, U. K., & Konar, A. (2009). Differential evolution using a neighborhood-based mutation operator. *IEEE Transactions on Evolutionary Computation*, 13(3), 526–553.
- de Melo, V. V., & Carosio, G. L. (2013). Investigating multi-view differential evolution for solving constrained engineering design problems. *Expert Systems with Applications*, 40(9), 3370–3377.
- de Melo, V. V., & Carosio, G. L. C. (2012). Evaluating differential evolution with penalty function to solve constrained engineering problems. *Expert Systems with Applications*, 39(9), 7860–7863.
- Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186, 311–338.
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7, 1–30.
- Dong, N., & Wang, Y. (2014). A memetic differential evolution algorithm based on dynamic preference for constrained optimization problems. *Journal of Applied Mathematics*, Article ID 606019. doi:10.1155/2014/606019.
- Elsayed, S. M., Sarker, R. A., & Mezura-montes, E. (2014). Self-adaptive mix of particle swarm methodologies for constrained optimization. *Information Sciences*, 277, 216–233.
- Elsayed, S. M., Sarker, R. A., & Essam, D. L. (2013). An improved self-adaptive differential evolution algorithm for optimization problems. *IEEE Transactions on Industrial Informatics*, 9(1), 89–99.
- Elsayed, S. M., Sarker, R., & Essam, D. L. (2012). On an evolutionary approach for constrained optimization problem solving. *Applied Soft Computing*, 12, 3208–3227.
- Elsayed, S. M., Sarker, R. A., & Essam, D. L. (2011). Multi-operator based evolutionary algorithms for solving constrained optimization problems. *Computers and Operations Research*, 38, 1877–1896.
- Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. (2012). Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures*, 110–111, 151–166.
- Fan, H. Y., & Lampinen, J. (2003). A trigonometric mutation operation to differential evolution. *Journal of Global Optimization*, 27(1), 105–129.
- Feoktistov, V. (2006). *Differential evolution: In search of solutions*. Berlin: Springer.
- García, S., Molina, D., Lozano, M., & Herrera, F. (2009). A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behavior: A case study on the CEC'2005 special session on real parameter optimization. *Journal of Heuristics*, 15, 617–644.
- Gong, W., Cai, Z., & Wang, Y. (2013). Repairing the crossover rate in adaptive differential evolution. *Applied Soft Computing*, 15, 149–168.
- Gong, W., Cai, Z., & Liang, D. (2014). Engineering optimization by means of an improved constrained differential evolution. *Computer Methods in Applied Mechanics and Engineering*, 268, 884–904.
- Gupta, S., & Tiwari, R. (2007). Multi-objective design optimization of rolling bearing using genetic algorithm. *Mechanism and Machine Theory*, 42, 1418–1443.
- Hansen, N., Müller, S. D., & Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1), 1–18.
- He, Q., & Wang, L. (2007). A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Applied Mathematics & Computation*, 186(2), 1407–1422.
- Huang, F., Wang, L., & He, Q. (2007). An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics and Computation*, 186(1), 340–356.
- Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization, Technical, Report-TR06, pp. 1–10.
- Liang, J. J., Runarsson, T. P., Mezura-Montes, E., Clerc, M., Suganthan, P. N., Coello, C. A. C., et al. (2006). Problem definitions and evaluation criteria for the CEC 2006, special session on constrained real-parameter optimization, technical report, Nanyang Technological University, Singapore.
- Mallipeddi, R., Suganthan, P. N., Pan, Q. K., & Tasgetiren, M. F. (2011). Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied Soft Computing*, 11(2), 1679–1696.
- Mallipeddi, R., & Suganthan, P. N. (2010). Ensemble of constraint handling techniques. *IEEE Transactions on Evolutionary Computation*, 14(4), 561–579.
- Mazhoud, I., Hadj-Hamou, K., Bignon, J., & Joyeux, P. (2013). Particle swarm optimization for solving engineering problems: a new constraint-handling mechanism. *Engineering Applications of Artificial Intelligence*, 26(4), 1263–1273.

- Mezura-Montes, E., & Cetina-Domínguez, O. (2012). Empirical analysis of a modified artificial bee colony for constrained numerical optimization. *Applied Mathematics and Computation*, *218*, 10943–10973.
- Mezura-Montes, E., Miranda-Varela, M. E., & del Carmen Gómez-Ramón, R. (2010). Differential evolution in constrained numerical optimization: an empirical study. *Information Sciences*, *180*(22), 4223–4262.
- Mezura-Montes, E., Coello, C. A. C., Velázquez-Reyes, J., & Muñoz-Dávila, L. (2007). Multiple trial vectors in differential evolution for engineering design. *Engineering Optimization*, *39*(5), 567–589.
- Mezura-Montes, E., & Coello, C. A. C. (2005). A simple multi-membered evolution strategy to solve constrained optimization problems. *IEEE Transactions on Evolutionary Computation*, *9*(1), 1–17.
- Mohamed, A. W. (2015a). An improved differential evolution algorithm with triangular mutation for global numerical optimization. *Computers & Industrial Engineering*, *85*, 359–375.
- Mohamed, A. W. (2015b). An efficient modified differential evolution algorithm for solving constrained non-linear integer and mixed-integer global optimization problems. *International Journal of Machine Learning and Cybernetics*, 1–19. doi:10.1007/s13042-015-0479-6.
- Mohamed, A. W., & Sabry, H. Z. (2012). Constrained optimization based on modified differential evolution algorithm. *Information Sciences*, *194*, 171–208.
- Montemurro, M., Vincenti, A., & Vannucci, P. (2013). The automatic dynamic penalization method (ADP) for handling constraints with genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, *256*, 70–87.
- Parsopoulos, K. E., & Vrahatis, M. N. (2005). Unified particle swarm optimization for solving constrained engineering optimization problems, ADV. NAT. Computation, LNCS (Vol. 3612, pp. 582–591), Springer, Berlin.
- Price, K., Storn, R., & Lampinen, J. (2005). *Differential evolution: A practical approach to global optimization*. New York: Springer.
- Qin, A. K., Huang, V. L., & Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, *13*(2), 398–417.
- Rao, R., Savsani, V., & Vakharia, D. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, *43*(3), 303–315.
- Ray, T., & Liew, K. (2003). Society and civilization: an optimization algorithm based on the simulation of social behavior. *IEEE Transactions on Evolutionary Computation*, *7*(4), 386–396.
- Sadollah, A., Bahreinejad, A., Eskandar, H., & Hamdi, M. (2013). Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Applied Soft Computing*, *13*, 2592–2612.
- Sarker, R. M., Elsayed, S. M., & Ray, T. (2014). Differential evolution with dynamic parameters selection for optimization problems. *IEEE Transactions on Evolutionary Computation*, *18*(5), 689–707.
- Storn, R., & Price, K. (1997). Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*(4), 341–359.
- Storn, R., & Price, K. (1995). Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces, 1995; technical report TR-95-012. ICSI.
- Tessema, B., & Yen, G. (2006). A self adaptive penalty function based algorithm for constrained optimization. In *2006 IEEE congress on evolutionary computation (CEC), Piscataway* (pp. 246–253).
- Tuba, M., & Bacanin, N. (2014). Improved seeker optimization algorithm hybridized with firefly algorithm for constrained optimization problems. *Neurocomputing*, *143*, 197–207.
- Venkatraman, S., & Yen, G. G. (2005). A generic framework for constrained optimization using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, *9*(4), 424–435.
- Wagdy Mohamed, A., Sabry, H.Z., & Farhat, A. (2011). Advanced differential evolution algorithm for global numerical optimization. *IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE)*, pp.156–161.
- Wang, Y., & Cai, Z. (2012). Combining multiobjective optimization with differential evolution to solve constrained optimization problems. *IEEE Transactions on Evolutionary Computation*, *16*(1), 117–134.
- Wang, Y., Cai, Z., & Zhang, Q. (2011). Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Transactions on Evolutionary Computation*, *15*(1), 55–66.
- Wang, L., & Li, L. (2010). An effective differential evolution with level comparison for constrained engineering design. *Structural and Multidisciplinary Optimization*, *41*, 947–963.
- Wang, Y., Cai, Z., & Zhou, Y. (2009). Accelerating adaptive trade-off model using shrinking space technique for constrained evolutionary optimization. *International Method for Numerical Methods in Engineering*, *77*(11), 1501–1534.
- Wang, Y., Cai, Z., Guo, G., & Zhou, Y. (2007). Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems. *IEEE Transactions on Systems Man and Cybernetics, Part B: Cybernetics*, *37*(3), 560–575.
- Yang, X.-S. (2009). Firefly algorithms for multimodal optimization. In O. Watanabe, & T. Zeugmann (Eds.), *Stochastic algorithms: Foundations and applications. Lecture notes in computer science* (Vol. 5792, pp. 169–178). Berlin: Springer.
- Yi, J., Li, X., Chu, C.-H., & Gao, L. (2016). Parallel chaotic local search enhanced harmony search algorithm for engineering design optimization. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-016-1255-5.
- Yi, W., Zhou, Y., Gao, L., Li, X., & Zhang, C. (2016). Engineering design optimization using an improved local search based epsilon differential evolution algorithm. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-016-1199-9.
- Yu, K., Wang, X., & Wang, Z. (2016). An improved teaching–learning-based optimization algorithm for numerical and engineering optimization problems. *Journal of Intelligent Manufacturing*, *27*, 831–843.
- Zhang, J. Q., & Sanderson, A. C. (2009). JADE: adaptive differential evolution with optional external archive. *IEEE Transactions on Evolutionary Computation*, *13*(5), 945–958.
- Zhang, M., Luo, W., & Wang, X. F. (2008). Differential evolution with dynamic stochastic selection for constrained optimization. *Information Sciences*, *178*(15), 3043–3074.