Information Sciences 378 (2017) 215–241

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# CADE: A hybridization of Cultural Algorithm and Differential Evolution for numerical optimization

Noor H. Awad[a], Mostafa Z. Ali[b,*], Ponnuthurai N. Suganthan[a], Robert G. Reynolds[c]

[a] School of Electrical and Electronic Engineering, NTU, 639798, Singapore
[b] Jordan University of Science and Technology, Irbid 22110, Jordan
[c] Wayne State University, Detroit, MI 48202, USA

ARTICLE INFO

*Article history:*
Received 10 July 2016
Revised 17 October 2016
Accepted 22 October 2016
Available online 27 October 2016

*Keywords:*
Evolutionary algorithm
Cultural algorithm
Differential evolution
Numerical optimization

ABSTRACT

Many real-world problems can be formulated as optimization problems. Such problems pose a challenge for researchers in the design of efficient algorithms capable of finding the best solution with the least computational cost. In this paper, a new evolutionary algorithm is proposed that combines the explorative and exploitative capabilities of two evolutionary algorithms, Cultural Algorithm (CA) and Differential Evolution (DE) algorithm. This hybridization follows the HTH (High-level Teamwork Hybrid) nomenclature in which two meta-heuristics are executed in parallel. The new algorithm named as CADE, manages an overall population which is shared between CA and DE simultaneously. Four modified knowledge sources have been used in proposed CA which are: topographical, situational, normative and domain. The role of the used acceptance function in belief space is to select the knowledge of the best individuals to update the current knowledge. A novel quality function is used to determine the participation ratio for both CA and DE, and then a competitive selection takes place in order to select the proportion of function evaluations allocated for each technique. This collaborative synergy emerges between the DE and CA techniques and is shown to improve the quality of solutions, beyond what each of these two algorithms alone. The performance of the algorithm is evaluated on a set of 50 scalable optimization problems taken from two sources. The first set of 35 came from existing benchmark sets available in the literature. The second set came from the 2014 IEEE Single Function optimization competition. The overall results show that CADE has a favorable performance and scalability behaviors when compared to other recent state-of-the-art algorithms. CADE's overall performance ranked at number 1 for each of the two sets of problems. It is suggested that CADE's success across such a broad spectrum of problem types and complexities bodes well for its application to new and novel applications.



* Corresponding author.
*E-mail addresses:* noor0029@ntu.edu.sg (N.H. Awad), mzali.pn@gmail.com (M.Z. Ali), epnsugan@ntu.edu.sg (P.N. Suganthan), Robert.reynolds@wayne.edu (R.G. Reynolds).

http://dx.doi.org/10.1016/j.ins.2016.10.039
0020-0255/

## 1. Introduction

Many real life problems can be formulated as optimization problems. Hence, a considerable amount of research has been devoted to the location of a global optimal solution. Generally, any optimization problem can be expressed as follows:

$$\text{Minimize} : f(\vec{X}),\ \vec{X} = \{x_1, x_2, ..., x_D\},\ x_i \in [x_l, x_u], \tag{1}$$

where $f(x)$ is the objective function being optimized over $\vec{X}$, $D$ is the dimensionality of the problem, and $x_l$, $x_u$ are the lower and upper bounds of parameter $x_i$ respectively.

In order to solve such problems, many nature-inspired methods have been suggested such as biological Evolutionary Computation (EC) [33], Tabu Search (TS) [18], Genetic Programming (GA) [21], Simulated Annealing (SA) [24], Practical Swarm Optimization (PSO) [22], and Differential Evolution (DE) [36]. The Cultural Algorithm (CA) provides a powerful tool for the solution of sophisticated problems and has successfully been applied to many optimization problems [31,23,26,37]. Cultural Algorithms can be defined as a cultural evolutionary model with two components, namely a belief space and population space, along with a set of communication protocols that coordinate the interaction between the two spaces. However, an issue facing all evolutionary algorithms is the potential for premature convergence [34,45,20,19]. In order to overcome this issue, researchers usually enhance the performance of their algorithms by the integration of new and efficient techniques that improve the search algorithm or merge other algorithms or local search methods with their algorithm to form hybrid algorithms. For example, in [30], an iterated local search method is merged with the CA that incorporates shared knowledge from pre-defined multi-populations. Knowledge is then disseminated between the sub-populations at certain time intervals to share their performance and use that information to affect the evolution of future generations.

Another tactic is the use of nature inspired algorithms to form the population space in CAs in order to increase the diversity of the population, and guide the search towards promising regions. For example, in [12] the authors used a PSO population-based CA while other researchers have used a DE in the population space of the CA as in [4]. The authors in [3] proposed a CA-ImLS algorithm that embedded an improved local search that consists of combining five parallel local searches in the CA in order to direct the search in multiple directions. Another approach that combined the Niche CA with Tabu search is proposed in [2]. That approach employed a niche clearing procedure in order to select the search agents. On the other hand, the authors in [1] introduced three different variants of CAs in which they embedded a niche model so as maintain multiple groups within the population and used it to solve a set of well-known engineering problems.

DE is a meta-heuristic algorithm that uses mutation and crossover schemes for real-valued optimization problems [36]. The DE algorithm is highly dependent on the setting of control parameters such as the scaling factor ($F$), and crossover rate ($CR$). From this point of view, many DE techniques have been proposed to overcome this issue by adaptively selecting these parameters so as to enhance the performance of the original algorithm [43,42,46,28]. For example, the self-adaptive DE algorithm (SaDE) updates the $F$ and $CR$ values based on performance in previous generations [32]. The jDE algorithm uses different $F$ and $CR$ values for each individual in the population space, and updates them using some heuristic rules [5]. JADE introduced a new mutation strategy with an optional archive and updated control parameters in an adaptive manner [48]. Other algorithms that hybridized DE with local searches and/or other meta-heuristic algorithms were suggested to enhance the performance and guide the search towards optimal solutions. For example, SaDE-MMTS is an algorithm that hybridized the SaDE algorithm with a modified multiple trajectory search that used self-adaptive sizes and a clearing procedure to select search agents [50] in the context of solving large-scale global optimization problems. Next, the authors in [43] provided a comprehensive review of the existing hybrid DE algorithm with PSO. The paper constructed a systematic taxonomy for the hybridization of both DE and PSO with several classification mechanisms for hybridization strategies. These five factor levels which are: hybridization level, relationship between parent optimizers, operating order, type of transferred information and type of information transfer. Those factors build a systematic taxonomy for hybridising different optimisers, especially population-based meta-heurisitcs such as DE and PSO. An example of hybrid approach that uses the CA and DE to predict crystalline structure solution from powder diffraction data can be found in [10].

Most of these hybridization techniques suffer from a lack of balance between the exploration and exploitation phases during the optimization process. In [9], an optimal contraction theorem for exploration and exploitation in optimization was given. In the CADE algorithm presented in this paper, the balance between exploration and exploitation is guaranteed by introducing a quality function. This function controls the execution of both CA and DE algorithms based upon their successes in previous generations. The motivation of the work reported in this paper is to provide a new hybrid mechanism that can benefit from the strengths of two evolutionary algorithms in order to accelerate their combined performance for solving different optimization problems. In this paper, we introduce a novel hybrid algorithm, namely CADE that combines the capabilities of two evolutionary algorithms: *A customized version of CA* and the well-known DE. In the canonical CA, Accept() function selects the best individuals that are used to update the belief space knowledge sources using the function Update(). After that, the Influence() function selects the knowledge sources to influence the evolution of the next generation of the population. In CAs, the major source for exploration is topographical knowledge. That knowledge is often developed in a top-down hierarchical manner. To support this side in the customized version of CA, DE can provide a complementary source of exploratory knowledge based upon the distribution of the entire population, and hence makes a perfect amalgam to enhance the search phases during the optimization process. This cooperation is facilitated in the following way. Both algorithms share the same population space and follow the High-level Teamwork hybrid (HTH) model in which two meta-heuristics are executed in parallel. A novel success-based quality function is used to determine which one of the algorithms will be

```
Begin
  set g=0
  initialize belief space, B^g
  initialize population space, P^g
  repeat
    evaluate P^g using Obj()
    update B^g using Accept()
    generate P^g using Influence()
    g=g+1
    select P^{g+1} from P^g
  until (termination condition met)
End
```

**Fig. 1.** Pseudo-code of Cultural Algorithm.

used to execute first and guide the search. Later proportions of evaluation for each component algorithm can be determined using the quality function based on the success of subpopulation (individuals that are currently under its guidance). The effectiveness of CADE algorithm is validated and compared with other algorithms using a benchmark set of 50 functions, which are organized into two experiments for a richer discussion and completeness.

The outline of the remainder of the paper is as follows. First, in Section 2 both the CA and DE algorithms are discussed. Next in Section 3, the proposed mechanism for the hybridization of the two approaches is elaborated. Section 4 describes the benchmark problems that will be used to examine the hybrid's performance, the parameter settings and the simulation results along with comparisons of the approach with other state-of-the-art algorithms. Finally, Section 5 summarizes the conclusions of this work.

## 2. Scientific background

### 2.1. Cultural Algorithm

As an extension of genetic algorithm, Reynolds [35] introduced CA as an evolutionary model that consists of belief and population spaces and a communication channel. Another component of the CA algorithm are the cultural knowledge sources that reside in the belief space that are used to influence the search of the individuals in the population. Five general categories of those knowledge sources include normative knowledge, topographical knowledge, domain knowledge, situational knowledge, and history or temporal knowledge. Fig. 1 describes the basic pseudo-code of the CA algorithm. The figure shows how the process is executed in each generation. Firstly, the objective function *Obj*() generates the population space individuals, and the *Accept*() function selects the best individuals that are used to update the belief space knowledge sources using the function *Update*(). After that, the *Influence*() function selects the knowledge sources to influence the evolution of the next generation of the population. More details on the used knowledge sources and how they affect the population as in the customized version of this proposed work will be given in Section 3.

### 2.2. Differential Evolution

DE [36] is an evolutionary algorithm (EA) that is used to solve optimization problems based on several choices of mutation and crossover strategies. It consists of a population of individual candidate solutions to which recombination, evaluation, and selection are iteratively applied. First the recombination produces a new candidate solution component based upon the weighted difference between two randomly selected population members that is added to a third member. This results in the perturbation of the population members relative to the spread of the whole population. Selection then allows self-organization of the sample space into areas of interest. This knowledge is compatible with the topographic knowledge source use CAs and can support its work for a better exploration during the search. DE is considered to be an effective global optimizer and a robust technique when used for the solution of a variety of real-world problems [47]. As such, it can be a potential source of topographic knowledge that is used by the Cultural to direct exploratory search.

DE configurations are described using the following terminology, DE/*x*/*y*/*z*: *x* is the solution that is perturbed such as "random" or "best"; *y* is the number of difference vectors used to perturb *x*. Each difference vector reflects the difference between two (randomly) selected but distinct population members. *z* represents the recombination operator used such as binomial (*bin*) or exponential (*exp*).

Five mutation strategies were implemented in the original DE [15,32] as described below:

$$DE/rand/1: \quad V_i^g = X_{r_1}^g + F \cdot X\left(X_{r_2}^g - X_{r_3}^g\right) \tag{2}$$

$$DE/best/1: \quad V_i^g = X_{best}^g + F \cdot \left(X_{r_1}^g - X_{r_2}^g\right) \tag{3}$$

$$DE/current-to-best/2: \quad V_i^g = X_i^g + F \cdot \left(X_{best}^g - X_i^g + X_{r_1}^g - X_{r_2}^g\right) \tag{4}$$

$$DE/best/2: \quad V_i^g = X_{best}^g + F \cdot \left(X_{r_1}^g - X_{r_2}^g + X_{r_3}^g - X_{r_4}^g\right) \tag{5}$$

$$DE/rand/2: \quad V_i^g = X_{r_1}^g + F \cdot \left(X_{r_2}^g - X_{r_3}^g + X_{r_4}^g - X_{r_5}^g\right) \tag{6}$$

where $V_i = \{v_i^1, v_i^2, ..., v_i^D\}$ is the mutant vector which is generated for each individual $X_i$ in the population space. $r_1$, $r_2$, $r_3$, $r_4$, $r_5$ represent random and mutually exclusive integers generated within the range [1, *NP*], *NP* is the population size. *F* is the scale factor that represents a positive number for controlling the scaling ratio of the difference vector. $X_{best}$ is the solution with the best fitness found so far by the search.

The original DE has defined a binomial (uniform) crossover as follows:

$$u_i = \begin{cases} v_i, \ if\ (with\ probability\ of\ CR)\ or\ (j = j_{rand}) \\ x_i, \ otherwise \end{cases}, \tag{7}$$

where $j = 1, 2... D$, and $u_i$ is the generated offspring. *CR* is the crossover rate which is a user-defined value. *CR* takes on values in the range [0,1] and helps the DE in controlling the copied portion of the mutant vector into the offspring/trial vector.

## 3. Proposed algorithm

A complete taxonomy of hybrid algorithms can be found in [38] in which two levels are used to derive four classifications. The two levels are: hierarchical level and flat level. The hierarchical level is used to reduce the number of classes whereas the flat level defines the arbitrary order of the chosen technique. Based on this, four different classifications for hybrid algorithms are derived as follows:

- LRH: Low-level Rely Hybrid
- HRH: High-level Rely Hybrid
- LTH: Low-level Teamwork Hybrid
- HTH: High-level Teamwork Hybrid

A hybrid algorithm follows the LRH model if one technique is embedded into a single-solution algorithm. It belongs to LTH if it is embedded into a population based algorithm. If the two algorithms are executed in sequence, then the hybrid algorithm follows HRH. It belongs to HTH if the two are executed in parallel.

The hybrid algorithm proposed here, CADE, belongs to the HTH model in which the CA and the DE algorithm are combined together. An HTH model is preferred when both algorithms are population-based algorithms in order to share one population simultaneously. Additionally, the diversity of the shared population is increased through the use of different modification schemes for its individuals. This helps to reduce the possibility of premature convergence and better guide the search toward promising regions. The following sub-sections explain the proposed algorithm in details.

Referring to proposed taxonomy in [44] and the five factor levels which have been proposed, the proposed algorithm defines the individuals of shared population between CA and DE as the parents of CADE algorithm. The relationship between those parent optimizers is defined when one shared population is used to set the participation ratio for both CA and DE. Hence, it is a collaborative relationship in which the strengths of the parents will be shared and migrate between different optimizers. The level of hybridization is at the population level since the population is used to define the parent operators for each CA and DE optimizers. A parallel operating order is used as we used the HTH model in CADE algorithm. To maintain an interactive influence between DE and CA, a duplex of type of information transfer is used and this satisfied by using a shared population in which the parent's information is transferred between both CA and DE before defining the participation ratio for each of them. The type of transferred information for DE are mutant vectors, trial vectors and the difference vectors while it is the so-far-best solution for CA.

### 3.1. The Cultural Algorithms

The belief space of the CA represents the repository of best knowledge that is used to guide the optimization process. In our algorithm, the belief space is composed of four knowledge sources that were tested to be the most influencing on the individuals during the evolution of the search for the used benchmarks. These knowledge sources are described next.

Topographical Knowledge (TK): Topographical Knowledge is the information about the spatial patterns for the best region of the search space. For the efficiency of memory management, we use the *k*-d tree (*k*-dimensional binary tree) in which each node can only have two children. This modification of the data structure utilizes the process of dividing any dimension in half and uses spatial characteristics during the optimization process. *TK* is viewed as a regional schemata and is modelled

**Algorithm:** Topographic Knowledge (*TK*)

**If** search is progressing
  **If** *curr_cell* is good
    $Y = mutate(X)$
  **Else**
    Choose another cell from the set $1 \le j \le s$
    $Y = \text{mutate}(p\_cb_j)$
  **Endif**
**Else**
  **If** $f(X) < f(p\_cb_h)$
    $Y = \text{mutate}(X)$
  **Else**
    Choose $C_t$ from the upper level cells
    $Y = mutate(p\_cb_t)$
  **Endif**
**Endif**

**Fig. 2.** Topographic knowledge in the modified CA.

**Algorithm:** Situational Knowledge (*SK*)

**While** ($j <= D$)
  **If** $x_i < gbest_i$
    $y_i = Rnd(x_i, gbest_i)$
  **Else if** $x_i > gbest_i$
    $y_i = Rnd(gbest_i, x_i)$
  **Else**
    $y_i = mutate(x_i)$
  **Endif**
  $j$+= 1
**End**

**Fig. 3.** Situational knowledge in the modified CA.

as a multidimensional grid, where the cells in this grid are represented as $sch_{c1}, sch_{c2},\ldots, sch_{cn}$. $sch_{ci}$ is the size of the cell for the $i$ th dimension. Each cell points to its children, and it has a value for the lower and upper bounds corresponding to the n dimensions. Those represent the ranges of the best solutions that were located in that cell. Assuming that the search space is initially divided into s cells $\langle sch_{c1}, sch_{c2},\ldots, sch_{cs}\rangle$, and the parent cell best agent $p_{cb_h}$, the cells will be divided further into a smaller set of cells. The pseudocode for the update mechanism for this knowledge source is given in Fig. 2. In this figure and the following ones, *mutate* $(X)$ is a Gaussian-randomly generated number with the mean $x$. Hence, $Y = mutate(X)$ will generate a value around X according to the distribution.

Situational Knowledge (SK): Situational knowledge maintains a set of cases that help the individuals move toward the exemplars. During the optimization process, situational knowledge saves the best exemplars and is responsible for generating new offspring based on their best knowledge [31]. Assuming that the global best individual is represented using the tuple $\langle gbest_1, gbest_2,\ldots, gbest_D\rangle$, then the methodology of generating the offspring under the influence of SK is as given in Fig. 3.

Normative Knowledge (NK): Normative knowledge represents standards and guidelines that can lead the individuals to promising new regions. It is responsible for keeping individuals in a region or moving them on to better regions by saving the acceptable behavior of promising individuals and their ranges [31]. Each variable will have a lower bound ($lb_i$) and an upper bound ($ub_i$) that are adjusted as the evolution progresses over time. The offspring is generated using the rule in Fig. 4.

Domain Knowledge (DK): The domain knowledge component employs problem domain information to guide the search process. It saves the features, parameters, and domain ranges of the best individuals from the population. It can also contain

**Algorithm:** Normative Knowledge (*NK*)

**While** ($j <= D$)

  **If** $x_i \notin (lb_i, ub_i)$

    $y_i = Rnd(lb_i, ub_i)$

  **Else**

    $y_i = mutate(x_i)$

  **Endif**

  $j += 1$

**End**

**Fig. 4.** Normative knowledge in the modified CA.

**Algorithm:** Domain Knowledge (*NK*)

**If** $max\nabla\ (X) > 0$

  $Y = mutate\_towards(X)$

**Else**

  $Y = mutate\_towards(gbest)$

**Endif**

**Fig. 5.** Domain knowledge in the modified CA.

relationships between problem parameters such as constraints as demonstrated below. The update method is similar to situational knowledge except that the re-initialization does not happen at every change of the search landscape [31]. This knowledge source is useful for dynamic landscapes, especially in terms of gradients of inclines and declines in the landscape. If one assumes a displacement disp as $\langle s_1, s_2, \ldots, s_n \rangle$ where $s_i$ takes on the values $-1$, 0, or 1 for a decreasing, same, or increasing $x_{ij}$. The gradient $\nabla(disp)$ is defined as:

$$\nabla(disp) = (obj(X + disp) - obj(X))/disp\ . \tag{8}$$

This knowledge source keeps track of the direction that leads to the fastest advancement in fitness ($max\nabla$), and the displacement. In this proposed work, this knowledge will have a light effect as we are dealing with static landscapes. The update process is given as in Fig. 5. In this figure, *mutate_towards*($X$) function is to mutate X in its steepest direction while *mutate_towards*(*gbest*) is to mutate the global best in the global best's steepest direction.

### 3.2. The differential evolution component

At the early stages of DE algorithm development, DE/rand/1 was the most widely used mutation strategy [36, 32]. However, other studies [48] suggest that greedy strategies such DE/best and DE/current-to-pbest, $p \in (0,\ 1]$, that use information about the best solution have some advantages for guiding the evolutionary search towards the promising regions. This mutation is a generalization of DE/current-to-best. Individuals are chosen randomly from the top 100$p$% solutions to play the role of the single best solution in DE/current-to-best, with the binomial crossover. The mutant vector is generated as shown in Eq. (4) where $X_{best}^{p,g}$ is the best individual in sub-population of the DE technique. At each generation $g$, the scaling factor $F_i$ is independently generated for each individual $x_i$ using a Cauchy distribution as $F_i = Cauchy(lop, 0.1)$ where $Cauchy(lop, 0.1)$ is a random sample from a Cauchy distribution with a location parameter $lop$, and scale parameter that is equal to 0.1. The value of $F_i$ is regenerated if $F_i < 0$ or $F_i > 1$. The Cauchy distribution is used to diversify the mutation factors. This will help avoid premature convergence that often occurs when using a mutation such as "current-to-pbest" especially at later generations [48]. Similarly, the crossover rate is generated under a normal distribution $CR_i = normal(CR, 0.1)$. A pseudo-code for the DE component used here is presented in Fig. 6.

### 3.3. Overall CADE algorithm

The goal of the proposed algorithm is to provide a basic framework for the hybridization that uses a dynamic quality function to regulate the interaction between the modified CA algorithm and the canonical DE. The simplicity of the framework supports its potential extension to other types of algorithms and adaptations in parameters. The proposed algorithm, CADE, manages an overall population space which is shared by CA and DE at the same time. The population space $P$ of the CADE algorithm at generation $t$ is represented as:

$$X_i^g = \left\langle x_{i,1}^g,\ x_{i,2}^g,\ \ldots,\ x_{i,D}^g \right\rangle,\ \ i = 1, 2, \ldots, NP, \tag{9}$$

**Algorithm:** Differential Evolution Component

**Do**

**For** each individual $X_i$ in the population

Generate three random integers $r_1, r_2 \in NP$ with $r_1 \neq r_2 \neq j$

Generate $F_i$ using Cauchy distribution

For each parameter $j$

$$V_{i,j}^{g} = X_{i,j}^{g} + F_i \cdot \left( X_{best}^{p,g} - X_{i,j}^{g} + X_{r_1,j}^{g} - X_{r_2,j}^{g} \right)$$

**EndFor**

Generate $CR_i$ using Normal Distribution

Apply binomial crossover on $V_i$ to generate $U_i$

Evaluate $U_i$ by calling objective function

Replace $X_i$ by $U_i$ if $U_i$ is better

**EndFor**

**Until the termination condition is met**

**Fig. 6.** Differential evolution complement in CADE.

where $NP$ is the size of the population, and $D$ is the dimension of the problem. Initially, the population should cover the search space of the problem being solved denoted by lower and upper boundaries, $X_l$ and $X_u$, respectively as follows:

$$x_{i,j} = x_{l,j} + rand(0,1) \cdot (x_{u,j} - x_{l,j}), \quad j = 1, 2, \ldots, D\,. \tag{10}$$

At the first generation $g_0$ and after the initial population $P_0$ is evaluated, the initial participation ratio $\Phi_T^{g_0}$ of each technique $T$, where $T \in [DE,\ CA]$, is computed as follows:

$$\Phi_T^{g_0} = \frac{NP}{|T|}, \tag{11}$$

and $|T|$ is the number of techniques used in the hybrid algorithm which is 2 in our case. A similar discussion can be used with the knowledge sources (four KSs) and their success and choice during future search. Each technique has its own auxiliary population $PA_T^{g_0}$ that is populated initially by selecting random individuals from the shared population $P$ according to its participation ratio. After that, each technique evolves its auxiliary population to produce a new subset of individuals and evaluates them. In the next step, a competitive comparison between the two techniques is used to compute a quality function which determines the next participation ratio $\Phi_T^g$ for each technique. The number of successful trial vectors that can enter the population is denoted as $n_{s,T_k}^{g-1}$ and is defined as:

$$n_{s,T_k}^{g-1} = \sum_{i=0}^{N_{T_k}} S_{i,T_k}^{g-1}, \tag{12}$$

where $N_{T_k}$ is the size of auxiliary population generated by technique $T_k$, and

$$S_{i,T_k}^{g-1} = \begin{cases} 1 & if\ f(U_{i,T_k}^{g-1}) < f(X_{i,T_k}^{g-1}) \\ 0 & otherwise \end{cases} \tag{13}$$

A trial vector is considered a successful vector if its function value is better than its parent. The quality function $QF_{T_k}^g$ for any of the used component algorithm is defined as:

$$QF_{T_k}^g = \frac{\Delta f_{T_{best}}^{+,g} - \Delta f_{T_k}^{+,g}}{\Delta f_{T_{best}}^{+,g}}, \tag{14}$$

where $\Delta f_{T_{best}}^{+,g}$ is the difference in the amount of fitness of the best performing technique in the last two generations as computed in Eq. (15) where $a_{f^+,T_k}^{g-1}(X_i)$ is the amount of fitness increment of $T_k$ at generation $g$-1. According to the above quality function, the participation ratio of each technique is updated for the current generation. The quality function of two techniques is then compared and the best one is used to guide a larger number of individuals in the next generation.

$$\Delta f_{T_k}^{+} = \sum_{i=0}^{n_{S,T_k}^{g-1}-1} a_{f^+,T_k}^{g-1}(X_i), \tag{15}$$

**Algorithm:** CADE Algorithm

Generate initial overall shared population $P_0$ of $NP$ individuals

Generate initial parameters of each technique $T$ where $T \in [DE, CA]$ as described in previous subsections

Evaluate initial population $P_0$

Compute initial participation ratio $\Phi_{T_k}^{g_0}$ of each subcomponent algorithm $\longrightarrow \Phi_T^{g_0} = \frac{NP}{|T|}$

Produces a subset of new individuals by each technique $T$ according to $\Phi_{T_k}^{g_0}$

**While** termination criterion not met **Do**

Compute Quality function $QF_g^T$ of each $T$

Update participation ratio $\Phi_g^T$ of each $T$ according to $QF_g^T$

**For** every technique $t_k$ in $T$ **Do**

Create new individuals from current population $P_g$ using technique $t_k$

Evaluate new individuals

Add new individuals to an auxiliary population $PA_g^{tk}$

**End For**

Combine auxiliary populations of techniques to generate $P_g$

**End While**

**Fig. 7.** Pseudo-code of CADE algorithm.

The participation ratio of any of the component algorithms is defined as follows:

$$\Phi_{T_k}^g = QF_{T_k}^g . NP . \tag{16}$$

The number of new individuals created by each technique is proportional to its participation ratio. The auxiliary population $PA_T^g$ is filled with new individuals after they are evaluated. Next, the new population for the next generation is produced by combining the two auxiliary populations. The pseudo-code of CADE algorithm is shown in Fig. 7.

## 4. Experimental results

### 4.1. Numerical benchmarks – experiment 1

The performance of the proposed algorithm is evaluated on a test-suite of 50 bound constrained single-objective benchmark functions with different characteristics taken from the literature [48,27,11,7,25]. To avoid the bias towards being successful at a small set of functions, the test set includes 50 functions with a broad variety of problem environments to introduce variability and to be able to draw a general conclusion. These functions are organized into two experiments that include functions with different characteristics and difficulties. Validating the performance of the proposed work over two sets of experiments also makes it easier to compare with the most related state-of-the-art algorithms from the literature as will be seen from the results tables, in terms of the competent algorithms and those reported results from literature of the algorithms we compare with. The functions in experiment 1 are well-known and frequently used scalable functions from the literature [48] [27,11,7]. These functions are described in Tables 1 and 2. Experiment 2 includes 15 functions that are chosen to cover all categories from the test suite of the IEEE CEC'14 special competition on real-parameter single objective optimization [25]. These functions are described in Table 3 and are tested at the 50$D$ and 100$D$ scales in order to match the highest set of tested dimension from the competition.

The problems from experiment 1 contain a variety of high-dimensional problems such as step functions, unimodal and multimodal functions, and noisy quadratic functions. Except for some test functions with low dimensions, the remaining test functions are tested at 30$D$. Moreover a scalability study for the first 17 scalable functions is conducted at 100$D$ in this experiment, as indicated in the tables. Functions $f_1$–$f_{17}$ are high dimensional problems. Function $f_6$ is a step function which is discontinuous and has one minimum. Function $f_7$ is a noisy quadratic function. Functions $f_8$–$f_{13}$ represent multimodal functions where the number of local minima increases exponentially with increasing the dimensionality of the problem [7]. Functions $f_{18}$–$f_{35}$ are all low dimensional problems with many local minima.

**Table 1**
Functions $F_1$–$F_{17}$ from the 35 chosen functions in this experimental study. These functions are tested at 30$D$ and since they are scalable, they are also used in the scalability study for experiment 1 at 100$D$.

| Experiment 1 | | |
|---|---|---|
| Function description | Search range | Optimal |
| $f_1(x) = \sum_{i=1}^{D} x_i^2$ | $[-100, 100]^D$ | 0 |
| $f_2(x) = \sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} \lvert x_i\rvert$ | $[-10, 10]^D$ | 0 |
| $f_3(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$ | $[-100, 100]^D$ | 0 |
| $f_4(x) = \max_i\{\lvert x_i\rvert\}$ | $[-100, 100]^D$ | 0 |
| $f_5(x) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$ | $[-30, 30]^D$ | 0 |
| $f_6(x) = \sum_{i=1}^{D} \lfloor x_i + 0.5\rfloor^2$ | $[-100, 100]^D$ | 0 |
| $f_7(x) = \sum_{i=1}^{D} ix_i^4 + rand[0, 1)$ | $[-1.28, 1.28]^D$ | 0 |
| $f_8(x) = -\sum_{i=1}^{D} x_i \sin\sqrt{\lvert x_i\rvert} + D.418.9828872743369$ | $[-500, 500]^D$ | 0 |
| $f_9(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$ | $[-5.12, 5.12]^D$ | 0 |
| $f_{10}(x) = \sum_{i=1}^{D} -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e$ | $[-32, 32]^D$ | 0 |
| $f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}) + 1$ | $[-600, 600]^D$ | 0 |
| $f_{12}(x) = \frac{\pi}{D}\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2[1+10\sin^2(\pi y_i + 1)] + (y_D - 1)^2\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$ | $[-50, 50]^D$ | 0 |
| $f_{13}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2[1 + \sin^2(2\pi x_D)\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$ | $[-50, 50]^D$ | 0 |
| $f_{14}(x) = \sum_{i=1}^{D} \lvert x_i\rvert^{(i+1)}$ | $[-1, 1]^D$ | 0 |
| $f_{15}(x) = \sum_{i=1}^{D} ix_i^4$ | $[-1.28, 1.28]^D$ | 0 |
| $f_{16}(x) = \sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} 0.5ix_i)^2 + (\sum_{i=1}^{D} 0.5ix_i)^4$ | $[-10, 10]^D$ | 0 |
| $f_{17}(x) = \sum_{i=1}^{D} \lvert x_i\sin(x_i) + 0.1x_i\rvert$ | $[-10, 10]^D$ | 0 |

**Table 2**
Functions $F_{18}$–$F_{35}$ from the 35 chosen functions in this experimental study. These functions are tested at the stated dimension size of their original description.

| Experiment 1 (continued) | | | |
|---|---|---|---|
| Function description | $D$ | Search range | Optimal |
| $f_{18}(x) = [\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j+\sum(x_i - a_{ij})^6}]^{-1}$ | 2 | $[-65.536, 65.536]^D$ | 1 |
| $f_{19}(x) = \sum_{i=1}^{11}[a_i - \frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}]^2$ | 4 | $[-5, 5]^D$ | 0.003075 |
| $f_{20}(x) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1x_2 + (4x_2^2 - 4)x_2^2$ | 2 | $[-5, 5]^D$ | −1.0316285 |
| $f_{21}(x) = (x_2^2 - \frac{5.1}{4\Pi^2}x_1^2 + \frac{5}{\Pi}x_1) + 10(1 - \frac{1}{8\Pi})\cos(x_1) + 10$ | 2 | $[-5, 10]^D$ | 0.398 |
| $f_{22}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_2^2 - 14x_2 + 6x_1x_2 + 3x_2^2)][30 + (2x_1 - 3x_2^2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$ | 2 | $[-2, 2]^D$ | 3 |
| $f_{23}(x) = 100(x_2 - x_1^2)^2 + (1 - x_1^2)^2 + 90(x_4 - x_3^2)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$ | 4 | $[-10, 10]^D$ | 0 |
| $f_{24}(x) = \{\sum_{i=1}^{5} i\cos[(i+1)x_1 + i]\}\{\sum_{j=1}^{5} j\cos[(j+1)x_2 + j]\}$ | 2 | $[-10, 10]^D$ | −186.7309 |
| $f_{25}(x) = [1.5 - x_1(1 - x_2)]^2 + [2.25 - x_1(1 - x_2^2)]^2 + [2.652 - x_1(1 - x_2^3)]^2$ | 2 | $[-4.5, 4.5]^D$ | 0 |
| $f_{26}(x) = -\cos x_1\cos x_2\exp(-(x_2 - \Pi)^2 - (x_1 - \Pi)^2)$ | 2 | $[-100, 100]^D$ | −1 |
| $f_{27}(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$ | 2 | $[-10, 10]^D$ | 0 |
| $f_{28}(x) = 0.5 + \frac{\sin^2\sqrt{x_1^2+x_2^2} - 0.5}{1+0.01(x_1^2+x_2^2)^2}$ | 2 | $[-10, 10]^D$ | 0 |
| $f_{29}(x) = -\sum_{i=1}^{4} c_i\exp(-\sum_{j=1}^{n} a_{ij}(x_j - p_{ij})^2)$ | 3 | $[0, 1]^D$ | −3.86 |
| $f_{30}(x) = -\sum_{i=1}^{4} c_i\exp(-\sum_{j=1}^{n} a_{ij}(x_j - p_{ij})^2)$ | 6 | $[0, 1]^D$ | −3.32 |
| $f_{31}(x) = -\sum_{i=1}^{5}[(x - a_i)^T + c_i]^{-1}$ | 4 | $[0, 10]^D$ | −10.1532 |
| $f_{32}(x) = -\sum_{i=1}^{7}[(x - a_i)^T + c_i]^{-1}$ | 4 | $[0, 10]^D$ | −10.40294 |
| $f_{33}(x) = -\sum_{i=1}^{10}[(x - a_i)^T + c_i]^{-1}$ | 4 | $[0, 10]^D$ | −10.53641 |
| $f_{34}(x) = x_1^2 + x_2^2 - \cos 18x_1 - \cos 18x_2$ | 2 | $[-1, 1]^D$ | −2 |
| $f_{35}(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\Pi x_1) - \cos(3\Pi x_2) + 0.7$ | 2 | $[-1.28, 1.28]^D$ | 0 |

#### *4.1.1. Compared algorithms and parametric setup*

The performance of CADE is compared with six other algorithms that include a classical DE and classical CA variants and four other state-of-the-art DE and CA algorithms, as follows:

1. Classic DE/rand/1/bin [36], the parameters for $F$ and $CR$ are set to 0.5 as recommended in [36,32], as well as we found that such setup gives the best overall results on these benchmark functions.
2. Original CA with five knowledge sources which are: Topographical KS, Situational KS, Normative KS, Domain KS and History KS [31]
3. JADE algorithm [48] in which we used two variants: the JADE without archive for $D = 30$ and JADE with archive for $D = 100$ as suggested in [48].
4. jDE algorithm [5], with $F_l = 0.1$, $F_u = 0.9$, and $\tau_1 = \tau_2 = 0.1$.
5. CA-ImLS algorithm [3].

**Table 3**
Experiment 2: functions $F_{36}$–$F_{50}$ taken from the IEEE CEC'14 special session & competition on real-parameter single objective optimization [28]. These functions are tested at 50*D* and 100*D*.

| Experiment 2 | | | | |
|---|---|---|---|---|
| Serial no. | Function name | Function description | Search range | Optimal |
| $f_{36}$ | Rotated high conditioned elliptic function | Corresponds to $f_1$ of [28] | $[-100, 100]^D$ | 100 |
| $f_{37}$ | Shifted and rotated Rosenbrock's function | Corresponds to $f_4$ of [28] | $[-100, 100]^D$ | 400 |
| $f_{38}$ | Shifted and rotated Ackley's function | Corresponds to $f_5$ of [28] | $[-100, 100]^D$ | 500 |
| $f_{39}$ | Shifted and rotated Weierstrass function | Corresponds to $f_6$ of [28] | $[-100, 100]^D$ | 600 |
| $f_{40}$ | Hybrid function 1 ($N=3$) | Corresponds to $f_{17}$ of [28] | $[-100, 100]^D$ | 1700 |
| $f_{41}$ | Hybrid function 2 ($N=3$) | Corresponds to $f_{18}$ of [28] | $[-100, 100]^D$ | 1800 |
| $f_{42}$ | Hybrid function 3 ($N=4$) | Corresponds to $f_{19}$ of [28] | $[-100, 100]^D$ | 1900 |
| $f_{43}$ | Hybrid function 4 ($N=4$) | Corresponds to $f_{20}$ of [28] | $[-100, 100]^D$ | 2000 |
| $f_{44}$ | Hybrid function 5 ($N=5$) | Corresponds to $f_{21}$ of [28] | $[-100, 100]^D$ | 2100 |
| $f_{45}$ | Composition function 1 ($N=5$) | Corresponds to $f_{23}$ of [28] | $[-100, 100]^D$ | 2300 |
| $f_{46}$ | Composition function 2 ($N=3$) | Corresponds to $f_{24}$ of [28] | $[-100, 100]^D$ | 2400 |
| $f_{47}$ | Composition function 3 ($N=3$) | Corresponds to $f_{25}$ of [28] | $[-100, 100]^D$ | 2500 |
| $f_{48}$ | Composition function 4 ($N=5$) | Corresponds to $f_{26}$ of [28] | $[-100, 100]^D$ | 2600 |
| $f_{49}$ | Composition function 5 ($N=5$) | Corresponds to $f_{27}$ of [28] | $[-100, 100]^D$ | 2700 |
| $f_{50}$ | Composition function 6 ($N=5$) | Corresponds to $f_{28}$ of [28] | $[-100, 100]^D$ | 2800 |

6. HS-CA algorithm [16] in which harmony search is merged with the Cultural algorithm.

In all of the experiments, *NP* was used as 40 in all of the algorithms, irrespective of the dimensionality of the problem. *F* and *CR* values are initially set to 0.5 in CADE. The use of adaptation for these parameters during the runs helps the DE to discover more promising regions and hence enhancing the performance of CA by using the shared population. As a result, the used parameter adaptation reflects the whole performance of the hybrid approach when compared to single DE or CA algorithms. The ratio of accepted individual's experiences into the belief space of the CA is set as 0.25. More discussion of the parameter setup will be presented in Section 4.2.1. Unless stated otherwise, the parameters for each of the algorithms were consistent with recommendations from the literature.

#### *4.1.2. Simulation approach*

Functions $f_1$ to $f_{17}$ are tested in 30*D*. These are scalable functions. In order to test how the performance of CADE changes with scaling the search space, we also test these functions in 100*D*. Functions $f_{18}$ to $f_{35}$ are tested over dimensions as indicated in Tables 1 and 2. The maximum number of function evaluations (FEs) that was used based on recommendations from previous sources is shown in Table 4. As indicated, maximum FEs used for CADE that is less than that for all of the other contestant algorithms since it converges faster over almost all problems. Different FEs limits were used for the 30*D* and 100*D* runs [48,27,11,7]. Only functions $f_{18}$ to $f_{35}$ are scalable functions and hence were used for the scalability study at 100*D*. All of the algorithm simulations were performed on an Intel® Core™ i7-4850HQ CPU, 2.3 GHz speed, and with 16 GB RAM. In order to have a fair start for all the algorithms, they all used the same initial population in each run. This makes any difference in their performances be attributed to their internal search operators.

#### *4.1.3. Results*

The mean and standard deviation of the results for 50 independent runs for each of the seven algorithms run on 35 benchmark functions for the specified dimensions as indicated in Tables 1 and 2 are shown in Table 5. As can be seen in this table, CADE performed at least as well as the other algorithms in all the 35 functions. CADE obtained a superior performance compared to the other algorithms in over half (18) of the 35 functions. It obtained an equal performance in 17 functions out of the 35 used, namely $f_2$, $f_6$, $f_8$, $f_9$, $f_{11}$, $f_{18}$, $f_{20}$, $f_{22}$, $f_{24}$, $f_{26}$, $f_{27}$, $f_{30}$–$f_{35}$. However, a comparison of their performances will not be complete using these statistics only. The significance of the differences between the results of the involved algorithms will be assessed with the statistical *t*-test at $\alpha = 0.05$, which is the most commonly used test on such functions from the literature. This will be used to get a first impression about the quality of the results. The *p*-values of this test will be reported. For the first set of results in Table 5, the *p*-values are reported in Table 6. In this table, if the results are too similar, the *p*-value was reported as "$> 0.9999$", and the test was extremely statistical significant then the *p*-value was reported as "$<$ 1.0E-15". Whenever these *p*-values are less than 0.05, it is a strong evidence against the acceptance of $H_0$, the null hypothesis that the results were produced by equivalent statistical processes.

In order to test the convergence rate in successful runs and the reliability of the algorithms under comparison, a summary of the success rate (SR) and the average number of function evaluations generated over successful runs (FESS), respectively, are presented in Tables 7–9. An experiment is marked as successful if the solution is found with adequate accuracy. For $f_1$–$f_{17}$, $f_{22}$, $f_{23}$, $f_{25}$–$f_{28}$, $f_{34}$, $f_{35}$ it is set as $10^{-8}$ and for $f_7$, $f_8$, $f_{18}$–$f_{21}$, $f_{24}$, $f_{29}$–$f_{33}$ it is set as $5 \times 10^{-3}$. CADE was able to obtain the best SR values with minimum FESS over all the functions. These settings are consistent with those in the literature. CADE was able to obtain the best SR values with minimum FESS over all of the 35 functions.

**Table 4**
Function evaluations of the benchmark functions at dimensions as indicated in Table 1 and Table 2 in addition to scalability over 100D for scalable functions ($f_1$–$f_{17}$), averaged over 50 runs.

| | FEs: CADE $D$ as in Tables 1 and 2 | FEs: other algorithms $D$ as in Tables 1 and 2 | FEs: CADE $D = 100$ | FEs: other algorithms $D = 100$ |
|---|---|---|---|---|
| $f_1$ | 2.8E4 | 1.5E5 | 6.9E4 | 8.0E5 |
| $f_2$ | 1.8E4 | 2.0E5 | 3.9E4 | 1.2E6 |
| $f_3$ | 3.5E4 | 5.0E5 | 5.2E4 | 3.2E6 |
| $f_4$ | 2.1E5 | 5.0E5 | 3.0E6 | 6.0E6 |
| $f_5$ | 3.0E5 | 2.0E6 | 6.5E5 | 8.0E6 |
| $f_6$ | 5.1E3 | 1.5E5 | 2.8E4 | 6.0E5 |
| $f_7$ | 2.8E4 | 3.0E5 | 5.6E5 | 2.4E6 |
| $f_8$ | 2.5E4 | 9.0E5 | 3.6E5 | 3.6E6 |
| $f_9$ | 1.1E5 | 5.0E5 | 7.8E5 | 3.6E6 |
| $f_{10}$ | 2.6E4 | 2.0E5 | 6.6E4 | 1.2E6 |
| $f_{11}$ | 4.6E4 | 3.0E5 | 7.8E4 | 1.2E6 |
| $f_{12}$ | 4.0E3 | 1.5E5 | 2.7E4 | 1.2E6 |
| $f_{13}$ | 2.0E4 | 1.5E5 | 5.6E4 | 1.2E6 |
| $f_{14}$ | 2.0E + 04 | 1.5E + 05 | 5.5E + 04 | 3.0E + 06 |
| $f_{15}$ | 1.0E + 05 | 3.0E + 05 | 1.0E + 05 | 5.0E + 05 |
| $f_{16}$ | 1.0E + 05 | 3.0E + 05 | 1.0E + 05 | 5.0E + 05 |
| $f_{17}$ | 1.0E + 05 | 5.0E + 05 | 1.0E + 05 | 7.0E + 05 |
| $f_{18}$ | 5.0E + 03 | 1.0E + 04 | NA | NA |
| $f_{19}$ | 1.0E + 04 | 4.0E + 04 | NA | NA |
| $f_{20}$ | 5.0E + 03 | 1.0E + 04 | NA | NA |
| $f_{21}$ | 5.0E + 03 | 1.0E + 04 | NA | NA |
| $f_{22}$ | 5.0E + 03 | 1.0E + 04 | NA | NA |
| $f_{23}$ | 1.0E + 04 | 3.0E + 04 | NA | NA |
| $f_{24}$ | 1.0E + 04 | 8.0E + 04 | NA | NA |
| $f_{25}$ | 1.0E + 04 | 1.0E + 04 | NA | NA |
| $f_{26}$ | 5.0E + 03 | 1.0E + 04 | NA | NA |
| $f_{27}$ | 5.5E + 03 | 1.0E + 05 | NA | NA |
| $f_{28}$ | 5.0E + 03 | 1.0E + 04 | NA | NA |
| $f_{29}$ | 5.0E + 03 | 1.0E + 04 | NA | NA |
| $f_{30}$ | 5.0E + 03 | 2.0E + 04 | NA | NA |
| $f_{31}$ | 5.0E + 03 | 1.0E + 04 | NA | NA |
| $f_{32}$ | 5.0E + 03 | 1.0E + 04 | NA | NA |
| $f_{33}$ | 5.0E + 03 | 1.0E + 04 | NA | NA |
| $f_{34}$ | 5.0E + 03 | 1.0E + 04 | NA | NA |
| $f_{35}$ | 5.0E + 03 | 1.0E + 04 | NA | NA |

Furthermore, an extensive statistical analysis is presented for the purpose of evaluating the statistical significance of the observed performance differences [13]. The idea is to use a statistical test procedure than can be used to rank the performance of a set of $S$ algorithms. This test should be able to signify differences in the performance ranking of at least two of the $S$ compared algorithms. Post hoc test analysis [13] was used to judge the cases in which any of the compared algorithms exhibits the largest significant variation from the base algorithm (control method).

Particularly, the Friedman test [14,17] was used to test the differences between the set $S$ related samples. The Friedman test is a non-parametric multiple comparison test, that is able to report significant differences between the behaviors of multiple algorithms. The rankings for the Friedman test are reported in Table 10. The test statistic of the Friedman test and its corresponding $p$-value is reported at the bottom of Table 10. The $p$-value that was obtained from the tests of the 35 benchmarks using the dimensions as specified in Tables 1 and 2 is equal to 1.3869E−10. The computed $p$-value strongly suggests that there is an extremely large statistically significant difference between the 7 compared algorithms, at $\alpha = 0.05$. Table 10 also highlights the ranking of all the compared algorithms. Friedman test assigns the highest rank to the best performing algorithm. CADE was able to obtain the highest rank compared to all of the other algorithms as can be seen in the first column of this table (when compared at dimensions as specified in Tables 1 and 2).

A comparison of the adjusted $p$-values using 4 tests, the Bonferroni–Dunn, Holm, Hochberg, and Hommel tests is presented in Table 11. The worst performing algorithm (CA in this case) was used as the control method against which the performance of each of the other algorithms was compared. As can be seen from that table, CADE algorithm obtained the most significant results at the specified significance level of $\alpha = 0.05$. The post hoc analysis was used to identify the cases where the proposed algorithm was able to show significant differences from the other algorithms. The results of the post hoc tests for Friedman are shown in Table 12. This table gives the tests for all pairs of the compared algorithms. Statistical significant entries are marked in boldface, and those most related tests that include CADE among the other algorithms have their rows highlighted in light grey. From the adjusted $p$-values of the tests in Table 12, it apparent that, at $\alpha = 0.05$, Nemenyi's, Holm's, Shaffer's reject null hypotheses 14-21. Bergmann's rejects hypotheses 13-21. JADE was found to have a close performance to CADE using these tests, despite the fact that CADE was able to obtain better function values in 18 functions

**Table 5**
Statistical results of all the algorithms on the benchmark functions over dimensions as indicated in Tables 1 and 2, averaged over 50 independent runs. The best entries are marked in boldface.

| | DE/rand/1/bin | JADE | jDE | CA | CA-ImLS | HS-CA | CADE |
|---|---|---|---|---|---|---|---|
| $f_1$ | 9.80E−14(8.40E−14) | 1.80E−60(8.40E−60) | 2.50E−28(3.50E−28) | 9.30E−13(2.70E−12) | 6.64E−53(9.38E−53) | 7.80E−41(3.95E−42) | **0.00E+00(0.00E+00)** |
| $f_2$ | 1.60E−09(1.10E−09) | 1.80E−25(8.80E−25) | 1.50E−23(1.00E−23) | 1.20E−09(3.10E−09) | 1.03E−27(7.44E−27) | 3.52E−21(9.15E−21) | **0.00E+00(0.00E+00)** |
| $f_3$ | 6.60E−11(8.80E−11) | 5.70E−61(2.70E−60) | 5.20E−14(1.10E−13) | 2.04E−10(4.12E−10) | 2.15E−66(8.95E−66) | 3.43E−72(7.46E−71) | **0.00E+00(0.00E+00)** |
| $f_4$ | 4.20E−01(1.10E+00) | 8.20E−24(4.00E−23) | 1.40E−15(1.00E−15) | 3.00E−02(3.00E−02) | 2.27E−41(4.45E−40) | 6.45E−53(7.82E−52) | **0.00E+00(0.0E+00)** |
| $f_5$ | **8.00E−02(5.60E−01)** | **8.00E−02(5.60E−01)** | **8.00E−02(5.60E−01)** | 1.50E+01(1.60E+01) | 7.80E+00(7.86E−01) | 5.34E+00(2.91E−01) | **8.00E−02(5.60E−01)** |
| $f_6$ | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | 1.50E−03(1.50E−03) | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** |
| $f_7$ | 4.70E+03(1.20E+03) | 6.40E−04(2.50E−04) | 3.30E−03(8.50E−04) | 4.80E-03(1.50E−03) | 1.28E−01(7.91E-02) | 6.92E−02(7.61E−02) | **5.30E−05(3.40E-05)** |
| $f_8$ | 5.70E+01(7.60E+01) | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | 5.60E+03(4.30E+03) | 5.26E+00(1.75E+00) | 7.41E+00(1.25E+00) | **0.00E+00(0.00E+00)** |
| $f_9$ | 7.10E+01(2.10E+01) | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** |
| $f_{10}$ | 5.90E+03(1.10E+03) | 4.40E−15(0.00E+00) | 4.70E−15(9.60E−16) | 1.50E−06(4.60E−06) | 4.66E−09(2.46E−09) | 2.85E−07(8.55E−07) | **0.00E+00(0.00E+00)** |
| $f_{11}$ | 9.70E−11(5.00E−11) | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | 1.08E−03(7.64E−03) | 4.11E−09(1.07E−10) | 3.10E−08(3.94E−11) | **0.00E+00(0.00E+00)** |
| $f_{12}$ | 1.10E−14(1.00E−14) | 1.60E−32(5.50E−48) | 2.60E−29(7.50E−29) | 2.27E++01(8.75E+01) | 4.43E−21(1.99E−24) | 3.43E−22(6.89E−24) | **1.50E−32(5.52E−48)** |
| $f_{13}$ | 7.50E−14(4.80E−14) | 1.40E−32(1.10E−47) | 1.90E−28(2.20E−28) | 5.30E−02(3.10E−02) | 1.73E−08(2.90E−09) | 7.15E−10(1.68E−11) | **0.00E+00(0.00E+00)** |
| $f_{14}$ | 1.76E−35(3.42E−34) | 5.37E−82(3.75E−81) | 3.29E−92(1.67E−91) | 5.93E−09(3.11E−09) | 1.27E−37(4.71E−13) | 2.37E−35(1.22E−36) | **0.00E+00(0.00E+00)** |
| $f_{15}$ | 3.66E−62(8.32E−61) | 1.19E−133(5.93E−133) | 5.30E−104(1.06E−103) | 2.00E−62(6.34E-62) | 2.33E−68(2.85E−68) | 4.73E−68(9.51E−67) | **0.00E+00(0.00E+00)** |
| $f_{16}$ | 3.9E−07(2.24E-07) | 7.22E−49(4.25E−48) | 7.77E−12(2.09E-11) | 1.42E−19(3.65E−18) | 6.24E−28(5.83E-28) | 1.79E−33(4.27E−35) | **0.00E+00(0.00E+00)** |
| $f_{17}$ | 9.12E−12(8.52E-11) | 5.78E−12(2.47E−12) | 9.04E−16(6.23E−15) | 3.42E−12(5.09E−13) | 9.26E-16(2.76E−16) | 3.17E−15(1.66E-15) | **0.00E+00(0.00E+00)** |
| $f_{18}$ | **1.00E+00(3.62E−15)** | 9.98E−01(1.53E−14) | 9.98E-01(2.38E−16) | **1.00E+00(3.29E−15)** | **1.00E+00(3.00E-19)** | **1.00E+00(4.64E−17)** | **1.00E+00(2.18E−23)** |
| $f_{19}$ | 4.85E−03(3.80E−17) | 3.07E−04(2.46E−19) | 3.07E−04(1.35E−18) | 2.80E−03(4.06E−18) | 3.62E−03(5.48E−18) | 6.36E−03(4.24E−16) | **3.07E−03(1.10E−20)** |
| $f_{20}$ | **−1.0316285(8.35E−13)** | −1.0316284(3.68E−09) | −1.0316284(2.69E−11) | **−1.0316285(5.82E−15)** | **−1.0316285(3.57E−15)** | **−1.0316285(7.35E−16)** | **−1.0316285(5.02E−17)** |
| $f_{21}$ | 0.39789(4.26E−10) | 0.39788(1.02E−08) | 0.39788(9.84E−11) | 0.39789(3.50E−07) | 0.379782(9.01E−15) | 0.389783(4.55E−14) | **0.39794(8.19E−32)** |
| $f_{22}$ | **3.00E+00(7.87E−14)** | **3.00E+00(4.44E−13)** | **3.00E+00(2.04E−14)** | **3.00E+00(1.85E−14)** | **3.00E+00(4.97E−15)** | **3.00E+00(2.61E−15)** | **3.00E+00(3.24E−18)** |
| $f_{23}$ | 4.17E−20(5.92E−19) | 4.21E−07(2.83E-06) | 4.19E−12(1.20E−11) | 9.57E−21(9.64E−21) | 2.78E−31(5.46E−32) | 6.32E−28(9.74E−28) | **0.00E+00(0.00E+00)** |
| $f_{24}$ | **−186.7309(5.73E−14)** | **−186.7309(1.10E−13)** | **−186.7309(2.19E−14)** | **−186.7309(5.60E−14)** | **−186.7309(7.47E−16)** | **−186.7309(6.19E−14)** | **− 186.7309(2.02E-23)** |
| $f_{25}$ | 5.31E−27(6.15E−26) | 4.19E−13(6.73E−13) | 2.52E−15(5.86E-15) | 1.37E−26(2.99E−27) | 2.61E−28(4.63E−28) | 1.57E−30(7.90E−29) | **0.00E+00(0.00E+00)** |
| $f_{26}$ | **−1.00E+00(0.00E+00)** | **−1.00E+00(2.64E−09)** | **−1.00E+00(3.88E-11)** | **−1.00E+00(0.00E+00)** | **−1.00E+00(0.00E+00)** | **−1.00E+00(0.00E+00)** | **−1.00E+00(0.00E+00)** |
| $f_{27}$ | **0.00E+00(0.00E+00)** | 1.29E−137(5.41E−137) | 2.23E−151(1.32E−150) | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** |
| $f_{28}$ | 4.74E−16(2.90E−15) | 3.19E−10(6.40E−10) | 4.38E−11(2.58E-10) | 2.45E−17(5.76E−16) | 3.04E−30(2.20E-31) | 2.58E−29(5.92E−29) | **0.00E+00(0.00E+00)** |
| $f_{29}$ | −3.86278(6.89E−14) | −3.86277(3.67E−14) | −3.86277(2.09E−15) | −3.86272(1.84E−14) | −3.86245(7.81E−16) | −3.86144(4.19E−15) | **−3.86083(3.28E−17)** |
| $f_{30}$ | −3.312(4.16E−02) | −3.041(8.69E−03) | −3.041(8.69E−03) | −3.320(5.39E−03) | −3.326(2.96E−05) | **−3.328(4.74E−04)** | **−3.322(0.00E+00)** |
| $f_{31}$ | **−10.1532(9.78E−07)** | −10.0445(7.14E−01) | −10.0296(5.86E−01) | **−10.1532(6.47E−08)** | **−10.1532(5.18E−12)** | **−10.1532(9.03E−09)** | **−10.1532(3.26E−27)** |
| $f_{32}$ | **−10.40294(5.51E−09)** | −10.40288(8.93E−05) | −10.40242(1.23E−03) | **−10.40294(7.88E−09)** | **−10.40294(8.19E−11)** | **−10.40294(4.97E−08)** | **−10.40294(1.36E−19)** |
| $f_{33}$ | **−10.53641(4.35E−09)** | −10.53189(7.41E−04) | −10.53046(5.23E−03) | **−10.53641(1.62E−09)** | **−10.53641(3.82E−12)** | **−10.53641(2.59E−10)** | **−10.53641(4.28E−18)** |
| $f_{34}$ | **−2.00E+00(0.00E+00)** | **−2.00E+00(1.03E−12)** | **−2.00E+00(0.00E+00)** | **−2.00E+00(0.00E+00)** | **−2.00E+00(0.00E+00)** | **−2.00E+00(0.00E+00)** | **−2.00E+00(0.00E+00)** |
| $f_{35}$ | **0.00E+00(0.00E+00)** | 3.03E−14(3.85E−14) | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** |

**Table 6**
*P*-values calculated for the *t*-test (significance level 0.05) corresponding to the results of Table 4. The best entries are marked in boldface.

| | DE/rand/1/bin | JADE | jDE | CA | CA-ImLS | HS-CA | CADE |
|---|---|---|---|---|---|---|---|
| $f_1$ | **5.96989E−11** | 0.1323719 | **5.466E−06** | **0.0174757** | **6.397E−06** | **< 1.0E−15** | NA |
| $f_2$ | **5.61773E−14** | 0.1504679 | **1.954E−14** | **0.0080137** | 0.3276839 | **0.0083842** | NA |
| $f_3$ | **2.2575E−06** | 0.1380688 | **0.0014472** | **0.0008986** | 0.0925643 | 0.7440442 | NA |
| $f_4$ | **0.008850303** | 0.1495822 | **2.027E−13** | **4.007E−09** | 0.7172065 | 0.5585465 | NA |
| $f_5$ | > 0.9999 | > 0.9999 | > 0.9999 | **2.269E−08** | **< 1.0E−15** | **< 1.0E−15** | NA |
| $f_6$ | > 0.9999 | > 0.9999 | > 0.9999 | **4.007E−09** | > 0.9999 | > 0.9999 | NA |
| $f_7$ | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | **1.332E−15** | **4.108E−08** | NA |
| $f_8$ | **2.2575E−06** | > 0.9999 | > 0.9999 | **2.113E−12** | **< 1.0E−15** | **< 1.0E−15** | NA |
| $f_9$ | **< 1.0E−15** | > 0.9999 | > 0.9999 | > 0.9999 | > 0.9999 | > 0.9999 | NA |
| $f_{10}$ | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | **0.0240423** | **< 1.0E−15** | **0.0212238** | NA |
| $f_{11}$ | **< 1.0E−15** | > 0.9999 | > 0.9999 | 0.317684 | **< 1.0E−15** | **< 1.0E−15** | NA |
| $f_{12}$ | **3.17864E−10** | **< 1.0E−15** | **0.0168617** | **0.0770818** | **< 1.0E−15** | **< 1.0E−15** | NA |
| $f_{13}$ | **4.66294E−15** | **< 1.0E−15** | **1.291E−07** | **2.22E−16** | **< 1.0E−15** | **< 1.0E−15** | NA |
| $f_{14}$ | 0.714819715 | 0.3114947 | 0.1657671 | **< 1.0E−15** | > 0.9999 | **< 1.0E−15** | NA |
| $f_{15}$ | 0.754737609 | 0.1581784 | **0.0008094** | **0.0287825** | **4.148E−07** | 0.7239686 | NA |
| $f_{16}$ | **< 1.0E−15** | 0.2308685 | **0.0106687** | 0.7823111 | **6.725E−10** | **< 1.0E−15** | NA |
| $f_{17}$ | 0.448274101 | **< 1.0E−15** | 0.3051715 | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | NA |
| $f_{18}$ | > 0.9999 | **< 1.0E−15** | **< 1.0E−15** | > 0.9999 | > 0.9999 | **< 1.0E−15** | NA |
| $f_{19}$ | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E-15** | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | NA |
| $f_{20}$ | > 0.9999 | **< 1.0E−15** | **< 1.0E−15** | > 0.9999 | > 0.9999 | 0.9999 | NA |
| $f_{21}$ | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | NA |
| $f_{22}$ | > 0.9999 | > 0.9999 | > 0.9999 | > 0.9999 | > 0.9999 | > 0.9999 | NA |
| $f_{23}$ | 0.617192393 | 0.2932703 | **0.0160723** | **4.82E−09** | **< 1.0E−15** | **2.681E−05** | NA |
| $f_{24}$ | > 0.9999 | > 0.9999 | > 0.9999 | > 0.9999 | > 0.9999 | > 0.9999 | NA |
| $f_{25}$ | 0.540352052 | **5.013E−05** | **0.0034755** | **< 1.0E−15** | **0.0001967** | 0.8877215 | NA |
| $f_{26}$ | > 0.9999 | > 0.9999 | > 0.9999 | > 0.9999 | > 0.9999 | > 0.9999 | NA |
| $f_{27}$ | > 0.9999 | 0.0949315 | 0.2334296 | > 0.9999 | > 0.9999 | > 0.9999 | NA |
| $f_{28}$ | 0.248756704 | **0.0008371** | 0.2311795 | 0.7625979 | **< 1.0E−15** | **0.0030948** | NA |
| $f_{29}$ | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | NA |
| $f_{30}$ | 0.179437118 | **< 1.0E−15** | **< 1.0E−15** | **2.282E−10** | **3.22E−14** | **0.0108099** | NA |
| $f_{31}$ | > 0.9999 | 0.28226 | 0.1384152 | > 0.9999 | > 0.9999 | > 0.9999 | NA |
| $f_{32}$ | > 0.9999 | **1.54E−05** | **0.004017** | > 0.9999 | > 0.9999 | > 0.9999 | NA |
| $f_{33}$ | > 0.9999 | **< 1.0E−15** | **1.233E−10** | > 0.9999 | > 0.9999 | > 0.9999 | NA |
| $f_{34}$ | > 0.9999 | > 0.9999 | > 0.9999 | > 0.9999 | > 0.9999 | > 0.9999 | NA |
| $f_{35}$ | > 0.9999 | **8.947E−07** | > 0.9999 | > 0.9999 | > 0.9999 | > 0.9999 | NA |

out of the 35 and an equal performance in 9 functions. This is due to the performance of JADE on the last 5 functions. However, as can be seen Tables 7–9, CADE was able to obtain an overall better success rate, a better convergence rate, and a better reliability. This was also supported by the results shown in Table 11 using the Bonferroni–Dunn, Holm, Hochberg, and Hommel tests, where CADE obtained the most significant divergence from the results obtained by the control method, CA.

A scalability test at 100*D* was performed for the scalable functions, $f_1$–$f_{17}$. The results at this dimension are given in Table 13. From this table, and the calculated *p*-values of the *t*-test in Table 14, it is obvious that CADE was able to obtain the best results over all functions, except for function $f_5$ where it was ranked second, being outperformed by jDE alone. Table 15 presents the SR and FESS values for all the algorithms at 100D. As can be seen in this table, the canonical algorithms failed on all functions. Most of the other algorithms failed to solve many of these functions when the dimension was 100. The least SR for CADE was 96 which was obtained at $f_5$. CADE achieved the best FESS over all of these algorithms compared to the other algorithms. Both the SR and FESS metrics support the conclusion that the relative performance of the CADE algorithm was not much affected by increasing the dimensionality of the search space. Moreover, as a third metric, a ranking of these algorithms based on Friedman test is given in Table 10. It is apparent that CADE was able to obtain the highest ranking, and a *p*-value of 9.0506E-11 that shows an extremely large statistically significant result when ranked with the other algorithms using the Friedman test. Table 16 shows a comparison using the adjusted *p*-values that resulted from the Bonferroni–Dunn, Holm, Hochberg, and Hommel tests. CADE was able to obtain the most statistically significant differences when measured against the variation from a control method (CA in this case). Table 17 shows the adjusted *p*-values from Nemenyi's, Holm's, Shaffer's and Bergmann's tests. It is apparent that Nemenyi's and Holm's tests reject the null hypotheses for the functions 14-21, while Shaffer's test reject the null hypotheses 13-21. Bergmann's rejects the null hypotheses for functions 12-21. Based upon the statistical results above, it is evident that CADE obtained a competitive improvement in performance when compared to the other state of the art algorithms along with improvements in convergence and reliability. Figs. 8–10 summarize the rankings of all algorithms over all functions, at as specified in Tables 1 and 2 and 100D respectively.

**Table 7**
Experimental results of functions $f_1$–$f_{13}$ over 30$D$, averaged over 50 independent runs.

| Functions | | *F1* | *F2* | *F3* | *F4* | *F5* | *F6* | *F7* | *F8* | *F9* | *F10* | *F11* | *F12* | *F13* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DE/rand/1/bin | SR | 100 | 100 | 100 | 6 | 98 | 100 | 100 | 60 | 0 | 100 | 100 | 100 | 100 |
| | FESS | 1.1E + 05 | 1.9E + 05 | 4.2E + 05 | 3.5E + 05 | 4.4E + 05 | 4.2E + 04 | 1.5E + 05 | 3.5E + 05 | – | 1.7E + 05 | 1.1E + 05 | 9.9E + 04 | 1.1E + 05 |
| JADE w/o archive | SR | 100 | 100 | 100 | 100 | 98 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| | FESS | 2.9E + 04 | 5.2E + 04 | 9.4E + 04 | 1.7E + 05 | 1.5E + 05 | 1.1E + 04 | 2.9E + 04 | 1.3E + 05 | 1.3E + 05 | 4.5E + 04 | 3.3E + 04 | 2.7E + 04 | 3.0E + 04 |
| jDE | SR | 100 | 100 | 100 | 100 | 98 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| | FESS | 6.0E + 04 | 8.3E + 04 | 3.4E + 05 | 3.0E + 05 | 5.8E + 05 | 2.3E + 04 | 1.0E + 05 | 8.9E + 04 | 1.2E + 05 | 9.1E + 04 | 6.3E + 04 | 5.5E + 04 | 6.0E + 04 |
| CA | SR | 100 | 96 | 94 | 0 | 80 | 0 | 100 | 64 | 100 | 4 | 98 | 70 | 88 |
| | FESS | 2.4E + 04 | 1.8E + 04 | 2.0E + 05 | – | 6.0E + 05 | – | 1.4E + 05 | 3.4E + 05 | 9.4E + 04 | 3.0E + 04 | 4.3E + 04 | 2.9E + 03 | 4.2E + 04 |
| CA-ImLS | SR | 100 | 100 | 100 | 100 | 62 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| | FESS | 2.3E + 04 | 1.6E + 04 | 1.8E + 05 | 2.7E + 05 | 2.6E + 05 | 2.1E + 04 | 1.3E + 05 | 1.9E + 05 | 1.1E + 06 | 2.5E + 05 | 2.3E + 05 | 3.4E + 05 | 4.6E + 05 |
| HS-CA | SR | 100 | 100 | 100 | 98 | 53 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| | FESS | 2.4E + 04 | 1.8E + 04 | 3.4E + 04 | 1.9E + 05 | 1.9E + 05 | 1.7E + 04 | 1.1E + 05 | 2.3E + 05 | 6.8E + 05 | 1.9E + 05 | 2.0E + 05 | 1.9E + 05 | 3.7E + 05 |
| CADE | SR | 100 | 100 | 100 | 100 | 98 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| | FESS | 1.3E + 04 | 1.5E + 04 | 4.6E + 04 | 1.8E + 05 | 1.0E + 05 | 5.1E + 03 | 2.1E + 04 | 7.5E + 04 | 1.0E + 05 | 2.1E + 04 | 3.7E + 04 | 1.8E + 03 | 2.9E + 04 |

**Table 8**
Experimental results of functions $f_{14}$–$f_{26}$. Functions $f_{14}$ to $f_{17}$ were tested at 30$D$. Functions $f_{18}$ to $f_{26}$ were tested over dimensions as indicated in Tables 1 and 2, averaged over 50 independent runs.

| Functions | | $f_{14}$ | $f_{15}$ | $f_{16}$ | $f_{17}$ | $f_{18}$ | $f_{19}$ | $f_{20}$ | $f_{21}$ | $f_{22}$ | $f_{23}$ | $f_{24}$ | $f_{25}$ | $f_{26}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DE/rand/1/bin | SR | 100 | 100 | 0 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| | FESS | 1.3E + 05 | 1.9E + 05 | – | 2.9E + 05 | 1.2E + 04 | 1.1E + 04 | 1.0E + 04 | 1.3E + 04 | 1.2E + 04 | 1.3E + 04 | 1.3E + 04 | 1.3E + 04 | 1.2E + 04 |
| JADE w/o archive | SR | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 86 | 100 | 100 | 96 |
| | FESS | 7.1E + 03 | 1.4E + 04 | 7.4E + 04 | 3.4E + 05 | 1.9E + 03 | 7.0E + 02 | 9.6E + 02 | 1.2E+ 03 | 6.2E + 03 | 2.2E + 04 | 1.2E + 04 | 5.9E + 03 | 8.7E + 03 |
| jDE | SR | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| | FESS | 1.3E + 04 | 2.9E + 04 | 2.3E + 05 | 2.5E+ 05 | 2.0E + 03 | 7.3E + 02 | 1.0E + 03 | 1.2E + 03 | 5.7E + 03 | 2.2E + 04 | 1.1E + 04 | 5.0E + 03 | 7.9E + 03 |
| CA | SR | 100 | 92 | 0 | 0 | 80 | 0 | 100 | 100 | 100 | 100 | 98 | 100 | 100 |
| | FESS | 1.0E + 05 | 1.2E + 05 | – | – | 1.1E + 04 | – | 9.3E + 03 | 1.1E + 04 | 1.2E + 04 | 1.1E + 04 | 1.2E + 04 | 1.1E + 04 | 1.2E + 04 |
| CA-ImLS | SR | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| | FESS | 1.1E + 04 | 1.3E + 04 | 9.8E + 04 | 2.4E + 05 | 1.1E + 04 | 1.1E + 04 | 9.3E + 03 | 1.2E + 04 | 1.0E + 04 | 1.1E + 04 | 1.3E + 04 | 1.3E + 04 | 1.3E + 04 |
| HS-CA | SR | 100 | 100 | 100 | 90 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| | FESS | 2.4E + 04 | 1.8E + 05 | 6.7E+ 04 | 1.9E + 05 | 1.9E + 04 | 1.0E + 04 | 9.8E + 03 | 1.6E + 04 | 1.1E + 04 | 1.2E + 04 | 1.2E + 04 | 1.1E + 04 | 1.0E + 04 |
| CADE | SR | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| | FESS | 1.8E + 03 | 1.3E + 04 | 1.4E + 03 | 1.8E + 04 | 1.2E + 03 | 7.2E + 01 | 3.5E + 02 | 1.2E + 03 | 5.7E + 03 | 6.2E + 03 | 8.1E + 03 | 4.2E + 03 | 1.0E + 04 |

**Table 9**
Experimental results of functions $f_{27}$-$f_{35}$ over dimensions as indicated in Tables 1 and 2, averaged over 50 independent runs.

| Functions | | $f_{27}$ | $f_{28}$ | $f_{29}$ | $f_{30}$ | $f_{31}$ | $f_{32}$ | $f_{32}$ | $f_{33}$ | $f_{34}$ | $f_{35}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| DE/rand/1/bin | SR | 100 | 100 | 100 | 0 | 100 | 100 | 100 | 100 | 100 | 100 |
| | FESS | 1.3E + 04 | 1.9E + 04 | 4.2E + 03 | – | 4.4E + 04 | 1.2E + 04 | 1.5E + 04 | 3.5E+ 04 | 1.4E + 04 | 1.7E + 04 |
| JADE w/o archive | SR | 100 | 100 | 100 | 0 | 80 | 100 | 94 | 100 | 100 | 100 |
| | FESS | 5.0E + 03 | 8.0E + 03 | 8.2E + 02 | – | 6.3E+ 03 | 6.5E + 03 | 7.8E + 03 | 7.0E+ 03 | 6.4E + 03 | 5.0E + 03 |
| jDE | SR | 100 | 100 | 100 | 0 | 74 | 98 | 72 | 100 | 100 | 100 |
| | FESS | 4.4E + 03 | 7.0E + 03 | 9.7E + 02 | – | 6.3E + 03 | 7.5E + 03 | 6.7E + 03 | 5.0E + 03 | 5.3E + 03 | 4.4E + 03 |
| CA | SR | 100 | 96 | 99 | 0 | 90 | 90 | 100 | 99 | 100 | 100 |
| | FESS | 1.3E + 04 | 1.9E + 04 | 4.1E + 03 | – | 4.0E + 04 | 1.4E + 04 | 1.4E + 04 | 3.4E + 04 | 1.2E + 04 | 1.4E + 04 |
| CA-ImLS | SR | 100 | 100 | 100 | 100 | 62 | 100 | 100 | 100 | 100 | 100 |
| | FESS | 1.0E + 04 | 1.6E + 04 | 1.8E + 03 | 2.7E + 05 | 2.6E + 04 | 2.1E + 03 | 1.3E + 04 | 1.4E + 04 | 1.1E + 04 | 1.0E + 04 |
| HS-CA | SR | 100 | 100 | 100 | 98 | 53 | 100 | 100 | 100 | 100 | 100 |
| | FESS | 1.4E + 04 | 1.7E + 04 | 2.14E + 4 | 1.9E + 05 | 2.9E + 05 | 2.9E + 03 | 1.1E + 04 | 2.3E + 04 | 1.2E + 04 | 1.2E + 04 |
| CADE | SR | 100 | 100 | 100 | 100 | 98 | 100 | 100 | 100 | 100 | 100 |
| | FESS | 2.1E + 03 | 4.6E + 03 | 4.9E + 02 | 1.1E + 04 | 6.8E + 03 | 2.1E + 03 | 2.3E + 03 | 1.9E + 03 | 2.5E + 03 | 3.2E + 03 |

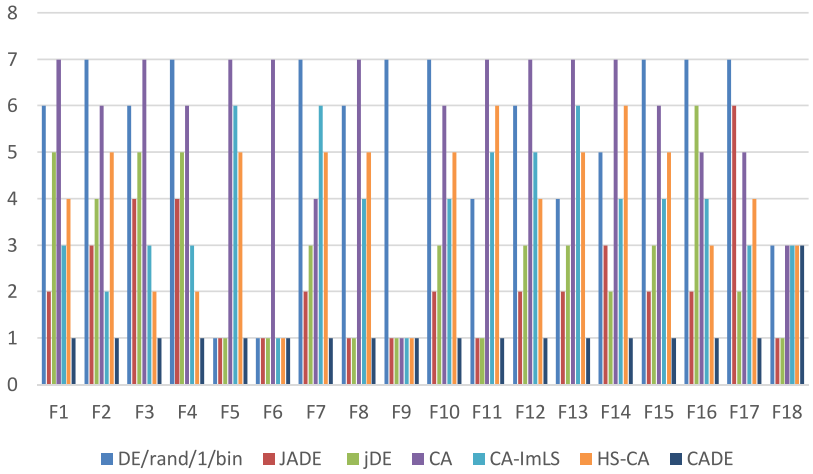


**Fig. 8.** Ranking for CADE and other state-of-the-art algorithms, in experiment 1, according to the mean value on functions (F1–F18) as specified in Tables 1 and 2.

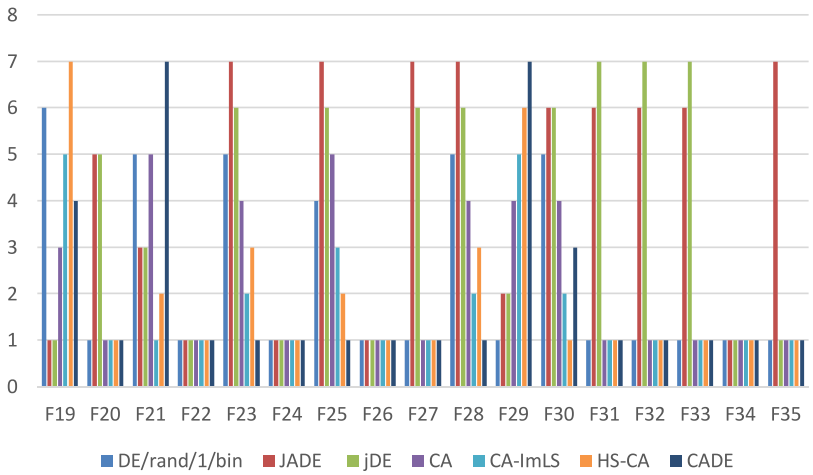


**Fig. 9.** Ranking for CADE and other state-of-the-art algorithms, in experiment 1, according to the mean value on functions (F19–F35) as specified in Tables 1 and 2.

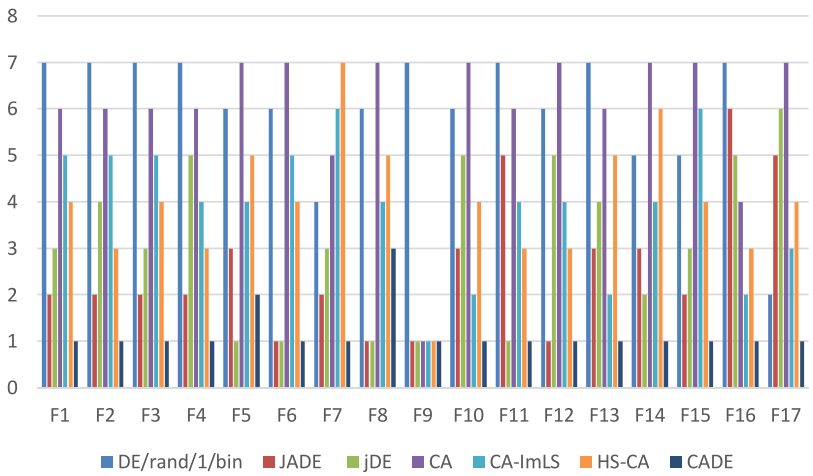


**Fig. 10.** Ranking for CADE and other state-of-the-art algorithms according to the mean value on every function at 100D.

**Table 10**
Ranking of all algorithms, achieved by the Friedman test at dimensions as specified in Tables 1 and 2, and at 100D.

| Algorithm | Ranking ($D$ as in Tables 1 and 2) | Ranking ($D = 100$) |
|---|---|---|
| DE/rand/1/bin | 2.1765 | 2.0000 |
| JADE | 5.2059 | 5.1471 |
| jDE | 4.6176 | 4.6176 |
| CA | 1.8529 | 1.8529 |
| CA-ImLS | 3.9412 | 3.9706 |
| HS-CA | 3.7059 | 3.8529 |
| **CADE** | **6.5000** | **6.5588** |
| Statistic | 58.6891 | 61.4811 |
| $p$-value | 1.3869E−10 | 9.0506E−11 |

**Table 11**
A comparison of adjusted $p$-values using the four different tests at dimensions as indicated in Tables 1 and 2. (control method: CA).

| $i$ | Hypothesis | Unadjusted $p$ | $p_{\text{Bonf}}$ | $p_{\text{Holm}}$ | $p_{\text{Hock}}$ | $p_{\text{Homm}}$ |
|---|---|---|---|---|---|---|
| 1 | CADE | **3.57E−10** | **2.14E−09** | **2.14E−09** | **2.14E−09** | **2.14E−09** |
| 2 | JADE | **6.04E−06** | **3.62E−05** | **3.02E−05** | **3.02E−05** | **3.02E−05** |
| 3 | jDE | **1.91E−04** | **0.001143** | **7.62E−04** | **7.62E−04** | **7.62E−04** |
| 4 | CA-ImLS | **0.004828** | **0.028968** | **0.014484** | **0.014484** | **0.014484** |
| 5 | HS-CA | **0.012394** | **0.074361** | **0.024787** | **0.024787** | **0.024787** |
| 6 | DE/rand/1/bin | 0.662375 | 3.974251 | 0.662375 | 0.662375 | 0.662375 |

**Table 12**
Adjusted $p$-values at dimensions as specified in Tables 1 and 2.

| i | Hypothesis | Unadjusted $p$ | $P_{Neme}$ | $p_{Holm}$ | $P_{Shaf}$ | $P_{Berg}$ |
|---|---|---|---|---|---|---|
| 21 | CA vs. CADE | **3.57E−10** | **7.50E−09** | **7.50E−09** | **7.50E−09** | **7.50E−09** |
| 20 | DE/rand/1/bin vs. CADE | **5.38E−09** | **1.13E−07** | **1.08E−07** | **8.07E−08** | **8.07E−08** |
| 19 | JADE vs. CA | **6.04E−06** | **1.27E−04** | **1.15E−04** | **9.05E−05** | **9.05E−05** |
| 18 | DE/rand/1/bin vs. JADE | **4.34E−05** | **9.12E−04** | **7.81E−04** | **6.51E−04** | **4.34E−04** |
| 17 | HS-CA vs. CADE | **1.63E−04** | **0.003415** | **0.002765** | **0.002439** | **0.001789** |
| 16 | jDE vs. CA | **1.91E−04** | **0.004001** | **0.003048** | **0.002858** | **0.002096** |
| 15 | CA-ImLS vs. CADE | **5.54E−04** | **0.011625** | **0.008304** | **0.008304** | **0.004982** |
| 14 | DE/rand/1/bin vs. jDE | **9.86E−04** | **0.020697** | **0.013798** | **0.010841** | **0.006899** |
| 13 | CA vs. CA-ImLS | **0.004828** | 0.101389 | 0.062764 | 0.053108 | 0.043452 |
| 12 | jDE vs. CADE | **0.011072** | 0.232505 | 0.13286 | 0.121788 | 0.06643 |
| 11 | CA vs. HS-CA | **0.012394** | 0.260265 | 0.136329 | 0.136329 | 0.074361 |
| 10 | DE/rand/1/bin vs. CA-ImLS | **0.017235** | 0.361938 | 0.172352 | 0.172352 | 0.074361 |
| 9 | DE/rand/1/bin vs. HS-CA | **0.039008** | 0.819175 | 0.351075 | 0.351075 | 0.156033 |
| 8 | JADE vs. HA-CA | **0.042929** | 0.901499 | 0.351075 | 0.351075 | 0.3005 |
| 7 | JADE vs. CADE | 0.080716 | 1.695041 | 0.565014 | 0.565014 | 0.403581 |
| 6 | JADE vs. CA-ImLS | 0.08785 | 1.844845 | 0.565014 | 0.565014 | 0.403581 |
| 5 | jDE vs. HS-CA | 0.218502 | 4.588547 | 1.092511 | 1.092511 | 0.874009 |
| 4 | jDE vs. CA-ImLS | 0.36126 | 7.58646 | 1.44504 | 1.44504 | 0.874009 |
| 3 | JADE vs. jDE | 0.427263 | 8.97252 | 1.44504 | 1.44504 | 1.281789 |
| 2 | DE/rand/1/bin vs. CA | 0.662375 | 13.90988 | 1.44504 | 1.44504 | 1.32475 |
| 1 | CA-ImLS vs. HS-CA | 0.750824 | 15.76729 | 1.44504 | 1.44504 | 1.32475 |

*4.1.4. Sensitivity analysis*

In this section, the performance of CADE will be assessed with respect to its most sensitive parameters. This is important in order to help improve its performance and save function evaluations. The parameters to be tuned in CADE include the scaling parameter ($F$), the crossover rate ($CR$), the ratio of accepted individuals to the belief space in CA ($p_{\%\text{ind}}$), and the population size ($NP$). These tests relied on varying the value of one parameter while fixing the values of the other parameters as discussed in earlier sections. Tables 18–20 show the results of these sensitivity tests over the 50 independent runs of CADE. For the sake of selecting representative functions, two functions of each of the first 3 categories were chosen while three functions were chosen from the last category because they contain a large set of local minima as reported in [7].

As can be seen from Table 18, for parameter ($F$), the best setup was at 0.5 over all the selected functions. The same is true about ($CR$). The best ratio for acceptance of individuals into the belief space before advancing to the next generation occurs at $p_{\%\text{ind}} = 0.25$. When $p_{\%\text{ind}}$ was set at 0.2 for $f_6$ CADE reached the same optimal value as when set at 0.25. However, the average performance over all the functions exhibits best scores when the parameter was set at 0.25. The best $NP$ size over these selected functions was 40 as can be seen in Table 19. An $NP$ size of 50 was a better selection for CADE over

**Table 13**
Statistical results of all the algorithms on the scalable benchmark functions over 100*D* as indicated in Tables 1 and 2, Averaged over 50 independent runs.

| | DE/rand/1/bin | JADE | jDE | CA | CA-ImLS | HS-CA | CADE |
|---|---|---|---|---|---|---|---|
| $f_1$ | 3.50E+01 (9.60E+00) | 5.40E−67(1.60E−66) | 5.00E−15(1.70E−15) | 3.20E−01(2.20E−01) | 5.24E−08(3.11E−07) | 2.29E−12(3.53E−11) | **0.00E+00(0.00E+00)** |
| $f_2$ | 2.30E+00(5.60E−01) | 9.20E−51(2.20E−50) | 4.10E−15(1.10E−15) | 3.10E−04(2.70E−04) | 6.82E−06(3.98E−05) | 1.01E−32(1.98E−31) | **0.00E+00(0.00E+00)** |
| $f_3$ | 2.10E+05(3.10E+04) | 2.20E−37(2.50E−37) | 5.40E−37(2.70E−02) | 5.30E+04(4.60E+03) | 1.27E+01(4.01E+00) | 3.47E−05(3.49E−06) | **0.00E+00(0.00E+00)** |
| $f_4$ | 9.30E+01(2.80E+00) | 3.20E−71(8.30E−71) | 3.10E−09(5.90E−10) | 8.30E−011.60E−01) | 6.21E−17(8.32E−18) | 5.71E−22(3.52E−21) | **0.00E+000.00E+00)** |
| $f_5$ | 4.30E+01(1.20E+01) | 4.00E−01(1.20E+00) | **9.10E−03(2.50E−02)** | 8.60E+01(2.50E+01) | 2.46E+01(7.65E+00) | 3.31E+01(1.76E+01) | 3.50E−01(1.10E+00) |
| $f_6$ | 3.20E+02(6.30E+01) | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | 5.40E+02(7.30 +01) | 9.91E+00(2.11E+00) | 7.90E+00(1.14E+00) | **0.00E+00(0.00E+00)** |
| $f_7$ | 2.90E−02(5.70E−03) | 7.80E−04(1.40E−04) | 8.10E−03(9.00E−04) | 3.80E−02(5.90E−03) | 2.96E−01(6.03E−02) | 1.52E+00(3.32E−01) | **5.60E−04(1.70E−04)** |
| $f_8$ | 3.00E+04(9.00E+02) | **1.10E−10(0.00E+00)** | **1.10E−10(0.00E+00)** | 3.10E+04(4.60E+02) | 8.12E+03(3.68E+02) | 8.52E+03(5.14E+02) | 5.00E+03(4.30E+02) |
| $f_9$ | 8.10E+02(1.80E+01) | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** |
| $f_{10}$ | 1.30E−01(2.40E−02) | 8.00E−15(0.00E+00) | 9.90E−14(2.00E−14) | 1.50E−01(3.40E−02) | 2.87E−19(1.66E−18) | 2.63E−14(1.20E−11) | **0.00E+00(0.00E+00)** |
| $f_{11}$ | 2.00E−01(5.80E−02) | 1.50E−04(1.00E−03) | **0.00E+00(0.00E+00)** | 2.40E−03(4.30E−03) | 9.06E−10(2.15E−10) | 5.29E−13(4.28E−15) | **0.00E+00(0.00E+00)** |
| $f_{12}$ | 1.70E+00(1.50E+00) | **4.70E−33 (6.80E−49)** | 1.70E−25 (7.70E−26) | 4.20E+02(1.80E+03) | 2.36E−26 (1.92E−27) | 5.72E−29(7.67E−28) | **4.70E−33(2.70E−48)** |
| $f_{13}$ | 5.10E+00(3.20E+00) | 1.40E−32(1.10E−47) | 2.10E−24(1.50E−24) | 5.00E−01(7.40E−01) | 1.92E−33(6.88E−38) | 8.61E−21(3.26E−20) | **0.00E+00(0.00E+00)** |
| $f_{14}$ | 4.90E−14(3.47E−13) | 2.23E−71(1.55E−70) | 1.83E−72(1.30E−71) | 3.97E−01(1.04E−01) | 4.80E−16(3.22E−17) | 5.84E−13(1.71E−13) | **0.00E+00(0.00E+00)** |
| $f_{15}$ | 1.09E−25(1.62E−25) | 5.91E−143(3.10E−142) | 4.59E−59(1.74E−58) | 6.54E−02(3.13E−02) | 3.91E−16(4.38E−15) | 7.20E−27(5.29E−26) | **0.00E+00(0.00E+00)** |
| $f_{16}$ | 3.65E+02(5.31E+01) | 3.95E+01(9.25E+01) | 9.90E+00(1.42E+01) | 1.79E+00(3.46E−01) | 8.39E−11(1.66E−11) | 9.35E−09(2.86E−09) | **7.16E−30(5.81E−29)** |
| $f_{17}$ | 6.90E−14(7.92E−14) | 9.18E−10(6.21E−09) | 6.25E−08(1.07E−07) | 2.26E−01(9.32E−02) | 6.81E−13(4.29E−13) | 3.17E−11(5.41E−12) | **3.92E−28(2.76E−29)** |

**Table 14**
*P*-values calculated for the *t*-test (significance level 0.05) corresponding to the results of Table 13.

| | DE/rand/1/bin | JADE | jDE | CA | CA-ImLS | HS-CA | CADE |
|---|---|---|---|---|---|---|---|
| *F1* | **< 1.0E−15** | **0.019741** | **< 1.0E−15** | **5.62E−14** | 0.234661 | 0.645213 | NA |
| *F2* | **< 1.0E−15** | **0.004398** | **< 1.0E−15** | **9.48E−11** | 0.226905 | 0.717213 | NA |
| *F3* | **< 1.0E−15** | 8.51E−08 | > 0.9999 | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | NA |
| *F4* | **< 1.0E−15** | **0.008252** | < 1.0E−15 | **< 1.0E−15** | **< 1.0E−15** | 0.252294 | NA |
| *F5* | **< 1.0E−15** | **0.02421** | NA | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | 0.031614 |
| *F6* | **< 1.0E−15** | > 0.9999 | > 0.9999 | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | NA |
| *F7* | **< 1.0E−15** | **4.11E−09** | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | NA |
| *F8* | **< 1.0E−15** | > 0.9999 | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | NA |
| *F9* | **< 1.0E−15** | > 0.9999 | > 0.9999 | > 0.9999 | > 0.9999 | > 0.9999 | NA |
| *F10* | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | 0.222833 | 0.987576 | NA |
| *F11* | **< 1.0E−15** | 0.289321 | > 0.9999 | **0.000223** | **< 1.0E−15** | **< 1.0E−15** | NA |
| *F12* | **1.37E−10** | > 0.9999 | **< 1.0E−15** | 0.102028 | **< 1.0E−15** | 0.596761 | NA |
| *F13* | **2.22E−15** | **< 1.0E−15** | **2.03E−13** | **1.4E−05** | **< 1.0E−15** | 0.065212 | NA |
| *F14* | 0.318196 | 0.309257 | 0.319694 | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | NA |
| *F15* | **1.5E−05** | 0.17959 | 0.065525 | **< 1.0E−15** | 0.52676 | 0.335829 | NA |
| *F16* | **< 1.0E−15** | **0.003691** | **8.32E−06** | **< 1.0E−15** | **< 1.0E−15** | **< 1.0E−15** | NA |
| *F17* | **1.06E−07** | 0.296288 | **0.000123** | **< 1.0E−15** | **2.66E−15** | **< 1.0E−15** | NA |

**Table 15**
Experimental results of high dimensional functions, $f_1$–$f_{17}$, over 100*D*, Averaged over 50 independent runs.

| Algorithms | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Fun | DE/rand/1/bin | | JADE/archive | | jDE | | CA | | CA-ImLS | | HS-CA | | CADE | |
| | SR | FESS | SR | FESS | SR | FESS | SR | FESS | SR | FESS | SR | FESS | SR | FESS |
| $f_1$ | 0 | – | 100 | 1.6E+05 | 100 | 5.5E+05 | 0 | – | 100 | 2.2E+05 | 44 | 6.9E+05 | 100 | 3.7E+04 |
| $f_2$ | 0 | – | 100 | 2.7E+05 | 100 | 7.6EE+05 | 0 | – | 100 | 6.4E+05 | 0 | – | 100 | 3.4E+04 |
| $f_3$ | 0 | – | 100 | 9.6E+05 | 0 | – | 0 | – | 0 | – | 100 | 1.9E+06 | 100 | 7.0E+05 |
| $f_4$ | 0 | – | 100 | 7.7E+05 | 100 | 5.7E+06 | 0 | – | 100 | 4.7E+06 | 0 | – | 100 | 2.7E+05 |
| $f_5$ | 0 | – | 90 | 1.5E+06 | 2 | 7.7E+06 | 0 | – | 0 | – | 55 | 1.1E+06 | 96 | 1.2E+06 |
| $f_6$ | 0 | – | 100 | 6.2E+04 | 100 | 2.2E+05 | 0 | – | 100 | 8.3E+04 | 100 | 3.2E+05 | 100 | 2.8E+04 |
| $f_7$ | 0 | – | 100 | 2.0E+05 | 96 | 2.0E+06 | 0 | – | 96 | 1.8E+05 | 67 | 5.9E+05 | 100 | 1.7E+04 |
| $f_8$ | 0 | – | 100 | 1.4E+06 | 100 | 9.5E+05 | 0 | – | 100 | 9.1E+05 | 100 | 1.4E+06 | 100 | 9.1E+05 |
| $f_9$ | 0 | – | 100 | 1.5E+06 | 100 | 1.4E+06 | 0 | – | 100 | 1.4E+06 | 100 | 1.1E+06 | 100 | 7.5E+05 |
| $f_{10}$ | 0 | – | 100 | 2.4E+05 | 100 | 8.0E+05 | 0 | – | 100 | 1.4E+06 | 0 | – | 100 | 5.2E+04 |
| $f_{11}$ | 0 | – | 98 | 1.7E+05 | 100 | 5.4E+05 | 0 | – | 100 | 1.5E+05 | 100 | 1.7E+05 | 100 | 5.3E+04 |
| $f_{12}$ | 0 | – | 100 | 1.4E+05 | 100 | 5.7E+05 | 0 | – | 100 | 5.2E+05 | 100 | 4.9E+05 | 100 | 1.4E+04 |
| $f_{13}$ | 0 | – | 100 | 1.6E+05 | 100 | 5.8E+05 | 0 | – | 100 | 1.8E+05 | 100 | 1.4E+05 | 100 | 8.6E+04 |
| $f_{14}$ | 0 | – | 100 | 8.3E+03 | 100 | 2.6E+04 | 0 | – | 100 | 8.0E+03 | 100 | 1.0E+04 | 100 | 3.2E+03 |
| $f_{15}$ | 0 | – | 100 | 3.5E+04 | 100 | 7.8E+04 | 0 | – | 100 | 3.7E+04 | 100 | 3.4E+04 | 100 | 1.0E+04 |
| $f_{16}$ | 0 | – | 0 | – | 0 | – | 0 | – | 0 | – | 0 | – | 100 | 3.2E+04 |
| $f_{17}$ | 0 | – | 100 | 1.3E+05 | 26 | 1.7E+05 | 0 | – | 66 | 1.0E+05 | 29 | 1.3E+05 | 100 | 9.2E+04 |

**Table 16**
A comparison of adjusted *p*-values using the four different tests at 100D, for the scalable functions. (control method: CA).

| i | Hypothesis | Unadjusted *p* | $P_{Bonf}$ | $p_{Holm}$ | $P_{Hock}$ | $p_{Homm}$ |
|---|---|---|---|---|---|---|
| 1 | CADE | **2.14E−10** | **1.28E−09** | **1.28E−09** | **1.28E−09** | **1.28E−09** |
| 2 | JADE | **8.76E−06** | **5.26E−05** | **4.38E−05** | **4.38E−05** | **4.38E−05** |
| 3 | jDE | **1.91E−04** | **0.001143** | **7.62E−04** | **7.62E−04** | **7.62E−04** |
| 4 | CA-ImLS | **0.004263** | **0.025581** | **0.01279** | **0.01279** | **0.010426** |
| 5 | HS-CA | **0.006951** | **0.041703** | **0.013901** | **0.013901** | **0.013901** |
| 6 | DE/rand/1/bin | 0.842677 | 5.056059 | 0.842677 | 0.842677 | 0.842677 |

$f_9$. Though, the average performance over all the functions was observed at $NP = 40$. The summary of the results in Tables 18 and 19 are also presented in Fig. 11.

The study considered the choice of the best DE configuration for this combination of benchmark functions. As can be seen in Table 20, among the choices used from DE/current-to-pbest/1/bin, DE/rand/1/exp, DE/best/1, DE/rand/1/bin and DE/current-to-best, the experiments show that the best results were obtained over most of the functions when DE/rand/1/bin is selected (Table 21).

**Table 17**
Adjusted $p$-values when $D = 100$.

| i | Hypothesis | Unadjusted $p$ | $P_{Neme}$ | $p_{Holm}$ | $P_{Shaf}$ | $P_{Berg}$ |
|---|---|---|---|---|---|---|
| 21 | CA vs. CADE | 2.14E−10 | 4.49E−09 | 4.49E−09 | 4.49E−09 | 4.49E−09 |
| 20 | DE/rand/1/bin vs. CADE | 7.62E−10 | 1.60E−08 | 1.52E−08 | 1.14E−08 | 1.14E-08 |
| 19 | JADE vs. CA | 8.76E−06 | 1.84E−04 | 1.66E−04 | 1.31E-04 | 1.31E−04 |
| 18 | DE/rand/1/bin vs. JADE | 2.16E−05 | 4.54E−04 | 3.89E−04 | 3.25E−04 | 2.16E−04 |
| 17 | jDE vs. CA | 1.91E−04 | 0.004001 | 0.003239 | 0.002858 | 0.002096 |
| 16 | HS-CA vs. CADE | 2.60E−04 | 0.005467 | 0.004165 | 0.003905 | 0.002864 |
| 15 | DE/rand/1/bin vs. jDE | 4.11E−04 | 0.008635 | 0.006168 | 0.006168 | 0.002878 |
| 14 | CA-ImLS vs. CADE | 4.77E−04 | 0.010027 | 0.006685 | 0.006168 | 0.004297 |
| 13 | CA vs. CA-ImLS | 0.004263 | 0.089532 | 0.055425 | 0.046898 | 0.038371 |
| 12 | CA vs. HS-CA | 0.006951 | 0.145961 | 0.083406 | 0.076456 | 0.041703 |
| 11 | DE/rand/1/bin vs. CA-ImLS | 0.007825 | 0.164333 | 0.086079 | 0.086079 | 0.046952 |
| 10 | jDE vs. CADE | 0.008798 | 0.184751 | 0.087977 | 0.087977 | 0.046952 |
| 9 | DE/rand/1/bin vs. HS-CA | 0.012394 | 0.260265 | 0.111542 | 0.111542 | 0.049574 |
| 8 | JADE vs. CADE | 0.056738 | 1.191503 | 0.453906 | 0.397168 | 0.283691 |
| 7 | JADE vs. HS-CA | 0.080716 | 1.695041 | 0.565014 | 0.565014 | 0.565014 |
| 6 | JADE vs. CA-ImLS | 0.112339 | 2.359113 | 0.674032 | 0.674032 | 0.565014 |
| 5 | jDE vs. HS-CA | 0.302049 | 6.343028 | 1.510245 | 1.510245 | 1.208196 |
| 4 | jDE vs. CA-ImLS | 0.382515 | 8.032805 | 1.530058 | 1.530058 | 1.208196 |
| 3 | JADE cs. jDE | 0.474921 | 9.973334 | 1.530058 | 1.530058 | 1.424762 |
| 2 | DE/rand/1/bin vs. CA | 0.842677 | 17.69621 | 1.685353 | 1.685353 | 1.685353 |
| 1 | CA-ImLS vs. HS-CA | 0.873845 | 18.35074 | 1.685353 | 1.685353 | 1.685353 |

**Table 18**
Sensitivity analysis with respect to scale factor ($F$), crossover rate ($CR$), percentage of selected experiences in the belief space of CA ($p_{\%ind}$), and population size ($NP$) over 50 independent runs for dimensions as indicated in Tables 1 and 2 for all selected functions from experiment 1, and functions $f_{40}$, $f_{45}$ and $f_{50}$ having a dimensionality used for this study as 50$D$.

| Optimal results over different parameter values | | | | | |
|---|---|---|---|---|---|
| $F$ | 0.1 | 0.3 | 0.5 | 0.7 | 0.9 |
| $f_1$ | 4.17E−19(1.01E−18) | 2.79E−34(3.56E−35) | **0.00E + 00(0.00E + 00)** | 4.88E−42(2.19E−41) | 3.64E−35(5.08E−39) |
| $f_2$ | 6.52E−22(5.31E−22) | 7.90E−46(2.47E−46) | **0.00E + 00(0.00E + 00)** | 5.19E−39(2.62E−40) | 4.68E−32(5.63E−32) |
| $f_6$ | 3.23E−31(5.04E−30) | 3.77E−56(6.82E−58) | **0.00E + 00(0.00E + 00)** | 1.66E−49(3.28E−49) | 7.51E−32(4.59E−32) |
| $f_8$ | 9.40E−09(5.35E−11) | 1.73E−29(6.83E−31) | **0.00E + 00(0.00E + 00)** | 0.00E + 00(0.00E + 00) | 2.89E−37(4.77E−37) |
| $f_9$ | 8.46E-28(2.71E−29) | 0.00E + 00(0.00E + 00) | **0.00E + 00(0.00E + 00)** | 0.00E + 00(0.00E + 00) | 2.51E−30(3.72E−30) |
| $f_{18}$ | 9.36E−01(1.46E−05) | 9.68E−01(4.72E−12) | **1.00E + 00(2.18E−23)** | 9.81E−01(4.66E−10) | 9.41E−01(3.79E−11) |
| $f_{19}$ | 2.99E−03(2.49E−04) | 3.07E−03(1.88E−09) | **3.07E−03(1.10E−20)** | 3.07E−03(5.91E−05) | 3.01E−03(2.72E−09) |
| $f_{40}$ | 9.32E + 02(3.62E + 02) | 6.39E + 02(2.90E + + 02) | **5.90E + 02(2.78E + 02)** | 6.43E + 02(2.86E + 02) | 8.13E + 02(3.24E + 02) |
| $f_{45}$ | 4.09E + 021.03E + 01 | 3.91E + 02(4.69E + 00) | **3.28E + 02(2.63E−15)** | 3.78E + 02(3.15E−01) | 4.19E + 02(5.27E + 00) |
| $f_{50}$ | 4.18E + 02(2.17E + 01) | 4.02E + 02(2.30E + 01) | **3.61E + 02(1.87E + 01)** | 4.10E + 02(1.85E + 01) | 4.29E + 02(1.96E + 01) |
| $CR$ | 0.1 | 0.3 | 0.5 | 0.7 | 0.9 |
| $f_1$ | 4.86E−05(5.46E−05) | 7.59E−09(1.59E−09) | **0.00E + 00(0.00E + 00)** | 3.77E−11(2.82E−12) | 1.94E−08(3.81E−09) |
| $f_2$ | 6.67E−02(3.16E−02) | 4.09E−11(5.73E−11) | **0.00E + 00(0.00E + 00)** | 1.00E−10(2.97E−10) | 6.45E−08(5.58E−09) |
| $f_6$ | 8.21E−09(2.78E−11) | 1.80E−22(1.02E−22) | **0.00E + 00(0.00E + 00)** | 3.93E−34(7.88E−33) | 4.48E−29(2.67E−29) |
| $f_8$ | 7.73E−08(1.14E−08) | 3.54E−19(2.13E−20) | **0.00E + 00(0.00E + 00)** | 3.21E−23(1.01E−24) | 1.87E−18(6.58E−18) |
| $f_9$ | 3.77E−12(2.08E−12) | 1.14E−18(4.59E−20) | **0.00E + 00(0.00E + 00)** | 6.72E−19(1.24E−19) | 2.99E−14(4.70E−14) |
| $f_{18}$ | 9.01E−01(1.35E−04) | 9.57E−01(9.56E−11) | **1.00E + 00(2.18E−23)** | 9.61E−01(1.53E−12) | 9.28E−01(7.28E−06) |
| $f_{19}$ | 2.39E−03(2.76E−05) | 2.68E−03(5.91E−04) | **3.07E−03(1.10E−20)** | 2.99E−03(4.72E−09) | 2.91E−03(3.42E−06) |
| $f_{40}$ | 6.64E + 02(2.90E + 02) | 6.16E + 02(2.83E + 02) | 6.25E + 02(2.88E + 02) | **5.90E + 02(2.78E + 02)** | 7.88E + 02(3.01E + 02) |
| $f_{45}$ | 4.26E + 02(5.86E + 00) | 3.91E + 02(1.60E + 00) | **3.28E + 02(2.63E−15)** | 4.78E + 02(1.08E + 01) | 5.27E + 02(2.79E + 01) |
| $f_{50}$ | 5.84E + 02(2.63E + 01) | 4.52E + 02(2.42E + 01) | **3.61E + 02(1.87E + 01)** | 4.62E + 02(2.14E + 01) | 5.37E + 02(2.91E + 01) |

### 4.2. Numerical benchmarks- experiment 2

The CADE algorithm is also tested using the benchmark functions from the special session and competition on single objective real-parameter numerical optimization held under the IEEE CEC 2014 [25]. Such benchmark functions span a various set of problem features and characteristics, including ruggedness, noise in fitness, multimodality, ill-conditioning, interdependence, and non-separability. 15 of the toughest problems have been chosen to cover all of the four categories: simple unimodal ($f_1$), multimodal ($f_4$, $f_5$ and $f_6$), hybrid ($f_{17}$, $f_{18}$, $f_{19}$, $f_{20}$ and $f_{21}$) and composition ($f_{23}$, $f_{24}$, $f_{25}$, $f_{26}$, $f_{27}$ and $f_{28}$).

#### 4.2.1. Compared algorithms and parametric setup

In this section, the performance of CADE is compared with eight other state-of-the-art contestants in the competition using experimental and previously reported results taken from the literature. For all of the algorithms under comparison the recommended parametric setup for the benchmark suite are selected using guidelines from the literature. The rest of

**Table 19**
Sensitivity analysis with respect to percentage of selected experiences in the belief space of CA ($p_{\%ind}$), and population size ($NP$) over 50 independent runs for dimensions as indicated in Tables 1 and 2 for all selected functions from experiment 1, and functions $f_{40}$, $f_{45}$ and $f_{50}$ having a dimensionality used for this study as 50$D$.

| $p_{\%ind}$ | 15% | 20% | 25% | 30% | 35% |
|---|---|---|---|---|---|
| $f_1$ | 1.02E−08(3.64E−08) | 4.92E−10(2.79E−11) | **0.00E+00(0.00E+00)** | 5.82E−11(1.48E−11) | 2.67E−09(3.55E−09) |
| $f_2$ | 8.27E−04(1.72E−04) | 4.83E−11(3.67E−13) | **0.00E+00(0.00E+00)** | 6.37E−15(5.77E−10) | 8.91E−07(4.88E−07) |
| $f_6$ | 3.28E−09(2.71E−09) | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | 3.62E−20(2.22E−10) | 4.88E−15(6.29E−15) |
| $f_8$ | 4.19E−11(3.76E−12) | 6.91E−23(1.53E−23) | **0.00E+00(0.00E+00)** | 5.31E−30(5.35E−30) | 7.33E−19(5.28E−19) |
| $f_9$ | 5.21E−17(4.81E−17) | 3.55E−19(1.78E−20) | **0.00E+00(0.00E+00)** | 7.29E−24(5.00E−24) | 8.63E−23(4.03E−23) |
| $f_{18}$ | 8.87E−01(9.27E−15) | 9.56E−01(3.81E−10) | **1.00E+00(2.18E−23)** | 9.79E−01(6.94E−10) | 9.01E−01(4.87E−11) |
| $f_{19}$ | 2.28E−03(2.53E−16) | 3.07E−03(4.26E−09) | **3.07E−03(1.10E−20)** | 3.07E−03(3.62E−13) | 2.92E−03(4.33E−08) |
| $f_{40}$ | 7.25E+02(3.07E+02) | 6.27E+02(2.95E+02) | **5.90E+02(2.78E+02)** | 6.33E+02(2.87E+02) | 6.91E+02(3.00E+02) |
| $f_{45}$ | 4.86E+02(1.06E+01) | 4.29E+02(2.78E+00) | **3.28E+02(2.63E−15)** | 4.18E+02(1.66E+00) | 4.93E+02(3.78E+00) |
| $f_{50}$ | 4.74E+02(3.45E+01) | 4.36E+02(2.28E+01) | 4.15E+02(2.59E+01) | **3.61E+02(1.87E+01)** | 4.64E+02(3.26E+01) |
| $NP$ | 10 | 30 | 40 | 50 | 70 |
| $f_1$ | 5.36E−05(3.29E−06) | 4.26E−13(6.28E−13) | **0.00E+00(0.00E+00)** | 6.27E−37(6.79E−38) | 8.71E−353.19E−35 |
| $f_2$ | 5.71E−07(9.03E−07) | 2.12E−19(8.48E−21) | **0.00E+00(0.00E+00)** | 5.09E−41(8.30E−40) | 7.81E−36(4.29E−36) |
| $f_6$ | 8.37E−06(4.61E−05) | 9.78E−21(3.28E−22) | **0.00E+00(0.00E+00)** | 6.29E−46(1.71E−46) | 3.89E−40(1.52E−39) |
| $f_8$ | 4.72E−05(4.42E−05) | 3.62E−26(1.18E−26) | **0.00E+00(0.00E+00)** | **0.00E+00(0.00E+00)** | 5.12E−30(1.63E−31) |
| $f_9$ | 7.12E−09(4.43E−09) | 8.19E−30(5.46E−30) | 3.79E−61(5.28E−61) | **0.00E+00(0.00E+00)** | 2.88E−49(5.66E−48) |
| $f_{18}$ | 8.39E−08(3.21E−08) | 9.79E−01(2.91E−31) | **1.00E+00(2.18E−23)** | 1.00E+00(6.71E−17) | 9.80E−01(4.82E−31) |
| $f_{19}$ | 2.39E−03(4.27E−12) | 3.07E−03(4.22E−10) | **3.07E−03(1.10E−20)** | 3.07E−03(7.26E−13) | 2.99E−03(5.22E−19) |
| $f_{40}$ | 6.88E+02(3.04E+02) | 6.27E+02(2.88E+02) | **5.90E+02(2.78E+02)** | 6.32E+02(2.91E+02) | 7.00E+02(2.88E+02) |
| $f_{45}$ | 4.89E+02(4.65E+00) | 3.81E+02(2.61E+00) | **3.28E+02(2.63E−15)** | 4.10E+02(1.08E+00) | 4.70E+02(3.52E+00) |
| $f_{50}$ | 4.80E+02(3.7E+01) | 4.17E+02(2.51E+01) | **3.61E+02(1.87E+01)** | 4.23E+02(2.13E+01) | 4.83E+02(3.19E+01) |

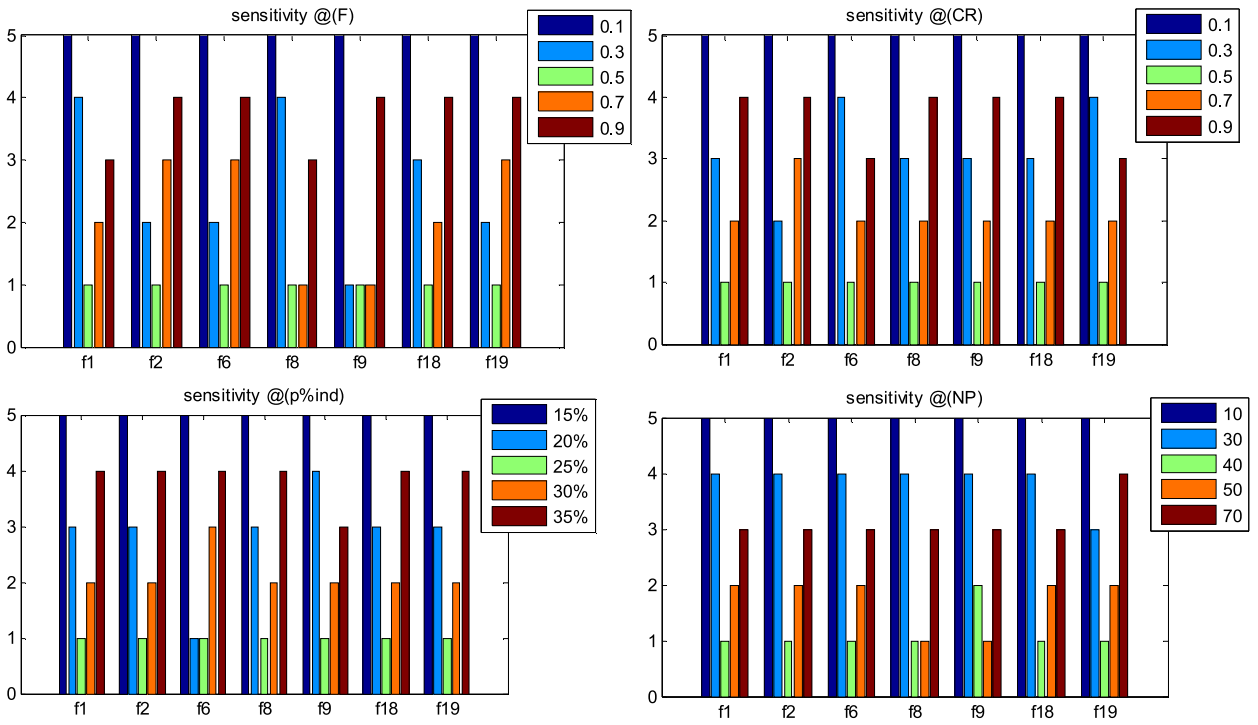


**Fig. 11.** Summary of the sensitivity tests for main parameters in CADE.

the parameters are adaptively set during the search. For CADE, the same parameter settings for category the first set of benchmark experiments have been used here.

1 CoDE with $NP=30$ [41]
2 dynNP-jDE [6] with a number of different population sizes ($pmax$) = 4 and $NP=200$.
3 EPSDE with $NP=50$ [49]
4 JADE with $NP=100$ [48]
5 SHADE with the archive rate = 2.0 and $NP=100$ [39]

**Table 20**
Sensitivity analysis with respect to the used type of DE over 50 independent runs for dimensions as indicated in Tables 1 and 2 for all selected functions from experiment 1, and functions $f_{40}$, $f_{45}$ and $f_{50}$ having a dimensionality used for this study as 50*D*.

| *DE-type* | DE/current-to-pbest | Rand/1/bin | Best/1 | DE/current-to-best | Rand/1/exp |
|---|---|---|---|---|---|
| $f_1$ | **0.00E + 00(0.00E + 00)** | 2.63E−18(4.21E−19) | 7.11E−09(3.46E−08) | 5.13E−19(1.89E−20) | 8.42E−19(9.46E−19) |
| $f_2$ | **0.00E + 00(0.00E + 00)** | 5.32E−19(1.70E−19) | 3.41E−10(9.26E−11) | 3.08E−17(1.67E−19) | 8.16E−17(9.83E−17) |
| $f_6$ | **0.00E + 00(0.00E + 00)** | 5.20E−19(7.92E−19) | 6.12E−05(2.15E−06) | 3.00E−17(7.11E−18) | 6.33E−16(5.35E−16) |
| $f_8$ | **0.00E + 00(0.00E + 00)** | **0.00E + 00(0.00E + 00)** | 6.89E−07(3.66E−08) | **0.00E + 00(0.00E + 00)** | 5.72E−22(1.40E−22) |
| $f_9$ | 0.00E + 00(0.00E + 00) | 4.67E−25(2.11E−27) | 1.09E−09(4.73E−10) | **6.24E−21(0.00E + 00)** | 3.99E−203.79E−21 |
| $f_{18}$ | **1.00E + 00(2.18E−23)** | 9.33E−01(9.42E−10) | 6.08E−01(1.73E−03) | 1.00E + 00(2.15E−18) | 9.75E−01(2.59E−03) |
| $f_{19}$ | **3.07E−03(1.10E−20)** | 3.07E−03(3.61E−17) | 3.07E−03(5.72E−03) | 2.99E−03(8.91E−15) | 3.07E−03(1.65E−16) |
| $f_{40}$ | **5.90E + 02(2.78E + 02)** | 6.52E + 02(3.14E + 02) | 9.15E + 02(3.63E + 02) | 6.71E + 02(3.09E + 02) | 6.99E + 02(2.90E + 02) |
| $f_{45}$ | **3.28E + 02(2.63E−15)** | 3.82E + 02(1.96E + 00) | 4.66E + 02(3.42E + 00) | 4.03E + 02(2.77E + 00) | 3.99E + 02(1.63E + 00) |
| $f_{50}$ | **3.61E + 02(1.87E + 01)** | 3.89E + 02(2.10E + 00) | 4.47E + 02(3.11E + 00) | 4.10E + 02(2.78E + 00) | 3.94E + 02(5.00E + 00) |

6 Gaussian Adaptation based Parameter Adaptation for Differential Evolution (GaPADE) [29]
7 An Evolutionary Algorithm Based on Covariance Matrix Leaning and Searching Preference for Solving CEC 2014 Benchmark Problems (CMLSP) [8]
8 L-SHADE with the archive rate = 2.0, a population that linearly decreases to 4.0 starting from $18 \times D$ [40].

*4.2.2. Results*

Following the guidelines provided in the IEEE CEC 2014 special session and competition [25], the maximum number of function evaluations (FEs) was set to $5 \times 10^5$ for 50*D* and $1 \times 10^6$ for 100*D*. The mean and the standard deviation of the best-of-run errors for 51 independent runs of all of the contestant algorithms on the 15 benchmark problems for $D = 50$ and $D = 100$ are given in Tables 21 and 22 respectively. The error is the absolute value of the difference between the real optimum value of the objective function $f_{\text{opt}}$ and the best result $f(\vec{X}_{best})$, i.e., $|f_{opt} - f(\vec{X}_{best})|$. A two-sided Wilcoxon rank sum test [13] is used to evaluate the statistical significance of the observed differences between CADE and other algorithms. At the 5% significance level, the tested hypothesis ($H_0$) was that the compared results are independent ones from identical continuous distributions. As indicated in the results tables (Tables 21 and 22), a "–" sign signifies the case where the compared algorithm shows an inferior to that of CADE, and a "+" sign identifies when it exhibits superior performance. In such cases $H_0$ is rejected. When the performance difference is not statistically significant, these cases are marked with a "=" sign. We also show the total number of the aforementioned cases at the end of the second table of each dimension, for each of the competitor algorithms as (+/=/−).

As indicated in Table 21, and considering the mean of the error values for 50*D* problems and as noted from results of the Wilcoxon test, CADE outperformed all contestant algorithms in a statistically significant manner over the 30 benchmark test functions. The last row of each of Tables 21 and 22 indicates a summary of the score of each algorithm in terms of wins "*w*", ties "*t*" and losses "*l*" of the 15 functions, as indicated by the Wilcoxon test, compared to CADE. The comparison of CADE against the other algorithms for $D = 50$ as shown in Table 21 indicates that the proposed algorithm was at least as good as the other competing algorithms in 14 functions. A careful scrutiny of the results of the comparison with the state-of-the-art algorithms in the second category of algorithms (Table 22) shows that CADE obtained the smallest best-of-the-run errors over 13 functions at 100*D*. Figs. 12 and 13 summarize the rankings of all algorithms at all functions at 50D and 100D respectively.

## 5. Conclusion

In this paper, a novel hybrid algorithm, CADE, was introduced that combined the capabilities of two evolutionary algorithms: a customized canonical version of the CA and the DE. In the canonical CA, the Accept() function selects the best individuals that are then used to update the Belief Space knowledge sources using the Update() function . Next, the Influence() function selects the knowledge sources that will influence the evolution of the next generation of the population. In CA's, the major source for exploration is topographic knowledge, knowledge about the functional landscape. In order to support this, the DE can provide a complementary source of exploratory knowledge based upon the distribution of the entire population of individuals in the search space, and hence it makes a perfect complement to the CA optimization process. This cooperation is facilitated in the following way. Both algorithms share the same population space and follow the High-level Teamwork hybrid (HTH) model in which two meta-heuristics are executed in parallel.

A novel success-based quality function was then used to guide the search. Later proportions of evaluation for each component algorithm can be determined using the quality function based on the success of subpopulation (individuals that are currently under its guidance). The effectiveness of CADE algorithm was then compared with 6 other state of the art algorithms using a benchmark set of 50 diverse functions. 35 of the functions came from standard benchmark libraries while the remaining 15 came from the IEEE 2014 Single Function Optimization contest. Its performance against the other algorithms was ranked at #1 statistically. In addition the performance of CADE did not degrade substantially when problem dimensionality was increased from 50 to 100.

**Table 21**
Mean and standard deviation of the error values for functions F1–F30 averaged over 51 runs @ 50D. Best entries are marked in boldface.

| | CoDE | dynNP-jDE | EPSDE | JADE | SHADE | CMLSP | GaAPADE | L-SHADE | CADE |
|---|---|---|---|---|---|---|---|---|---|
| F1 | 2.3214E+05(1.0747E+05) | 3.2264E+05(1.1734E+05) | 1.7374E+06(5.5064E+06) | 1.5275E+04(1.3422E+04) | 1.8614E+04(1.3638E+04) | 1.1457E+04(5.9267E+04) | 1.1728E+03(3.7791E+03) | 1.2000E+03(1.5154E+03) | **5.8233E+02(8.1367E+00)** |
| | − | − | − | − | − | − | − | − | |
| F4 | 2.6067E+01(3.3789E+01) | 8.1654E+01(2.1947E+01) | 4.1596E+01(2.2985E+01) | 1.9346E+01(3.9290E+01) | **9.7224E+00(2.9435E+01)** | 1.0113E+02(2.2918E+02) | 4.9494E+01(4.9078E+01) | 5.9000E+01(4.5608E+01) | 1.3028E+01(1.4806E+01) |
| | − | − | − | − | + | − | − | − | |
| F5 | 2.0032E+01(6.8626E−02) | 2.0384E+01(2.9280E−02) | 2.0596E+01(3.2400E−02) | 2.0356E+01(3.6845E−02) | 2.0140E+01(1.9410E−02) | 2.1138E+01(3.4259E−02) | 2.0006E+01(5.9577E−03) | 2.0255E+01(4.5920E−02) | **2.0000E+01(1.7103E−06)** |
| | = | − | − | − | = | = | = | − | |
| F6 | 8.4983E+00(3.3570E+00) | 7.3352E+00(5.6880E+00) | 4.5596E+01(2.6277E+00) | 1.6589E+01(6.6285E+00) | 5.1707E+00(2.4284E+00) | 3.6735E+00(3.1376E+00) | 1.0381E+01(3.5001E+00) | 2.6000E−01(5.2278E−01) | **6.5263E−02(1.0057E−02)** |
| | − | − | − | − | − | − | − | − | |
| F17 | 1.5714E+04(1.3291E+04) | 1.2583E+04(8.3862E+03) | 2.2167E+05(1.4896E+05) | 2.3000E+03(5.9502E+02) | 2.2347E+03(7.8109E+02) | 1.9605E+03(7.8968E+02) | 1.0322E+03(4.0737E+02) | 1.4567E+03(5.1303E+02) | **5.9041E+02(2.7840E+02)** |
| | − | − | − | − | − | − | − | − | |
| F18 | 3.2916E+02(3.4839E+02) | 2.5984E+02(2.4692E+02) | 3.6879E+03(8.1501E+03) | 1.8203E+02(4.7150E+01) | 1.5375E+02(3.6617E+01) | 3.7008E+03(1.8239E+04) | 4.6955E+01(1.9380E+01) | 1.0161E+02(1.3841E+01) | **3.8638E+01(3.2782E+01)** |
| | − | − | − | − | − | − | − | − | |
| F19 | 6.1691E+00(9.8380E−01) | 1.0832E+01(8.4404E−01) | 2.4584E+01(1.6861E+00) | 1.3421E+01(5.2199E+00) | 9.3810E+00(3.2466E+00) | 1.4425E+01(2.4440E+01) | 9.0591E+00(2.3999E+00) | 8.3000E+00(1.8136E+00) | **5.2944E+00(2.9363E−01)** |
| | = | − | − | − | − | − | − | − | |
| F20 | 2.1861E+02(2.3515E+02) | 3.9163E+01(1.3337E+01) | 4.5080E+02(5.6348E+02) | 7.9529E+03(6.9640E+03) | 1.9816E+02(5.7311E+01) | 3.4630E+01(1.4610E+01) | 3.1197E+01(1.7026E+01) | 1.4000E+01(4.5644E+00) | **1.3751E+01(1.7620E+00)** |
| | − | − | − | − | − | − | − | = | |
| F21 | 6.4912E+03(5.5372E+03) | 2.7281E+03(2.5200E+03) | 7.4370E+04(8.5331E+04) | 2.4322E+04(1.6455E+05) | 1.2297E+03(4.0604E+02) | 1.1439E+03(3.3960E+02) | 5.4510E+02(2.0042E+02) | 5.2000E+02(1.4909E+02) | **4.9736E+02(1.0041E+02)** |
| | − | − | − | − | − | − | − | − | |
| F23 | 3.4400E+02(1.1482E−13) | 3.4400E+02(1.1482E−13) | 3.3705E+02(3.4523E−12) | 3.4400E+02(3.7646E−13) | 3.4400E+02(1.1482E−13) | 3.4401E+02(2.2646E−03) | 3.4400E+02(0.0000E+00) | 3.4400E+02(4.4592E−13) | **3.2857E+02(2.6345E−15)** |
| | − | − | − | − | − | − | − | − | |
| F24 | 2.7139E+02(2.9312E+00) | 2.6546E+02(1.6705E+00) | 2.7312E+02(5.7937E+00) | 2.7482E+02(1.9487E+00) | 2.7478E+02(1.7334E+00) | 2.7082E+02(2.0033E+00) | 2.7373E+02(2.1945E+00) | 2.8000E+02(6.6170E−01) | **2.3718E+02(4.7854E+00)** |
| | − | − | − | − | − | − | − | − | |
| F25 | 2.0792E+02(4.7505E+00) | 2.0691E+02(1.3449E+00) | 2.0116E+02(2.0229E+00) | 2.1699E+02(6.7636E+00) | 2.1217E+02(6.9165E+00) | 2.0625E+02(9.5297E−01) | 2.0530E+02(4.0734E−01) | 2.1000E+02(3.6488E−01) | **2.0008E+02(1.7928E+00)** |
| | − | − | = | − | − | = | = | − | |
| F26 | 1.0618E+02(2.3701E+01) | 1.0035E+02(4.1976E−02) | 1.0035E+02(4.7521E−02) | 1.0237E+02(1.3950E+01) | 1.0431E+02(1.9529E+01) | 1.0032E+02(5.1095E−02) | 1.0017E+02(3.2929E−02) | **1.0000E+02(1.3980E+01)** | **1.0000E+02(3.7826E−01)** |
| | − | − | − | − | − | = | = | **=** | |
| F27 | 5.3680E+02(6.5540E+01) | 3.8452E+02(4.8217E+01) | 1.5566E+03(5.8870E+01) | 4.4058E+02(5.6802E+01) | 4.5329E+02(5.6995E+01) | 3.9728E+02(1.4429E+02) | 5.6604E+02(5.7387E+01) | 3.3546E+02(3.0280E+01) | **3.1963E+02(3.8270E+01)** |
| | − | − | − | − | − | − | − | − | |
| F28 | 1.1734E+03(4.4835E+01) | 1.1052E+03(3.8147E+01) | 3.8746E+02(1.1591E+01) | 1.1289E+03(4.1317E+01) | 1.1413E+03(5.2616E+01) | 1.4799E+03(1.0532E+03) | 1.1615E+03(3.6653E+01) | 1.1222E+03(2.9101E+01) | **3.6186E+02(1.8755E+01)** |
| | − | − | − | − | − | − | − | − | |
| *w/t/l* | 0/2/13 | 0/0/15 | 0/1/14 | 0/0/15 | 1/1/13 | 0/3/12 | 0/3/12 | 0/2/13 | |

**Table 22**
Mean and standard deviation of the error values for functions f1–f30 averaged over 51 runs @ 100D. Best entries are marked in boldface.

| | CoDE | dynNP-jDE | EPSDE | JADE | SHADE | CMLSP | GaAPADE | L-SHADE | CADE |
|---|---|---|---|---|---|---|---|---|---|
| F1 | 7.8408E+05(2.3046E+05) | 1.5304E+06(4.7144E+05) | 3.5283E+05(1.7736E+05) | 1.1644E+05(6.1455E+04) | 1.4280E+051.0527E+05) | 3.3332E+07(2.3395E+08) | 1.5337E+05(6.9922E+04) | 1.7000E+05(5.6543E+04) | **7.5869E+02(1.2618E+02)** |
| | − | − | − | − | − | − | − | − | |
| F4 | 1.5142E+02(3.3456E+01) | 1.6598E+02(3.3829E+01) | 1.3580E+02(4.2348E+01) | 7.7261E+01(5.5558E+01) | 9.2526E+01(4.6796E+01) | 4.4470E+02(1.1051E+03) | 1.5948E+02(3.0777E+01) | 1.7000E+02(3.0634E+01) | **1.9718E+01(8.0341E+00)** |
| | = | = | − | − | − | − | − | − | |
| F5 | 2.0028E+01(5.4426E−02) | 2.0601E+01(3.2027E−02) | 2.1069E+01(5.4879E−02) | 2.0482E+01(7.4319E−02) | 2.0200E+01(1.4669E−02) | 2.1311E+01(2.8635E−02) | 2.0011E+01(1.5886E−02) | 2.1000E+01(3.1110E−02) | **2.0000E+01(2.4722E−06)** |
| | = | − | − | − | − | − | = | − | |
| F6 | 4.4880E+01(6.5725E+00) | 4.0142E+01(1.5199E+01) | 1.2968E+02(4.3802E+00) | 4.4605E+01(1.4840E+01) | 3.3509E+01(5.2490E+00) | 3.2537E+01(1.0866E+01) | 4.4779E+01(1.0268E+01) | 8.7000E+00(2.3336E+00) | **9.7493E−01(3.2946E−01)** |
| | − | − | − | − | − | − | − | − | |
| F17 | 1.2409E+05(4.0440E+04) | 1.2110E+05(4.9013E+04) | 2.5556E+06(4.4312E+06) | 1.2039E+04(4.8820E+03) | 1.2354E+04(4.7972E+03) | 4.2129E+05(2.9666E+06) | 3.8822E+03(5.8167E+02) | 4.4000E+03(7.1340E+02) | **1.7285E+03(2.3194E+02)** |
| | − | − | − | − | − | − | − | − | |
| F18 | 9.9298E+02(1.0156E+03) | 3.2104E+02(3.8450E+02) | 2.9169E+03(3.4706E+03) | 1.0748E+03(9.5074E+02) | 5.0701E+02(3.8870E+02) | 2.5709E+06(1.8357E+07) | 3.7024E+02(6.8761E+02) | 2.2000E+02(1.6791E+01) | **6.7294E+01(1.1428E+01)** |
| | − | − | − | − | − | − | − | − | |
| F19 | 9.2009E+01(1.7679E+01) | 9.0449E+01(1.2663E+00) | 5.2023E+01(2.0511E+01) | 9.9450E+01(8.3109E+00) | 9.7832E+01(1.5491E+01) | 6.8988E+01(2.2368E+01) | 9.4355E+01(1.9937E+00) | 9.6000E+01(2.2879E+00) | **4.4213E+01(1.4713E+01)** |
| | − | − | − | − | − | − | − | − | |
| F20 | 1.9187E+03(9.4802E+02) | 2.6591E+02(5.6307E+01) | 1.2954E+03(2.6087E+03) | 9.5316E+03(1.5166E+04) | 4.9152E+02(1.3438E+02) | 1.4696E+02(5.2736E+01) | 1.7836E+02(4.4764E+01) | 1.5000E+02(5.2469E+01) | **1.2241E+02(5.2824E+01)** |
| | − | − | − | − | − | − | − | − | |
| F21 | 5.8445E+04(2.5505E+04) | 3.6871E+04(1.5383E+04) | 5.2754E+05(8.8753E+05) | 3.6989E+03(1.1894E+03) | 3.6938E+03(1.6153E+03) | 3.6108E+03(7.9740E+02) | 1.7311E+03(5.0012E+02) | 2.3000E+03(5.3084E+02) | **9.8592E+02(2.1786E+02)** |
| | − | − | − | − | − | − | − | − | |
| F23 | 3.4823E+02(3.3301E−13) | 3.4823E+02(1.1482E−13) | 3.4504E+02(4.8172E−10) | 3.4823E+02(9.0668E−13) | 3.4823E+02(4.5610E−13) | 3.5071E+02(2.1390E+00) | 3.4823E+02(5.4853E−13) | 3.4823E+02(3.6628E−13) | **3.4867E+02(4.6092E−14)** |
| | = | = | = | = | = | = | = | = | |
| F24 | 3.8531E+02(4.7593E+00) | 3.6749E+02(2.6976E+00) | 4.0616E+02(8.3431E+00) | 3.9879E+02(5.2302E+00) | 3.9436E+02(4.4474E+00) | 3.8910E+02(1.0101E+01) | 3.8755E+02(3.0567E+00) | 3.9000E+02(2.8738E+00) | **2.9152E+02(4.2211E+00)** |
| | − | − | − | − | − | − | − | − | |
| F25 | 2.5231E+02(1.8111E+01) | 2.5804E+02(9.2705E+00) | 2.6567E+02(2.6963E+01) | 2.7524E+02(6.8524E+00) | 2.5836E+02(5.8654E+00) | 2.3572E+02(6.1367E+00) | 2.0055E+02(3.9079E+00) | 2.0000E+02(4.0892E−13) | **2.0000E+02(3.4183E−16)** |
| | − | − | − | − | − | − | = | = | |
| F26 | 2.0011E+02(1.8248E−02) | 1.9233E+02(2.7060E+01) | **1.3373E+02(4.7403E+01)** | 2.0009E+02(7.4145E−03) | 2.0008E+02(1.2627E−02) | 1.9854E+02(1.4332E+01) | 1.8829E+02(3.2400E+01) | 2.0000E+02(6.2672E−13) | **1.7528E+02(1.0001E+00)** |
| | − | + | + | − | − | − | − | − | |
| F27 | 1.1731E+03(1.3449E+02) | 4.9295E+02(1.5533E+02) | 3.7424E+03(8.0297E+01) | 1.0447E+03(1.1466E+02) | 9.5033E+02(8.8368E+01) | 1.1908E+03(4.0545E+02) | 1.2183E+03(3.0437E+02) | 3.8000E+02(3.2793E+01) | **3.3074E+02(2.4615E+01)** |
| | − | − | − | − | − | − | − | − | |
| F28 | 2.3304E+03(2.4505E+02) | 2.1182E+03(1.4790E+02) | **8.3725E+02(1.9107E+02)** | 2.3307E+03(2.7172E+02) | 2.3170E+03(2.5071E+02) | 5.9375E+03(4.6533E+03) | 2.3507E+03(1.8030E+02) | 2.3000E+03(4.6454E+01) | **9.7993E+02(1.3835E+02)** |
| | − | − | + | − | − | − | − | − | |
| w/t/l | 0/3/12 | 1/2/12 | 2/1/12 | 0/1/14 | 0/1/14 | 0/1/14 | 0/3/12 | 0/2/13 | |

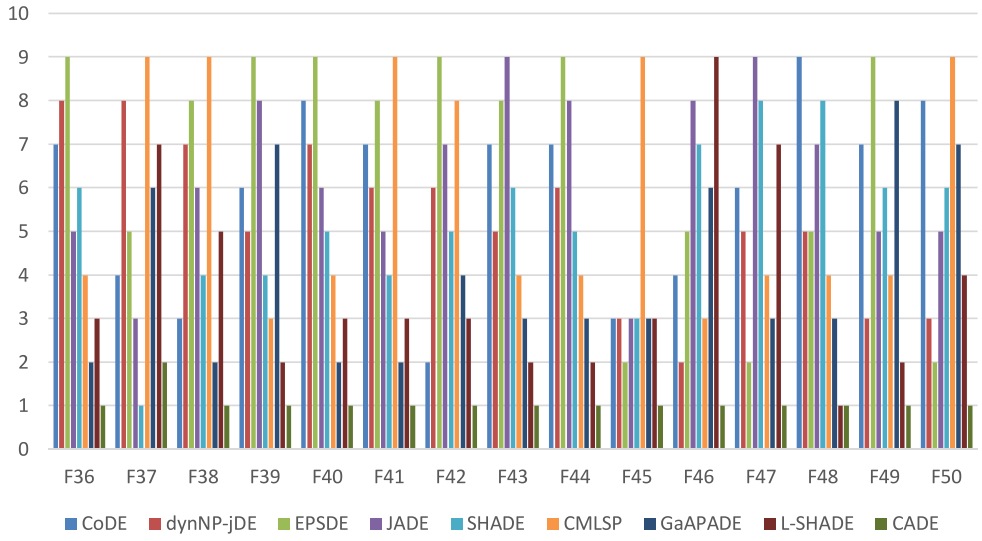


**Fig. 12.** Ranking for CADE and other state-of-the-art algorithms, in experiment 2, according to the mean value on every function at 50*D*.

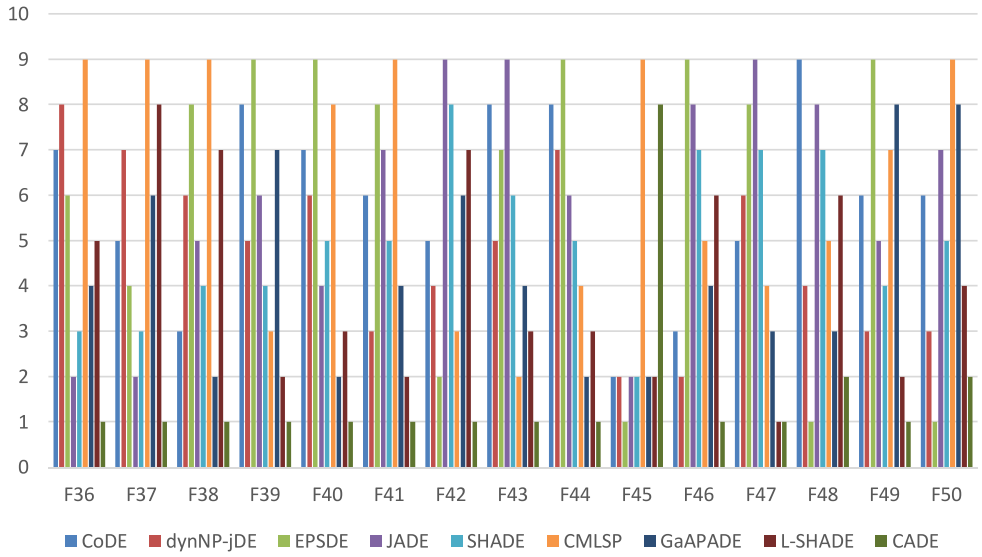


**Fig. 13.** Ranking for CADE and other state-of-the-art algorithms, in experiment 2, according to the mean value on every function at 100*D*.

## References

[1] M.Z. Ali, N.H. Awad, A novel class of niche hybrid cultural algorithms for continuous engineering optimization, Inf. Sci. 267 (2014) 158–190.
[2] M.Z. Ali, N.H. Awad, R.G. Reynolds, Hybrid niche cultural algorithm for numerical global optimization, in: Proc. IEEE Congr. Evol. Comput., Cancun, 2013, pp. 309–316.
[3] N.H. Awad, M.Z. Ali, R.M. Duwairi, Cultural algorithm with improved local search for optimization problems, in: Proc. IEEE Congr. Evol. Comput., Cancun, 2013, pp. 284–291.
[4] R.L. Becerra, C.A.C. Coello, Cultured differential evolution for constrained optimization, Comput. Meth. Appl. Mech. Eng 195 (2006) 4303–4322.
[5] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (6) (2006) 646–657.
[6] J. Brest, M.S. Maucec, Population size reduction for the differential evolution algorithm, Appl. Intell. 29 (2008) 228–247.
[7] Z. Cai, W. Gong, C.X. Ling, H. Zhang, A clustering-based differential evolution for global optimization, Appl. Soft Comput. 11 (2011) 1363–1379.
[8] L. Chen, H.-L. Liu, Z. Zheng, S. Xie, An evolutionary algorithm based on covariance matrix leaning and searching preference for solving CEC 2014 benchmark problems, in: Proc. IEEE Congr. Evol. Comput., 2014, pp. 2672–2677.
[9] J. Chen, B. Xin, Z. Peng, L. Dou, J. Zhang, Optimal contraction theorem for exploration–exploitation tradeoff in search and optimization, IEEE Trans. Syst. Man Cybern. A 39 (2009) 680–691.
[10] S.Y. Chong, M. Tremayne, Combined optimization using cultural and differential evolution: Application to crystal structure solution from powder diffraction data, Chem. Commun. 39 (2006) 4078–4080.
[11] P. Civicioglu, Artificial cooperative search algorithm for numerical optimization problems, Inf. Sci. 229 (2013) 58–76.

[12] L.S. Coelho, V.C. Mariani, An efficient particle swarm optimization approach based on cultural algorithm applied to mechanical design, in: Proc. IEEE Congr. Evol. Comput, Canada, 2006, pp. 1099–1104.
[13] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[14] J. Derrac, S. Garcia, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (2011) 3–18 1.
[15] R. Gamperle, S.D. Muller, P. Koumoutsakos, A parameter study for differential evolution, in: Proc. Advances Intell. Syst., Fuzzy Syst., Evol. Comput, Crete, Greece, 2002, pp. 293–298.
[16] X.Z. Gao, X. Wang, T. Jokinen, S.J. Ovaska, A. Arkkio, K. Zenger, A hybrid optimization method for wind generator design, Int. J Innov. Comput. 8 (6) (2012) 4347–4373.
[17] S. Garcia, F. Herrera, An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons, J. Mach. Learn. Res. 9 (2009) 2677–2694.
[18] F. Glover, Heuristic for integer programming using surrogate constraints, Decision Sci. 8 (1) (1977) 156–166.
[19] Y.-N. Guo, J. Cheng, Y.-Y. Cao, Y. Lin, A novel multi-population cultural algorithm adopting knowledge migration, Soft Comput 15 (2011) 897–905.
[20] J. He, F. Xu, Chaotic-search-based cultural algorithm for solving unconstrained optimization problem, Model. Simul. Eng. 2011 (2011) 1–6.
[21] J.H. Holland, Adaption in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, 1975.
[22] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, 4, 1995, pp. 1942–1948.
[23] Y. Kim, S.-B. Cho, A hybrid cultural algorithm with local search for traveling salesman problem, in: IEEE Int. Conf. Robot. (CIRA), 2009, pp. 188–192.
[24] S. Kirkpatrick, C. Gelatt, M. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680 New series.
[25] J.J. Liang, B.-Y. Qu, P.N. Suganthan, Problem definitions and evaluation criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, Technical Report 201311, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Technical Report, Nanyang Technological University, Singapore, December 2013.
[26] C.-J. Lin, C.-H. Chen, C.-T. Lin, A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications, IEEE Trans. Syst., Man, Cybern. C, Syst. 39 (2009) 55–68.
[27] G. Liu, Y. Li, X. Nie, H. Zheng, A novel clustering-based differential evolution with 2 multi-parent crossovers for global optimization, Appl. Soft Comput. 12 (2) (2012) 663–681.
[28] R. Mallipeddi, P.N. Suganthan, Differential evolution algorithm with ensemble of parameters and mutation and crossover strategies, in: Proc. Swarm Evolutionary and Memetic Computing Conference, 2010, pp. 71–78.
[29] R. Mallipeddi, G. Wu, M. Lee, P.N. Suganthan, Gaussian adaptation based parameter adaptation for differential evolution, in: Proc. IEEE Congr. Evol. Comput., 2014, pp. 1760–1767.
[30] T.T Nguyen, X. Yao, An experimental study of hybridizing cultural algorithms and local search, Int. J. Neural Syst. 18 (2008) 1–18.
[31] B. Peng, R.G. Reynolds, Cultural algorithms: knowledge learning in dynamic environments, in: Proc. IEEE Congr. Evol. Comput., 2004, pp. 1751–1758.
[32] K. Qin, P.N. Suganthan, Self-adaptive differential evolution algorithm for numerical optimization, in: Proc. IEEE Congr Evol Comput, 2005, pp. 1785–1791.
[33] I. Rechenberg, Cybernetic solution path of an experimental problem, Royal Aircraft Establishment Library Translation 1122 (August 1965).
[34] R.G. Reynolds, M.Z. Ali, Cultural algorithms: knowledge-driven engineering optimization via weaving a social fabric as an enhanced influence function, in: Proc. IEEE Congr. Evol. Comput., 2008, pp. 4192–4199.
[35] R.G. Reynolds, An introduction to cultural algorithms, in: Proc. Annu. Conf. Evol. Program, 1994, pp. 131–139.
[36] R. Storn, K.V. Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359.
[37] Y. Sun, L. Zhang, X. Gu, A hybrid co-evolutionary cultural algorithm based on particle swarm optimization for solving global optimization problems, Neurocomputing 98 (2012) 76–89.
[38] E.-C. Talbi, A taxonomy of hybrid metaheuristics, J. Heuristics 8 (5) (2002) 541–564.
[39] R. Tanabe, A. Fukunaga, Success-history based parameter adaptation for differential evolution, in: Proc. IEEE CEC, 2013, pp. 71–78.
[40] R. Tanabe, A. Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: Proc. IEEE CEC, 2014, pp. 1658–1665.
[41] Y. Wang, Z. Cai, Q. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, IEEE Tran. Evol. Comput. 15 (2011) 55–66.
[42] H. Wang, Z. Wu, S. Rahnamayan, Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems, Soft Comput. 15 (11) (2011) 2127–2140.
[43] M. Weber, F. Neri, V. Tirronen, Shuffle or update parallel differential evolution for large-scale optimization, Soft Comput. 15 (11) (2011) 2089–2107.
[44] B. Xin, J. Chen, J. Zhang, H. Fang, Z.-H. Peng, Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: a review and taxonomy, IEEE Trans. Syst. Man Cybern. C 42 (2012) 744–767.
[45] X. Xue, M. Yao, R. Cheng, A novel selection operator of cultural algorithm, Knowl. Eng. Manage. 123 (2012) 71–77.
[46] Z. Yang, K. Tang, X. Yao, Scalability of generalized adaptive differential evolution for large-scale continuous optimization, Soft Comput. 15 (11) (2011) 2141–2155.
[47] J. Zhang, V. Avasarala, A.C. Sanderson, T. Mullen, Differential evolution for discrete optimization: an experimental study on combinatorial auction problems, in: Proc. IEEE World Congr. Comput. Intell., Hong Kong, China, 2008, pp. 2794–2800.
[48] J. Zhang, A.C. Sanderson, JADE: adaptive differential evolution with optional external archive, IEEE Trans Evol. Comput. 13 (5) (2009) 945–958.
[49] J. Zhang, A.C. Sanderson, An approximate Gaussian model of differential evolution with spherical fitness functions, in: Proc. IEEE Congr. Evol. Comput, Singapore, 2007, pp. 2220–2228.
[50] S.-Z. Zhao, P.N. Suganthan, S. Das, Self-adaptive differential evolution with multi-trajectory search for large-scale optimization, Soft Comput. 15 (11) (2011) 2175–2185.