METHODOLOGIES AND APPLICATION

Adaptive multi-population inflationary differential evolution

Marilena Di Carlo[1] · **Massimiliano Vasile[1]** · **Edmondo Minisci[1]**

Abstract

This paper proposes a multi-population adaptive version of inflationary differential evolution algorithm. Inflationary differential evolution algorithm (IDEA) combines basic differential evolution (DE) with some of the restart and local search mechanisms of Monotonic Basin Hopping (MBH). In the adaptive version presented in this paper, the DE parameters CR and F are automatically adapted together with the size of the local restart bubble and the number of local restarts of MBH. The proposed algorithm implements a simple but effective mechanism to avoid multiple detections of the same local minima. The novel mechanism allows the algorithm to decide whether to start or not a local search. The algorithm has been extensively tested over more than fifty test functions from the competitions of the Congress on Evolutionary Computation (CEC), CEC 2005, CEC 2011 and CEC 2014, and compared against all the algorithms participating in those competitions. For each test function, the paper reports best, worst, median, mean and standard deviation values of the best minimum found by the algorithm. Comparisons with other algorithms participating in the CEC competitions are presented in terms of relative ranking, Wilcoxon tests and success rates. For completeness, the paper presents also the single population adaptive IDEA, that can adapt only CR and F, and shows that this simpler version can outperform the multi-population one if the radius of the restart bubble and the number of restarts are properly chosen.

Keywords Global optimisation · Differential evolution · Multi-population algorithm · Adaptive algorithm

1 Introduction

Differential evolution (DE), proposed by Price et al. (2006), is a well-known population-based evolutionary algorithm for solving global optimisation problems over continuous spaces. Literature indicates that DE exhibits very good performance over a wide variety of optimisation problems (Das and Suganthan 2011). However, although being a very efficient optimiser, its local search ability has long been questioned and work has been done to improve its local con-

✉ Marilena Di Carlo
marilenadicarlo01@gmail.com

Massimiliano Vasile
massimiliano.vasile@strath.ac.uk

Edmondo Minisci
edmondo.minisci@strath.ac.uk

[1] Department of Mechanical and Aerospace Engineering, James Weir Building, 75 Montrose Street, Glasgow G1 1XJ, UK

vergence by combining DE with local optimisation strategies (Qing 2010).

In previous works by the authors, Locatelli and Vasile (2015) and Vasile et al. (2011), it was demonstrated that DE can converge to a fixed point, a level set or a hyperplane that does not contain the global minimum. The collapse of the population to a fixed point or a neighbourhood of a fixed point from which DE cannot escape was one of the motivation for the development of inflationary differential evolution algorithm (Vasile et al. 2011).

IDEA is based on the hybridisation of DE with the restarting procedure of monotonic basin hopping (MBH) (Wales and Doye 1997); it implements both a local restart in the neighbourhood of a local minimum and a global restart in the whole search space. IDEA was shown to give better results than a simple DE, but its performance is dependent upon the parameters controlling both the DE and MBH heuristics (Vasile et al. 2011). These parameters are the crossover probability CR, the differential weight F, the radius of the local restart bubble δ_{local} and the number of local restarts n_{LR}, whose best settings are problem dependent. Different adaptive mechanisms for adjusting CR and F

during the search process can be found in the literature, (Brest et al. 2006, 2013; Liu and Lampinen 2005; Omran et al. 2005); a parameter-less adaptive evolutionary algorithm has been presented in Papa (2013). However, no approach has been proposed so far to adapt δ_{local} and n_{LR}. In this paper, we present a simple mechanism to adapt CR and F within a single population and a multi-population strategy to adapt δ_{local} and n_{LR}. The multi-population version of IDEA is in the following called MP-AIDEA (Multi Population Adaptive Inflationary Differential Evolution Algorithm).

The resulting algorithm was extensively tested over 51 test problems from the single objective global optimisation competitions of the Congress on Evolutionary Computation (CEC) 2005, 2011 and 2014. Tests to assess the performance of the algorithm include rankings, Wilcoxon test and success rate. It will be shown that the adaptive version of IDEA always ranks in the first three best algorithms in every competition for every number of dimensions except for the CEC 2014 test set with 30 dimensions. Furthermore, it will be shown that the simple adaptation of CR and F within a single population can outperform the multi-population version with adaptation of δ_{local} and n_{LR} if these two parameters are properly chosen.

This paper extends the work presented in Di Carlo et al. (2015). In Di Carlo et al. (2015), the basic mechanisms that constitute MP-AIDEA were introduced, and the performance of MP-AIDEA was measured only by a relative ranking against other algorithms. This paper provides a more detailed explanation of all the mechanisms and heuristics inside MP-AIDEA; moreover, it presents an extensive empirical assessment of its performance, using several metrics in addition to the relative ranking. As part of this extensive performance evaluation, we compare MP-AIDEA against a number of other algorithms and a single population version of MP-AIDEA with no adaptive local restart. Detailed results obtained for each test functions are also presented, so that the paper can be used as a reference for comparison against other algorithms.

The paper starts stating the problem we are trying to solve in Sect. 2 and briefly introducing the basic principles and fundamental theoretical developments underneath inflationary differential evolution in Sect. 3. The adaptation mechanisms are presented, together with the resulting adaptive multi-population version of IDEA, MP-AIDEA, in Sect. 5. The test cases are presented in Sect. 6, and the obtained results are presented in Sect. 6.1. Finally, the paper presents the results of all the comparative tests in Sects. 6.2, 6.3 and 6.4. Section 7 concludes the paper.

2 Problem statement

This paper is concerned with the following class of global minimisation problems with box constraints:

$$\min_{\mathbf{x} \in B} f(\mathbf{x}) \tag{1}$$

with $f : B \subseteq \mathbb{R}^{n_{\text{D}}} \to \mathbb{R}$, n_{D} the number of dimensions and the box B defined by the upper and lower boundaries $\mathbf{x}_{\text{lower}} \leq \mathbf{x} \leq \mathbf{x}_{\text{upper}}$. In the following, we will use a gradient-based local search algorithm; therefore, we further require that $f \in C^2(B)$. Note, however, that this is not a strict requirement as we can show that the algorithm can work also when a finite number of non-differentiable points exist.

3 Inflationary differential evolution

This section briefly recalls the working principles of inflationary differential evolution and presents the parameters that the algorithm proposed in this paper adapts. Following the notation introduced in Vasile et al. (2011), we express the general DE process as a discrete dynamical system. The governing equation, for the i-th individual at generation k, is expressed as:

$$\mathbf{x}_{i,k+1} = \mathbf{x}_{i,k} + S(\mathbf{x}_{i,k} + \mathbf{u}_{i,k}, \mathbf{x}_{i,k})\mathbf{u}_{i,k} \tag{2}$$

with

$$\begin{aligned}\mathbf{u}_{i,k} = \mathbf{e}\big[&G\mathbf{x}_{r_1,k} + (1-G)\mathbf{x}_{i,k} + F(\mathbf{x}_{r_2,k} - \mathbf{x}_{r_3,k}) \\ &+ (1-G)F(\mathbf{x}_{\text{best},k} - \mathbf{x}_{i,k})\big]\end{aligned} \tag{3}$$

where G can be either 0 or 1 [with $G = 1$ corresponding to the DE strategy DE/*rand* and $G = 0$ corresponding to the DE strategy DE/*current-to-best* (Price et al. 2006)]. In Eq. (3), r_1, r_2 and r_3 are integer numbers randomly chosen in the population, and $\mathbf{e}$ is a mask containing random numbers of 0 and 1 according to:

$$e_t = \begin{cases} 1 \Rightarrow U \leq CR \\ 0 \Rightarrow U > CR \end{cases} \quad t = 1, \ldots, n_{\text{D}} \tag{4}$$

U is a random number taken from a random uniform distribution [0, 1]. The product between $\mathbf{e}$ and the term in square brackets in Eq. (3) has to be intended component-wise. In this work, given $u_{t,i,k}$, the t-th component of the trial vector $\mathbf{u}_{i,k}$, the following correction is applied to satisfy the box constraints (Zhang and Sanderson 2009):

$$u_{t,i,k} = \begin{cases} \left(x_{t,i,k} + x_{t,\text{lower}}\right)/2, & \text{if } u_{t,i,k} < x_{t,\text{lower}} \\ \left(x_{t,i,k} + x_{t,\text{upper}}\right)/2, & \text{if } u_{t,i,k} > x_{t,\text{upper}} \end{cases} \tag{5}$$

The selection function S is defined as:

$$S(\mathbf{x}_{i,k} + \mathbf{u}_{i,k}, \mathbf{x}_{i,k}) = \begin{cases} 1 & \text{if } f(\mathbf{x}_{i,k} + \mathbf{u}_{i,k}) < f(\mathbf{x}_{i,k}) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

In the general case in which the indices r_1, r_2 and r_3 can assume any value, in Vasile et al. (2011) it was demonstrated that the population can converge to a fixed point different from a local minimum or to a level set. Furthermore, in Locatelli and Vasile (2015) it was demonstrated that DE can converge to a hyperplane that does not contain the global minimum. Finally, consider the following proposition.

Proposition 1 *Consider the subset $\Psi = \{\mathbf{x} \in B : f(\mathbf{x}) \leq \bar{f}\}$ and the superset ϕ such that:*

1. *$\Psi \subset \phi$*
2. *$\mathbf{x}_{i,k+1} \in \phi, \forall i$*
3. *$\forall \mathbf{y} \in \phi \setminus \Psi, f(\mathbf{y}) > \bar{f}$*

then if the population at iteration k is entirely contained in Ψ it cannot escape from Ψ at any future iteration.

Proof The proof descends from the definition of S. Suppose that a candidate individual $\mathbf{x}_{i,k+1}$ was generated by map (2) then, because of point 3 of the proposition, it would be rejected by the selection operator. □

Therefore, when the population contracts within a ball $B_c \subseteq \Psi$ of radius ρ_l, DE can only converge to a point or a subset within B_c. We call ρ_l the contraction limit, in the following.

In inflationary differential evolution, the DE heuristics is iterated until the population reaches the contraction limit. A local search is then started from the best individual in the population $\mathbf{x}_{\text{best}}$, the corresponding local minimum $\mathbf{x}_{\text{LM}}$ is saved in an archive of local minima A and the population is restarted in a bubble B_R of radius δ_{local} around the local minimum $\mathbf{x}_{\text{LM}}$. This mechanisms is borrowed from the basic logic underneath monotonic basin hopping (Wales and Doye 1997). To assess if the contraction condition is satisfied, the maximum distance between all possible combinations of individuals of the population at generation k, $\rho^{(k)}$, is computed:

$$\rho^{(k)} = \max\left(||\mathbf{x}_{i,k} - \mathbf{x}_{l,k}||\right) \quad i, l = 1, \ldots, N_{\text{pop}} \tag{7}$$

where N_{pop} is the number of individuals in the population. The contraction is verified when $\rho^{(k)} \leq \bar{\rho}\rho_{\text{max}}$, where $\rho_{\text{max}} = \max_k \rho^{(k)}$ is the maximum value of ρ ever recorded until generation k and $\bar{\rho}$ is one of the parameters of the algorithm, the contraction threshold. This contraction criterion

is consistent with Proposition 1 under the assumption that $\rho_l = \bar{\rho}\rho_{\text{max}}$.

After a number n_{LR} of such local restarts, without any improvement of the current best solution, the archive A collects all the local minima found so far. At this point, the population is restarted globally in the search space so that every individual is initially at a distance $\sqrt{n_D}\delta_{\text{global}}$ from the centres of the clusters of the local minima in A. During local restarts, the most important information is preserved in the local minimum. The assumption is that the basin of attraction of that local minimum has already been explored and that exploration led to the convergence of the population to B_c. When the population is restarted globally the essential information, all the local minima, is stored in the archive A. Here the assumption is that IDEA has completely explored a funnel structure resulting in a cluster of minima.

These restart procedures were proven to be very effective in a series of difficult real problems in which the landscape presents multiple funnels (see Vasile et al. 2011 for additional details).

The complete inflationary differential evolution process with trial vector (3) is governed by the following key parameters: N_{pop}, CR and F, G, $\bar{\rho}$, δ_{local}, n_{LR}, δ_{global}. From experience, we know that δ_{global} is not a critical parameter in most of the cases while CR, F, δ_{local} and n_{LR} play a significant role and are not always easy to define. The parameters CR and F are applied to update each individual in a population while δ_{local} and n_{LR} are applied to restart the whole population. Therefore, in this paper we propose two adaptation mechanisms, one for CR and F and one for δ_{local} and n_{LR}. In particular, the adaptation mechanisms of CR, F and δ_{local} are such as to result in the definition of numerical values for these parameters, to be used by the algorithm. On the contrary, the use of n_{LR} is replaced by a mechanism that allows the algorithm to decide when to perform a local or global restart, so that the definition of a numerical value for n_{LR} is not required anymore.

4 Adaptation mechanisms

Because of the very nature of CR and F, δ_{local} and n_{LR}, the automatic adaptation of CR and F requires only the evaluation of the success of each candidate increment $\mathbf{u}_{i,k}$. On the other hand, the adaptation of δ_{local} and n_{LR} requires the evaluation of the success of the restart of an entire population. Therefore, in this paper it is proposed to extend the working principle of inflationary differential evolution by evolving n_{pop} populations in parallel, where n_{pop} is defined a priori.

Each population adapts its own values of CR and F. We use a stigmergic approach in which the CR and F of each individual are drawn from a joint probability distribution,

over a set of possible values of CR and F, that evolves with the population.

All populations are then concurrently adapting δ_{local} and the number of local restarts. More specifically, the adaptation mechanism of the local restart bubble evolves a probability distribution function over a range of possible values of δ_{local}. Each population draws values from that probability distribution and at each local restart increases the probability associated to the value of δ_{local} that led to a transition from one local minimum to another. The range of δ_{local} is also adapted by taking the mean and the minimum distance among the local minima in A.

The number of local restarts, instead, is dictated by the contraction of a population within the basin of attraction of an already identified local minimum. Given a local minimum $\mathbf{x}_{\text{LM}} \in A$ and a list of $n_{\text{best},LM}$ best individuals from which a local search converged to $\mathbf{x}_{\text{LM}}$, the size of the basin of attraction of $\mathbf{x}_{\text{LM}}$ is defined as
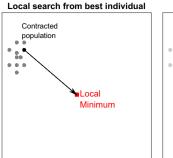
$$d_{\text{basin},LM} = \min_{j} ||\mathbf{x}_{\text{best},j} - \mathbf{x}_{\text{LM}}||, \quad j \in 1, ..., n_{\text{best},LM} \quad (8)$$
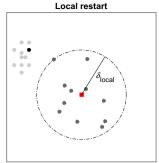
Each local minimum $\mathbf{x}_{\text{LM}}$ in A, therefore, is associated to a particular $d_{\text{basin},LM}$. Figure 1 illustrates this mechanism. Once $d_{\text{basin},LM}$ is estimated, every time the condition $\rho_m^{(k)} \leq \bar{\rho} \rho_{m,\text{max}}$ is satisfied for population m, if the best individual $\mathbf{x}_{\text{best},m}$ is at a distance lower than $d_{\text{basin},LM}$ from $\mathbf{x}_{\text{LM}}$, then no local restart is performed but the population is restarted globally in the search space. The number $n_{\text{best},LM}$ is set to 4 in this implementation.

The overall algorithm, called Multi-Population Adaptive Inflationary Differential Evolutionary Algorithm (MP-AIDEA), is described in more detail in the following section.

5 Multi-population adaptive inflationary differential evolution

MP-AIDEA is described in Algorithm 1. Let n_{pop} be the number of populations and m the index identifying each population. With reference to Algorithm 1, after initialisation of main parameters and functionalities (Algorithm 1, line 1), MP-AIDEA starts by running n_{pop} Differential Evolutions in parallel, one per population (Algorithm 1, line 3). During each evolution process, the parameters F and CR are automatically adapted following the approach presented in Sect. 5.2. When a population m contracts within a ball B_c of radius $\bar{\rho} \rho_{m,\text{max}}$, the evolution of that population is stopped. Once all the populations have contracted, the relative position of the best individual of each population, $\mathbf{x}_{\text{best},m}$ with respect to the local minima in A, $\mathbf{x}_{\text{LM}}$, is assessed (Algorithm 1, line 7). This step makes use of all the minima found by all populations and, therefore, it has to be regarded as an
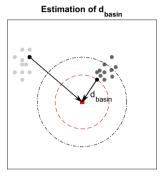

Fig. 1 Identification of the basin of attraction of local minimum $\mathbf{x}_{\text{LM}}$

Algorithm 1 MP-AIDEA

1: Initialisation (Section 5.1, Algorithm 2)
2: **for** $m \in [1, \ldots, n_{pop}]$ **do**
3:     Run Differential Evolution with adaptive CR and F until contraction to B_c (Section 5.2, Algorithms 3 and 4)
4: **end for**
5: **for** $m \in [1, \ldots, n_{pop}]$ **do**
6:     $\mathbf{x}_{best,m} = argmin_{\mathbf{x}_{m,i} \in P_m} f(\mathbf{x}_{m,i})$
7:     **if** $(\forall LM : [\|\mathbf{x}_{best,m} - \mathbf{x}_{LM}\| > d_{basin,LM}$ **or** $i_{LM} < n_{best,LM}])$ **or** $A = \emptyset$ **then**
8:         Run local search and find local minimum $\mathbf{x}_{min,m}^{(s_m)}$
9:         $s_m = s_m + 1$
10:         **if** $\exists LM : \|\mathbf{x}_{min,m}^{(s_m)} - \mathbf{x}_{LM}\| \leq \varepsilon \Delta$ **then**
11:             $i_{LM} = i_{LM} + 1$
12:             $d_{basin,LM} = \min[d_{basin,LM}, ||\mathbf{x}_{best,m} - \mathbf{x}_{LM}||]$
13:             $LR_m = 1$
14:         **else**
15:             $\mathbf{x}_{LM} \leftarrow \mathbf{x}_{min,m}^{(s_m)}$
16:             Store local minima $\mathbf{x}_{LM}$ in A, compute $d_{basin,LM} = \|\mathbf{x}_{best,m} - \mathbf{x}_{LM}\|$
17:             $LR_m = 1$
18:         **end if**
19:     **else**
20:         $LR_m = 0$
21:     **end if**
22: **end for**
23: Update distribution of δ_{local} (Algorithm 6)
24: Restart populations with Algorithm 7 using LR_m, δ_{local} and δ_{global}

25: If total number of function evaluations is lower than maximum number of function evaluations, $n_{feval,max}$, goto (2)

---

information sharing mechanism among populations. If the best individual of population m is not within the basin of

attraction of any previously detected local minimum (that is, $\forall LM : \|\mathbf{x}_{\text{best},m} - \mathbf{x}_{\text{LM}}\| > d_{\text{basin},LM}$) then a local search is run (Algorithm 1, line 8) and the resulting local minimum is stored in the archive A (Algorithm 1, line 16). The flag for the local restart, LR_m, is set to 1. On the contrary, if the best individual of population m is inside the basin of attraction of a previously detected local minimum, the local search is not performed and LR_m is set to 0 (Algorithm 1, line 20).

Before running a local or a global restart (Algorithm 1, line 24), the probability distribution associated to δ_{local} and its range are updated (Algorithm 1, line 23). After restarting the population, if the number of maximum function evaluations, $n_{\text{feval,max}}$, is not exceeded, the process restarts from line 2 in Algorithm 1. Each part of Algorithm 1 is explained in more details hereafter.

5.1 Initialisation

The steps for the initialisation of MP-AIDEA are presented in Algorithm 2. MP-AIDEA starts with the initialisation of n_{pop} populations, with N_{pop} individuals each, in the search space B. The number of function evaluations for each population is set to zero, $n_{\text{feval},m} = 0$ and $\bar{\rho}$, δ_{global}, are initialised to the values specified by the user. The counter of the number of local search per population, s_m, is set to 0.

Algorithm 2 MP-AIDEA: initialisation

1: Set values for $n_{pop}, N_{pop}, \bar{\rho}, \delta_{global}, \varepsilon, s_m = 0 \; \forall m \in [1, \ldots, n_{pop}]$

2: Set $n_{\text{feval},m} = 0$ and $k_m = 1$ (generation number) for each populations $m \in [1, \ldots, n_{pop}]$

3: Initialize population P_m with individuals $\mathbf{x}_{m,i} \; \forall m \in [1, \ldots, n_{pop}]$ and $\forall i \in [1, \ldots, N_{pop}]$

4: Compute $\Delta = \|\mathbf{x}_{upper} - \mathbf{x}_{lower}\|$ where $\mathbf{x}_{lower}$ and $\mathbf{x}_{upper}$ are the lower and upper boundaries of the search space

---

5.2 Differential evolution and the adaptation of CR and F

For each population m, a DE process is run (Algorithm 3, line 6), using Equations 2, 3, 4 and 6. The parameter G, in Equation 3, assumes values equal to 0 or 1 with probability 0.5. During the advancement from parents to offspring, each individual of the population is associated to a different value of CR and F, drawn from a distribution $\mathbf{CRF}_m^{(k_m)}$ (Algorithm 3, lines 1, 2, 3). $\mathbf{CRF}_m^{(k_m=1)}$ is initialised as a uniform distribution with $(n_D + 1)^2$ points in the space $CR \in [0.1, 0.99]$ and $F \in [-0.5, 1]$ (Algorithm 3, line 1). A Gaussian kernel is then allocated to each node and a probability density function is built by Parzen approach (Minisci and Vasile 2014). The values of CR and F to be associated to the individuals of the population are drawn from this dis-

tribution (Algorithm 3, line 4). A change value dd linked to each kernel is initialised to zero (Algorithm 3, line 3) and is used during the advancement of the population from parents to children to adapt CR and F (Algorithm 3, line 8). The adaptation of CR and F is summarised in Algorithm 4 and described in the following.

Algorithm 3 Differential Evolution with adaptive CR and F

1: Regular meshes $\mathbf{CR}$ and $\mathbf{F}$ with $(n_D + 1)^2$ points (n_D is the dimensionality of the problem) in the space $CR \in [0.1, 0.99] \times F \in [-0.5, 1]$ are created

2: Initialize $\mathbf{CRF}_m^{(k_m=1)}$ with points of the mesh: $\mathbf{CRF}_{m,q,1}^{(k_m=1)} \leftarrow \mathbf{CR}_q$ and $\mathbf{CRF}_{m,q,2}^{(k_m=1)} \leftarrow \mathbf{F}_q$ for all $q \in [1, \ldots, (n_D + 1)^2]$

3: Associate to each row of $\mathbf{CRF}_m^{(k_m=1)}$ an element $dd_{m,q}^{(k_m=1)} = 0$ for all $q \in [1, \ldots, (n_D + 1)^2]$

4: Sample $\mathbf{CR}_m^{(k_m)}$ and $\mathbf{F}_m^{(k_m)}$ from a bi-variate distribution on the two dimensional lattice defined by the rows of $\mathbf{CRF}_m^{(k_m=1)}$

5: **for** $i \in [1, \ldots, N_{pop}]$ **do**

6: $\quad \mathbf{x}_{m,i}^{(k_m+1)} \leftarrow \text{DE}\left(\mathbf{x}_{m,i}^{(k_m)}, \mathbf{CR}_m^{(k_m)}, \mathbf{F}_m^{(k_m)}\right)$

7: $\quad n_{feval,m} = n_{feval,m} + 1$

8: $\quad$ Update $\mathbf{CRF}_m^{(k_m)}$ (Algorithm 4)

9: **end for**

10: $k_m = k_m + 1$

11: Row sort $\mathbf{CRF}_m^{(k_m+1)}$ in terms of $dd_m^{(k_m+1)}$ values

12: Compute $\rho_m^{(k_m)} = \max(\|\mathbf{x}_{m,i}^{(k_m)} - \mathbf{x}_{m,l}^{(k_m)}\|) \; \forall \mathbf{x}_{m,i}^{(k_m)}, \mathbf{x}_{m,l}^{(k_m)} \in P_m^{(k_m)}$

13: Until $\rho_m^{(k_m)} \leq \bar{\rho} \cdot \rho_{m,max}$, where $\rho_{m,max} = \max\left[\rho_m^{(k_m=1)}, \rho_m^{(k_m=2)}, \ldots \rho_m^{(k_m)}\right]$, or $k_m < 10D$, goto (4)

Algorithm 4 Updating the joint distribution $\mathbf{CRF}$

1: **if** $f(\mathbf{x}_{m,i}^{(k_m+1)}) < f(\mathbf{x}_{m,i}^{(k_m)})$ **then**

2: $\quad$ Compute $df_{m,i}^{(k_m+1)} = \|f(\mathbf{x}_{m,i}^{(k_m+1)}) - f(\mathbf{x}_{m,i}^{(k_m)})\| \; \forall i \in [1, \ldots, n_{pop}]$

3: $\quad$ **for** $q \in [1, \ldots, (n_D + 1)^2]$ **do**

4: $\quad\quad$ **if** $dd_{m,q} < df_{m,i}^{(k_m+1)}$ **then**

5: $\quad\quad\quad \mathbf{CRF}_{m,q,2}^{(k_m)} \leftarrow F_{m,i}^{(k_m)}$

6: $\quad\quad\quad dd_{m,q}^{(k_m)} \leftarrow df_{m,i}^{(k_m+1)}$

7: $\quad\quad\quad$ **if** $df_{m,i}^{(k_m+1)} > CRC$ **then**

8: $\quad\quad\quad\quad \mathbf{CRF}_{m,q,1}^{(k_m)} \leftarrow CR_{m,i}^{(k_m)}$

9: $\quad\quad\quad$ **end if**

10: $\quad\quad$ **end if**

11: $\quad$ **end for**

12: **end if**

For each individual i of each population m, the adaptation mechanism for CR and F is started only if the child is characterised by an objective function value lower than the parent's one, that is $f(\mathbf{x}_{m,i}^{(k_m+1)}) < f(\mathbf{x}_{m,i}^{(k_m)})$ (Algorithm 4, line 1). If this condition is verified, the difference in objective function between parent and child at subsequent generation, $df_{m,i}^{(k_m+1)} = \|f(\mathbf{x}_{m,i}^{(k_m+1)}) - f(\mathbf{x}_{m,i}^{(k_m)})\|$, is computed (Algo-

rithm 4, line 2). Then the sorted elements of $\mathbf{CRF}_m^{(k_m)}$ are sequentially evaluated; the q-th value of CR in $\mathbf{CRF}_m^{(k_m)}$ is identified as $\mathbf{CRF}_{m,q,1}^{(k_m)}$ and the q-th value of F is identified as $\mathbf{CRF}_{m,q,2}^{(k_m)}$. The first time that $dd_{m,q}^{(k_m)}$ (the dd value associated to the q-th row of $\mathbf{CRF}_m^{(k_m)}$) is lower than $df_{m,i}^{(k_m+1)}$ (Algorithm 4, line 4), the differential weight $F_{m,i}^{(k_m)}$ used for the individual $\mathbf{x}_{m,i}^{(k_m)}$ substitutes $\mathbf{CRF}_{m,q,2}^{(k_m)}$ and $df_{m,i}^{(k_m+1)}$ substitutes $dd_{m,q}^{(k_m)}$ (Algorithm 4, lines 5 and 6). This is because $F_{m,i}^{(k_m)}$ produced a bigger decrease in the objective function than $\mathbf{CRF}_{m,q,2}^{(k_m)}$ (as shown by $df_{m,i}^{(k_m+1)} > dd_{m,q}^{(k_m)}$). For CR, the value associated to $\mathbf{x}_{m,i}^{(k_m)}$ substitutes $\mathbf{CRF}_{m,q,1}^{(k_m)}$ (Algorithm 4, line 8) only if $df_{m,i}^{(k_m+1)}$ is greater than a given value CRC (Algorithm 4, line 7), (Minisci and Vasile 2014).
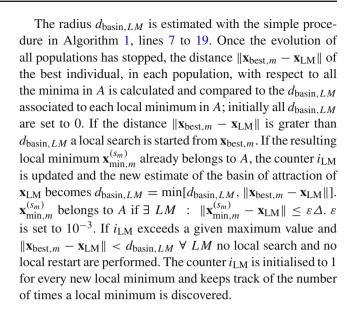
The DE stops according to the contraction condition presented in Sect. 3. In order to prevent an excessive use of resources when the population partitions, a fail safe criterion was introduced that stops the DE after $10D$ generations (Algorithm 3, line 13).

5.3 Local search and restart mechanisms

After the evolution of all populations has stopped, MP-AIDEA checks if the best individual of each population is inside the basin of attraction of any previously detected local minimum (see Algorithm 1, line 7). If that is not the case, a local search is performed from the best individual and the population is locally restarted within a hypercube with edge equal to $2\delta_{\text{local}}$ around the detected local minimum; otherwise, no local search is performed and the population is restarted globally in the whole search space (Algorithm 1, line 24). Prior to the implementation of the restart mechanisms, MP-AIDEA updates the estimation of the size of the basin of attraction of each minimum, the archive A (see Algorithm 1, lines 5 to 22) and the distribution over the possible values of $2\delta_{\text{local}}$ (see Algorithm 1, line 23). In the following the identification of the basin of attraction, the estimation of δ_{local} and the two restart mechanisms are described in more details.

5.3.1 Identification of the basin of attraction

In order to mitigate the possibility of running multiple local searches that converge to already discovered local minima, MP-AIDEA estimates for each local minimum in A the radius of the basin of attraction of that local minimum. The radius of the basin of attraction is here defined as the distance $d_{\text{basin},LM}$ from a given local minimum $\mathbf{x}_{\text{LM}}$ such that if the best individual in population m, $\mathbf{x}_{\text{best},m}$, is at a distance from $\mathbf{x}_{\text{LM}}$ lower than $d_{\text{basin},LM}$ a local search starting from $\mathbf{x}_{\text{best},m}$ would converge to $\mathbf{x}_{\text{LM}}$.

The radius $d_{\text{basin},LM}$ is estimated with the simple procedure in Algorithm 1, lines 7 to 19. Once the evolution of all populations has stopped, the distance $\|\mathbf{x}_{\text{best},m} - \mathbf{x}_{\text{LM}}\|$ of the best individual, in each population, with respect to all the minima in A is calculated and compared to the $d_{\text{basin},LM}$ associated to each local minimum in A; initially all $d_{\text{basin},LM}$ are set to 0. If the distance $\|\mathbf{x}_{\text{best},m} - \mathbf{x}_{\text{LM}}\|$ is grater than $d_{\text{basin},LM}$ a local search is started from $\mathbf{x}_{\text{best},m}$. If the resulting local minimum $\mathbf{x}_{\text{min},m}^{(s_m)}$ already belongs to A, the counter i_{LM} is updated and the new estimate of the basin of attraction of $\mathbf{x}_{\text{LM}}$ becomes $d_{\text{basin},LM} = \min[d_{\text{basin},LM}, \|\mathbf{x}_{\text{best},m} - \mathbf{x}_{\text{LM}}\|]$. $\mathbf{x}_{\text{min},m}^{(s_m)}$ belongs to A if $\exists\ LM\ :\ \|\mathbf{x}_{\text{min},m}^{(s_m)} - \mathbf{x}_{\text{LM}}\| \leq \varepsilon\Delta$. ε is set to 10^{-3}. If i_{LM} exceeds a given maximum value and $\|\mathbf{x}_{\text{best},m} - \mathbf{x}_{\text{LM}}\| < d_{\text{basin},LM}\ \forall\ LM$ no local search and no local restart are performed. The counter i_{LM} is initialised to 1 for every new local minimum and keeps track of the number of times a local minimum is discovered.

5.3.2 Adaptation of δ_{local}

When a population m is locally restarted, individuals are generated by taking a random sample, with Latin Hypercube, within a hypercube with edge equal to $2\delta_{local,m}$. The dimension $\delta_{local,m}$ is drawn from a probability distribution that is progressively updated at every restart. We use a kernel approach with kernels centred in the elements of a vector $\mathbf{B}$ (see Algorithm 6) containing a range of possible values of $\delta_{local,m}$. The vector $\mathbf{B}$ is initialised, with the procedure presented in Algorithm 5, when all populations performed a local search for the first time and at every global restart. During initialisation the distance between all the local minima in the archive A is computed (Algorithm 5, line 1) and $\mathbf{B}$ is initialised with values spanning the interval between the minimum and the mean distance among minima (Algorithm 5, lines 2–3). The mean values instead of the max is used to limit the size of the restart bubble and speed up convergence under the assumption that a local restart needs to lead to the local exploration of the search space. In the experimental tests, it will be shown that this working assumption is generally verified and $\delta_{local,m}$ tends to converge to small values. Then, a second vector $\mathbf{dd}_b$, with the same number of components of $\mathbf{B}$, is initialised to zero (Algorithm 5, line 4).

During the update phase of $\delta_{local,m}$, MP-AIDEA uses the index s_m to keep track of the number of times population m performed a local search and calculates the difference p_m between two subsequent local minima (see Algorithm 6, line 5). The value p_m is then compared to the elements in $\mathbf{dd}_b$ and when $dd_{b,q} < p_m$ then $\delta_{local,m}$ replaces B_q, and p_m replaces $dd_{b,q}$ (Algorithm 6, lines 7-10). In other words, if the $\delta_{local,m}$ used to restart population m led to a local minimum $\mathbf{x}_{\text{min},m}^{(s_m)}$ different from $\mathbf{x}_{\text{min},m}^{(s_m-1)}$, the local minimum previously

identified by the same population, the probability of sampling $\delta_{local,m}$ is increased.

Algorithm 5 Initialise **B**

1: Compute d_{minMIN} and $d_{minMEAN}$
2: Create 1-dimensional regular grid with $(n_D + 1)$ points in the interval $[d_{minMIN}, d_{minMEAN}]$
3: Initialise **B** with points of the grid
4: Initialise vector $\mathbf{dd}_b$ associated to **B** with element $dd_{b,q} = 0$ for all $q \in [1, \ldots, (n_D + 1)]$

Algorithm 6 Update the distribution of δ_{local}

1: **if** All populations did local search for the 1st time **then**
2:     Create vector **B** using Algorithm 5
3: **end if**
4: **for** $m \in [1, \ldots, n_{pop}]$ **do**
5:     Compute $p_m = ||\mathbf{x}_{min,m}^{(s_m)} - \mathbf{x}_{min,m}^{(s_m - 1)}||$
6:     **for** $q \in [1, \ldots, n_D + 1]$ **do**
7:         **if** $\mathbf{dd}_{b,q} < p_m$ **then**
8:             $\mathbf{B}_q \leftarrow \delta_{local,m}$
9:             $\mathbf{dd}_{b,q} \leftarrow p_m$
10:         **end if**
11:     **end for**
12: **end for**
13: Row sort **B** according to $\mathbf{dd}_b$ values

Algorithm 7 MP-AIDEA: local and global restart

1: **for** $m \in [1, \ldots, n_{pop}]$ **do**
2:     **if** $LR_m = 1$ **then**
3:         Sample $\delta_{local,m}$ from the kernel distribution over the values in **B**
4:         L.R.: Initialise population P_m in a hypercube centred in $\mathbf{x}_{min,m}^{(s_m)}$ with edge $2\delta_{local,m}$ for all $m \in \{1, \ldots, n_{pop}\}$
5:     **else**
6:         Cluster local minima in A and compute cluster baricentres $\mathbf{x}_c$
7:         G.R.: Initialise population $P_m = \{\mathbf{x}_{m,i} : ||\mathbf{x}_{m,i} - \mathbf{x}_c|| > \sqrt{n_D}\delta_{global}, \forall i \in \{1, \ldots, N_{pop}\}\}$
8:         Initialise vector **B** using Algorithm 5
9:     **end if**
10: **end for**

5.3.3 Local and global restart

After the identification of the basin of attraction and the update of the value of δ_{local}, populations undergo a restart process in which a new population is generated either by sampling a neighbourhood of a local minimum (local restart) or by sampling the whole search space (global restart). The two restart procedures are described in Algorithm 7.

The local restart procedure takes the latest identified local minimum $\mathbf{x}_{min,m}^{(s_m)}$ of population m and restart the population

with Latin Hypercube sampling in a box centred in $\mathbf{x}_{min,m}^{(s_m)}$ with edge length $2\delta_{local,m}$.

The global restart procedure identifies clusters of local minima with a Fuzzy C-Mean algorithm (Bezdek 1981), computes the centre of each cluster and initialises population m so that each individual is at distance at least $\sqrt{n_D}\delta_{global}$ from each of the centres of the clusters (Algorithm 7, lines 6 and 7).

At each local and global restart, the **CRF** matrix is re-initialised while the vector **B** is initialised only after every global restart. The motivation for re-initialising **CRF** at every restart is twofold: on the one hand different values of *CR* and *F* might be optimal in different parts of the search space, and on the other hand convergence to the optimal value of *CR* and *F* is not always guaranteed. In search spaces with uniform and homogeneous structures, restarting **CRF** and **B** might lead to an overhead on the computational cost; therefore, in future implementations we will test the possibility of retaining **CRF** and **B** across the restart process.

5.4 Computational complexity

The computational complexity of MP-AIDEA is defined by the three main sets of operations:

- **Local search**. The local search uses the Matlab *fmincon* function which implements an SQP scheme with BFGS estimation of the Hessian matrix. Since the matrix is generally dense, its decomposition is $\mathcal{O}(n_D^3)$.
- **Adaptation of CR and F**. The adaptation of *CR* and *F* for each individual in each population is the other expensive bit of the algorithm and is $\mathcal{O}(n_{pop}N_{pop}n_D^2)$ (see line 2 in Algorithm 1, line 8 in Algorithm 3 and line 3 in Algorithm 4). As a comparison, the computational complexity of the standard DE is $\mathcal{O}(N_{pop})$.
- **Restart mechanisms.** The cost of the local restart procedure is limited to the generation of $n_{pop}N_{pop}$ individuals, while the global restart has a cost associated also to clustering, which is $\mathcal{O} = (n_{LM}^2 n_D n_{iter})$ (Bezdek 1981), where n_{iter} is the number of iterations for the clustering, and one associated to the verification that the new population is far from the clusters, which is $\mathcal{O}(N_{pop}n_{LM})$ (see line 7 of Algorithm 7).

Overall when $n_{pop}N_{pop} < n_D$ the dominant algorithmic cost is the local search while the adaptation of *CR* and *F* becomes more expensive for large and numerous populations. Since in the experimental test cases we will use $N_{pop} = n_D$ and $n_{pop} = 4$ the overall algorithmic complexity remains $\mathcal{O}(n_D^3)$.

Table 1 Functions of the CEC 2005 test set

	Unimodal functions	
	1	Shifted sphere function
	2	Shifted Schwefel's problem
	3	Shifted rotated high conditioned elliptic function
	5	Schwefel's problem with global optimum on bounds
	Multimodal functions	
	6	Shifted Rosenbrock's function
	7	Shifted rotated Griewank's function without bounds
	8	Shifted rotated Ackley's function with global optimum on bounds
	9	Shifted Rastrigin's function
	10	Shifted rotated Rastrigin's function
	11	Shifted rotated Weierstrass function
	12	Schwefel's problem
	13	Expanded extended Griewank's plus Rosenbrock function
	14	Shifted rotated expanded Scaffer's
	Hybrid composition functions	
	15	Hybrid composition function
	16	Rotated hybrid composition function
	18	Rotated hybrid composition function
	19	Rotated hybrid composition function with narrow basin for the global opt.
	20	Rotated hybrid composition function with the global optimum on the bounds
	21	Rotated hybrid composition function
	22	Rotated hybrid composition function with high condition number matrix

Table 2 Functions of the CEC 2011 test set

1	Parameter estimation for frequency-modulated sound waves (n_D=6)
2	Lennard-Jones potential problem (n_D=30)
3	The bifunctional catalyst blend optimal control problem (n_D=1)
5	Tersoff potential for model Si(B) (n_D=30)
6	Tersoff potential for model Si(C) (n_D=30)
7	Spread spectrum radar polyphase code design (D=20)
10	Circular antenna array design problem (n_D=12)
12	Messenger: spacecraft trajectory optimisation problem (n_D=26)
13	Cassini 2: spacecraft trajectory optimisation problem (n_D=22)

6 Experimental performance analysis

The effectiveness of MP-AIDEA is tested on a benchmark composed of three test sets. The three test sets are made of functions taken from three past competitions of the Congress on Evolutionary Computation (CEC). We took 20 functions from CEC 2005 (Suganthan et al. 2005), 9 real-world problems from CEC 2011 (Das and Suganthan 2010) and 22 functions from CEC 2014 (Liang et al. 2013), for a total of 51 different problems. The list of functions used in each test set is reported in Tables 1, 2 and 3. They include both academic test functions and real-world optimisation problems. Since we are interested in solving problem (1), all functions selected for this benchmark are continuous and differentiable

We used four different metrics to evaluate MP-AIDEA against the algorithms that participated in the three CEC competitions:

– Metric 1: Best, worst, median, mean and standard deviation of the best result over a given number of independent runs of the algorithm.
– Metric 2: Ranking against the other algorithms using the same ranking approach proposed in the CEC 2011 competition.
– Metric 3: Wilcoxon test. This is used to compare MP-AIDEA to the algorithm participating in the CEC 2011 and CEC 2014 for which the source code is available online.

Table 3 Functions of the CEC 2014 test set

	Unimodal functions	
	1	Rotated high conditioned elliptic function
	2	Rotated Bent Cigar function
	3	Rotated Discus function
	Multimodal functions	
	4	Shifted and rotated Rosenbrock's function
	5	Shifted and rotated Ackley's function
	7	Shifted and rotated Griewank's function
	8	Shifted Rastrigin's function
	9	Shifted and rotated Rastrigin's function
	10	Shifted Schwefel's function
	11	Shifted and rotated Schwefel's function
	13	Shifted and rotated HappyCat function
	14	Shfited and rotated HGBat function
	15	Shifted and rotated expanded Griewank's plus Rosenbrock's function
	16	Shfited and rotated expanded Scaffer's F6 function
	Hybrid function	
	17	Hybrid function 1
	18	Hybrid function 2
	20	Hybrid function 4
	21	Hybrid function 5
	Composition function	
	23	Composition function 1
	24	Composition function 2
	25	Composition function 3
	28	Composition function 6

Table 4 Settings for the CEC 2005, CEC 2011 and CEC 2014 test functions

	CEC 2005	CEC 2011	CEC 2014
Problems settings			
n_D	10, 30, 50	–	10, 30, 50, 100
$n_{feval,max}$	10000 n_D	150000	10000 n_D
n_{runs}	25	25	51
MP-AIDEA settings			
n_{pop}	4	4	4
N_{pop}	n_D	n_D	n_D
$\bar{\rho}$	0.2	0.2	0.2
δ_{global}	0.1	0.1	0.1

– Metric 4: Success rate. This is used to compare MP-AIDEA to the algorithm participating in the CEC 2011 and CEC 2014 for which the source code is available online.

The settings of MP-AIDEA were maintained constant for all problems within a particular test set and were changed going from one test set to another. This is in line with the way all the other algorithms competed. Table 4 summarises the parameters and settings used for the CEC 2005, CEC 2011 and CEC 2014 test functions. More details about the chosen parameters will be given in Sect. 6.1.

The ranking of the algorithms participating in every competition was adjusted to account only for their performance on the selected subset of differentiable functions.

It will be shown that all metrics lead to similar conclusions: MP-AIDEA ranks among the first four algorithms, if not first, in all three test sets and for all dimensions. We will also show that MP-AIDEA can detect previously undiscovered minima on some particularly difficult functions.

The current implementation of MP-AIDEA can be found open source at https://github.com/strath-ace/smart-o2c together with the benchmark of test cases.

6.1 Test sets

This section briefly describes each test set, the settings of MP-AIDEA and metric 1 for all test sets.

Table 5 Objective functions error of the CEC 2005 test set in dimension 10D and 30D

	Best	Worst	Median	Mean	Std
10D					
1	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
2	0.00e+00	1.14e−13	0.00e+00	0.00e+00	4.34e−14
3	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
5	5.60e−06	1.70e−04	6.17e−05	6.59e−05	4.36e−05
6	3.04e−10	2.33e−09	1.80e−09	1.60e−09	6.06e−10
7	4.83e−13	1.48e−02	1.02e−10	1.97e−03	4.21e−03
8	2.00e+01	2.00e+01	2.00e+01	2.00e+01	6.65e−10
9	0.00e+00	9.95e−01	0.00e+00	3.98e−02	1.99e−01
10	0.00e+00	3.98e+00	1.99e+00	1.79e+00	1.04e+00
11	3.29e+00	5.88e+00	5.31e+00	4.71e+00	6.18e−01
12	0.00e+00	1.19e−12	5.68e−14	1.71e−13	2.79e−13
13	9.87e−03	5.31e−01	2.66e−01	2.40e−01	1.58e−01
14	3.32e−01	3.52e+00	2.13e+00	2.11e+00	6.70e−01
15	0.00e+00	4.00e+02	2.84e−14	2.98e+01	8.14e+01
16	0.00e+00	1.15e+02	1.00e+02	9.53e+01	2.25e+01
18	3.00e+02	9.00e+02	8.00e+02	7.18e+02	2.43e+02
19	3.00e+02	9.06e+02	8.00e+02	7.45e+02	2.03e+02
20	3.00e+02	9.38e+02	8.00e+02	6.83e+02	2.46e+02
21	3.00e+02	8.00e+02	3.00e+02	4.20e+02	1.50e+02
22	3.00e+02	8.01e+02	7.54e+02	6.53e+02	2.01e+02
30D					
1	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
2	0.00e+00	2.27e−13	1.14e−13	5.68e−14	6.46e−14
3	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
5	1.81e−01	1.52e+00	4.58e−01	5.13e−01	2.97e−01
6	5.81e−10	4.07e+00	8.25e−03	3.45e−01	8.95e−01
7	4.26e−13	1.79e−11	2.64e−12	4.58e−12	4.86e−12
8	2.00e+01	2.00e+01	2.00e+01	2.00e+01	9.26e−13
9	0.00e+00	5.97e+00	2.21e+00	2.40e+00	1.49e+00
10	1.99e+01	4.78e+01	3.18e+01	3.05e+01	7.16e+00
11	1.57e+01	2.69e+01	2.09e+01	2.12e+01	2.99e+00
12	8.24e−12	5.89e+02	1.05e+01	1.22e+02	2.06e+02
13	8.88e−01	2.66e+00	1.64e+00	1.60e+00	4.44e−01
14	1.10e+01	1.26e+01	1.17e+01	1.17e+01	3.77e−01
15	2.27e+01	4.00e+02	4.00e+02	3.15e+02	1.37e+02
16	4.16e+01	6.85e+01	5.68e+01	5.69e+01	6.99e+00
18	8.00e+02	9.11e+02	9.09e+02	8.87e+02	4.43e+01
19	8.00e+02	9.12e+02	9.06e+02	8.73e+02	5.14e+01
20	8.00e+02	9.13e+02	9.07e+02	8.78e+02	4.99e+01
21	5.00e+02	5.00e+02	5.00e+02	5.00e+02	4.91e−11
22	8.78e+02	9.22e+02	9.10e+02	9.06e+02	1.04e+01

6.1.1 CEC 2005 test set

Following the rules of the CEC 2005 competition, MP-AIDEA was applied to the solution of the problems in the CEC 2005 test set in dimension $n_D = 10$, 30 and 50, with a maximum number of function evaluation equal to $n_{feval,max} = 10000 n_D$. The experiments were repeated for a total of $n_{runs} = 25$ independent runs for each function (Suganthan et al. 2005). Functions 4, 17, 24 and 25 of the CEC

Table 6 Objective functions error of the CEC 2005 test set in dimension 50D

	Best	Worst	Median	Mean	Std
1	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
2	5.68e−14	5.68e−13	1.14e−13	5.68e−14	1.45e−13
3	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
5	8.28e−01	1.97e+01	2.52e+00	4.25e+00	4.82e+00
6	3.80e−10	3.11e+01	2.58e+01	2.27e+01	8.82e+00
7	6.11e−12	2.25e−07	8.05e−11	1.00e−08	4.50e−08
8	2.00e+01	2.00e+01	2.00e+01	2.00e+01	2.00e−12
9	4.97e+00	1.29e+01	7.96e+00	8.41e+00	2.14e+00
10	5.47e+01	1.01e+02	7.66e+01	7.61e+01	1.17e+01
11	3.62e+01	5.94e+01	4.57e+01	4.64e+01	6.50e+00
12	4.80e+01	9.37e+03	8.07e+02	1.24e+03	1.84e+03
13	2.87e+00	5.00e+00	3.96e+00	3.89e+00	6.35e−01
14	2.04e+01	2.19e+01	2.12e+01	2.12e+01	4.13e−01
15	2.57e+01	4.00e+02	2.88e+02	3.08e+02	1.00e+02
16	5.10e+01	7.65e+01	6.08e+01	6.25e+01	6.96e+00
18	3.04e+02	9.34e+02	9.24e+02	8.65e+02	1.30e+02
19	8.00e+02	9.34e+02	9.25e+02	8.92e+02	5.85e+01
20	3.00e+02	9.65e+02	9.13e+02	8.52e+02	1.32e+02
21	5.00e+02	5.00e+02	5.00e+02	5.00e+02	7.65e−08
22	9.20e+02	9.70e+02	9.48e+02	9.50e+02	1.31e+01

2005 competition were not included in the test set because non-differentiable.

The number of populations in MP-AIDEA was set to $n_{pop} = 4$ and the number of individuals in each population was set to $N_{pop} = n_D$. The number of populations to be deployed on a particular problem depends on the type and complexity of that problem, and the available number of function evaluations. We tested MP-AIDEA with different numbers of populations from 1 to 4 (results using MP-AIDEA with one population are presented in Sect. 6.2). Results showed that MP-AIDEA with 4 populations performs consistently well on all benchmarks, and, thus, we decided to present our findings for $n_{pop} = 4$. The contraction limit was set to $\bar{\rho} = 0.2$ and the global restart distance was set to $\delta_{global} = 0.1$ (Table 4). In line with the metrics presented at the CEC 2005 competition, Tables 5 and 6 reports the difference, in the objective value, between the result obtained with MP-AIDEA and the known global minimum.

Table 7 reports the best objective function error values obtained by all the algorithms participating in the CEC 2005 competition and MP-AIDEA for functions 13 and 16 and $n_D = 10$. According to the CEC 2005 specifications, the accuracy level for the detection of the global minimum is 10^{-2} for these functions. MP-AIDEA is able to identify the global minimum of both functions 13 and 16. Previously only EvLib (Becker 2005) succeeded in identifying the global minimum of function 13 and no other algorithm managed to find the global minimum of function 16.

6.1.2 CEC 2011 test set

Following the rules of the CEC 2011 competition (Das and Suganthan 2010), MP-AIDEA was run for $n_{feval,max} = 150000$ function evaluations on the CEC2011 test set. The experiments were repeated for $n_{runs} = 25$ independent runs. Test functions with equality and inequality constraints were not included in the tests. The number of populations n_{pop}

Table 7 CEC 2005 best objective function error values for functions 13 and 16, $n_D = 10$

Algorithm	Function 13	Function 16
BLX-GL50 (García-Martínez and Lozano 2005)	3.70e−01	7.20e+01
BLX-MA (Molina et al. 2005)	3.80e−01	9.00e+01
CoEVO (Pošik 2005)	4.70e−01	1.20e+02
DE (Ronkkonen et al. 2005)	4.60e−01	1.50e+02
DE (Bui et al. 2005)	2.70e−01	1.00e+02
DMS-L-PSO (Liang and Suganthan 2005)	2.50e−01	5.20e+01
EDA (Yuan and Gallagher 2005)	1.60e+00	1.30e+02
ES (Costa 2005)	7.90e−01	9.70e+01
EvLiv (Becker 2005)	9.90e−03	1.20e+02
flexGA (Alonso et al. 2005)	4.20e−02	1.10e+02
G-CMA-ES (Auger and Hansen 2005b)	4.10e−01	7.90e+01
K-PCX (Sinha et al. 2005)	3.30e−01	8.80e+01
L-CMA-ES (Auger and Hansen 2005a)	1.90e−01	6.10e+01
L-SaDE (Qin and Suganthan 2005)	1.20e−01	8.60e+01
SPC-PNX (Ballester et al. 2005)	3.50e−01	9.10e+01
MP-AIDEA	9.87e−03	0.00e+00

Table 8 Objective functions of the CEC 2011 test set

	Best	Worst	Median	Mean	Std
1	9.30e−19	1.09e+01	6.67e−15	8.44e−01	2.92e+00
2	−2.84e+01	−2.71e+01	−2.76e+01	−2.79e+01	4.74e−01
3	1.15e−05	1.15e−05	1.15e−05	1.15e−05	5.83e−17
5	−3.68e+01	−3.45e+01	−3.60e+01	−3.61e+01	7.40e−01
6	−2.92e+01	−2.30e+01	−2.74e+01	−2.72e+01	2.32e+00
7	5.00e−01	7.13e−01	5.00e−01	5.31e−01	5.30e−02
10	−2.18e+01	−2.14e+01	−2.16e+01	−2.16e+01	1.42e−01
12	6.88e+00	1.51e+01	1.22e+01	1.15e+01	2.53e+00
13	8.71e+00	1.98e+01	1.43e+01	1.34e+01	3.10e+00

was set to 4 and the number of individuals in each population was set to $N_{pop} = 30$ regardless of the dimensionality of the problem. The contraction limit and the global restart distance were set, respectively, to $\bar{\rho} = 0.2$ and $\delta_{global} = 0.1$ (Table 4). Table 8 reports the best, worst, median, mean objective function found by MP-AIDEA and the associated standard deviation.

6.1.3 CEC 2014 test set

In line with the rules of the CEC 2014 competition (Liang et al. 2013), MP-AIDEA was applied to the solution of the functions in the CEC 2014 test set in dimension $n_D = 10, 30, 50$ and 100, with maximum number of function evaluations $n_{feval,max} = 10000n_D$. The experiments were repeated for $n_{runs} = 51$ independent runs. Non-differentiable functions 6, 12, 19, 22, 26, 27, 29 and 30 were not included in the test set (see Table 3). The number of populations was set to $n_{pop} = 4$ and the number of individuals in each population was set to $N_{pop} = n_D$. The contraction limit and the global restart distance were set, respectively, to $\bar{\rho} = 0.2$ and $\delta_{global} = 0.1$ (Table 4).

Tables 9 and 10 report the difference between the objective value found by MP-AIDEA and the known global minimum. In agreement to the guidelines of the competition error values smaller than 10^{-8} are reported as zero, (Liang et al. 2013). Table 11 reports the best objective function values obtained by all the algorithms participating in the competition and MP-AIDEA for functions 9, 10, 11 and 15 in 10 dimensions. MP-AIDEA finds the global minimum of function 11, unlike all the other competing algorithms, and gives good results for the other functions.

6.2 Ranking

In this section, MP-AIDEA is ranked against a group of algorithms participating in each CEC competition. The rankings include those algorithms that reported their results in a paper and MP-AIDEA with two different settings:

– $n_{pop} = 4$ and $N_{pop} = n_D$. This settings will be indicated as "MP-AIDEA" in the following and corresponds to the settings that was used to generate the results in Sect. 6.1.
– $n_{pop} = 1$, $N_{pop} = 4n_D$; MP-AIDEA adapts CR and F but uses fixed values for δ_{local} and n_{LR}. In particular, $n_{LR} = 10$ and $\delta_{local} = 0.1$, unless otherwise specified. This settings will be indicated as "MP-AIDEA, $n_{pop} = 1$" in the following.

The ranking method follows the rules of the CEC 2011 competition, (Suganthan 2011). All algorithms are ranked on the basis of the best and mean values of the objective function obtained over a certain number of runs. The following procedure is used to obtain the ranking:

– for each function, algorithms are ranked according to the best objective value;
– for each function, algorithms are ranked according to the mean objective value;
– the ranking for the best and mean objective values of a particular algorithm are added up over all the problems to get the absolute ranking.

In the following, the rankings obtained for the CEC 2005, CEC 2011 and CEC 2014 test sets are presented.

6.2.1 CEC 2005 test set

The rankings obtained for $n_D = 10$, $n_D = 30$ and $n_D = 50$ are reported in Table 12. Only the competing algorithms that reported in their paper also the results obtained for the hybrid functions of the CEC 2005 competition (Table 1) are considered. Results show that, for $n_D = 10$ and $n_D = 30$, MP-AIDEA with adaptation of δ_{local} and n_{LR} is ranked first, while for $n_D = 50$ results are better when using MP-AIDEA with non-adapted $\delta_{local} = 0.1$ and $n_{LR} = 10$. In any case, both settings outperform the winning algorithm of the competition CEC 2005.

Table 9 Objective functions error of the CEC 2014 test set in dimension 10D and 30D

	Best	Worst	Median	Mean	Std
10D					
1	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
2	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
3	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
4	0.00e+00	4.34e+00	0.00e+00	8.50e−02	6.07e−01
5	2.85e−06	2.00e+01	1.31e−05	1.84e+00	5.38e+00
7	0.00e+00	1.23e−02	0.00e+00	2.51e−03	4.02e−03
8	0.00e+00	9.95e−01	0.00e+00	1.37e−01	3.46e−01
9	0.00e+00	3.98e+00	1.99e+00	1.87e+00	9.26e−01
10	0.00e+00	1.19e+02	1.87e−01	9.33e+00	2.79e+01
11	0.00e+00	2.95e+02	3.67e+01	8.82e+01	8.69e+01
13	3.83e−02	1.09e−01	6.52e−02	6.98e−02	1.69e−02
14	1.06e−02	6.40e−02	2.23e−02	2.48e−02	1.03e−02
15	1.97e−02	4.54e−01	3.25e−01	3.10e−01	9.14e−02
16	2.07e−01	2.53e+00	1.42e+00	1.38e+00	5.15e−01
17	0.00e+00	1.43e+02	6.18e+00	2.85e+01	4.53e+01
18	4.90e−03	3.05e+00	6.56e−02	4.29e−01	6.15e−01
20	5.85e−03	2.89e+00	2.23e−01	4.67e−01	5.69e−01
21	1.44e−02	5.87e+01	4.97e−01	4.09e+00	1.39e+01
23	3.29e+02	3.29e+02	3.29e+02	3.29e+02	3.05e−12
24	1.00e+02	1.11e+02	1.06e+02	1.05e+02	3.61e+00
25	1.00e+02	1.19e+02	1.00e+02	1.03e+02	5.07e+00
28	1.01e+02	4.81e+02	3.57e+02	3.47e+02	6.58e+01
30D					
1	0.00e+00	0.00e+00	0.00e+00	0.00e+00	6.88e−14
2	0.00e+00	0.00e+00	0.00e+00	0.00e+00	4.16e−13
3	0.00e+00	0.00e+00	0.00e+00	0.00e+00	2.13e−14
4	0.00e+00	0.00e+00	0.00e+00	0.00e+00	3.64e−13
5	2.00e+01	2.00e+01	2.00e+01	2.00e+01	3.19e−04
7	0.00e+00	0.00e+00	0.00e+00	0.00e+00	1.59e−13
8	0.00e+00	3.98e+00	1.99e+00	2.24e+00	1.14e+00
9	1.09e+01	3.28e+01	2.19e+01	2.26e+01	5.54e+00
10	3.69e+00	1.25e+02	1.07e+01	1.83e+01	3.09e+01
11	6.94e+02	2.34e+03	1.55e+03	1.56e+03	3.70e+02
13	1.25e−01	2.56e−01	1.91e−01	1.91e−01	3.23e−02
14	1.06e−01	2.19e−01	1.47e−01	1.54e−01	2.23e−02
15	1.36e+00	2.98e+00	2.00e+00	2.05e+00	4.06e−01
16	8.40e+00	1.12e+01	1.00e+01	1.00e+01	6.53e−01
17	1.56e+02	9.22e+02	5.21e+02	5.13e+02	1.79e+02
18	1.23e+01	4.70e+01	2.63e+01	2.73e+01	9.35e+00
20	4.29e+00	2.85e+01	1.56e+01	1.57e+01	5.78e+00
21	7.57e+00	5.36e+02	2.38e+02	2.31e+02	9.89e+01
23	3.15e+02	3.15e+02	3.15e+02	3.15e+02	1.35e−10
24	2.00e+02	2.26e+02	2.22e+02	2.19e+02	8.81e+00
25	2.00e+02	2.04e+02	2.03e+02	2.03e+02	6.52e−01
28	6.31e+02	8.56e+02	7.93e+02	7.74e+02	6.20e+01

Table 10 Objective functions error of the CEC 2005 test set in dimension 50D and 100D

	Best	Worst	Median	Mean	Std
50D					
1	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
2	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
3	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
4	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
5	2.00e+01	2.00e+01	2.00e+01	2.00e+01	2.40e−05
7	0.00e+00	0.00e+00	0.00e+00	0.00e+00	6.58e−13
8	2.98e+00	1.09e+01	7.96e+00	7.67e+00	1.84e+00
9	3.68e+01	8.76e+01	5.77e+01	5.83e+01	1.06e+01
10	6.58e+00	2.49e+02	1.75e+01	5.22e+01	6.41e+01
11	2.21e+03	4.86e+03	3.96e+03	3.85e+03	5.21e+02
13	2.07e−01	3.83e−01	3.01e−01	3.08e−01	4.51e−02
14	1.68e−01	2.68e−01	2.32e−01	2.32e−01	2.48e−02
15	3.38e+00	6.31e+00	4.94e+00	4.93e+00	6.68e−01
16	1.78e+01	2.07e+01	1.91e+01	1.91e+01	6.20e−01
17	5.72e+02	1.70e+03	1.07e+03	1.05e+03	2.65e+02
18	4.12e+01	1.40e+02	7.31e+01	7.04e+01	2.07e+01
20	5.10e+01	1.88e+02	9.97e+01	1.02e+02	2.84e+01
21	3.71e+02	1.07e+03	7.79e+02	7.63e+02	1.53e+02
23	3.44e+02	3.44e+02	3.44e+02	3.44e+02	9.45e−08
24	2.52e+02	2.71e+02	2.54e+02	2.56e+02	3.89e+00
25	2.00e+02	2.10e+02	2.07e+02	2.07e+02	1.51e+00
28	1.02e+03	1.25e+03	1.16e+03	1.15e+03	5.45e+01
100D					
1	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
2	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
3	0.00e+00	0.00e+00	0.00e+00	0.00e+00	3.75e−12
4	0.00e+00	3.99e+00	9.32e−12	3.13e−01	1.08e+00
5	2.00e+01	2.00e+01	2.00e+01	2.00e+01	6.35e−06
7	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
8	1.59e+01	4.28e+01	3.08e+01	2.98e+01	5.26e+00
9	1.44e+02	2.10e+02	1.78e+02	1.76e+02	1.83e+01
10	1.29e+02	1.08e+03	4.92e+02	5.24e+02	2.34e+02
11	8.36e+03	1.13e+04	9.92e+03	9.91e+03	6.78e+02
13	3.12e−01	5.14e−01	4.44e−01	4.37e−01	4.11e−02
14	2.58e−01	3.56e−01	3.01e−01	3.04e−01	2.19e−02
15	1.02e+01	2.27e+01	1.63e+01	1.63e+01	2.41e+00
16	3.92e+01	4.35e+01	4.17e+01	4.17e+01	7.96e−01
17	2.09e+03	3.69e+03	2.73e+03	2.78e+03	4.29e+02
18	1.57e+02	2.63e+02	2.09e+02	2.10e+02	3.09e+01
20	2.67e+02	5.98e+02	4.25e+02	4.21e+02	8.30e+01
21	8.88e+02	2.15e+03	1.51e+03	1.53e+03	3.00e+02
23	3.48e+02	3.48e+02	3.48e+02	3.48e+02	1.39e−03
24	3.63e+02	3.80e+02	3.69e+02	3.70e+02	3.25e+00
25	2.00e+02	2.54e+02	2.00e+02	2.14e+02	1.99e+01
28	1.70e+03	2.46e+03	2.23e+03	2.15e+03	2.11e+02

Table 11 CEC 2014 best objective function error values for functions 9, 10, 11 and 15, $n_D = 10$

Algorithm	Func. 9	Func. 10	Func. 11	Func. 15
b3e3pbest (Bujok et al. 2014)	2.60e+00	0.00e+00	9.50e+01	5.70e−01
CMLSP (Chen et al. 2014)	0.00e+00	2.50e−01	3.60e+00	4.50e−01
DE-b6e6rl (Polakova et al. 2014)	2.50e+00	0.00e+00	3.60e+01	4.90e−01
FCDE (Li et al. 2014)	8.00e+00	3.10e−01	1.40e+02	6.50e−01
FERDE (Qu et al. 2014)	3.00e+00	0.00e+00	3.80e−01	3.50e−01
FWA-DM (Yu et al. 2014)	2.00e+00	9.10e−13	4.00e+01	3.20e−01
GaAPADE (Mallipeddi et al. 2014)	1.90e+00	2.40e−02	2.40e+00	3.80e−01
L-SHADE (Tanabe and Fukunaga 2014)	2.20e−03	0.00e+00	3.90e−01	2.10e−01
MVMO (Erlich et al. 2014)	9.90e−01	6.20e−02	3.40e+00	2.10e−01
NRGA (Yashesh et al. 2014)	9.90e−01	3.70e+00	1.90e+01	3.70e−01
OptBees (2014 b)	2.00e+00	3.50e+00	1.30e+02	6.30e−01
POBL-ADE (Hu et al. 2014)	1.00e+00	2.20e+01	3.60e+00	1.70e−01
rmalschma (Molina et al. 2014)	9.90e−01	6.20e−02	1.90e−01	3.10e−01
RSDE (Xu et al. 2014)	2.00e+00	3.50e+00	1.90e+01	3.60e−01
SOO (Preux et al. 2014)	9.00e+00	1.30e+02	3.50e+02	4.40e−01
SOO-BOBYQA (Preux et al. 2014)	9.00e+00	1.30e+02	3.50e+02	4.20e−01
UMOEAs (Elsayed et al. 2014)	9.90e−01	6.20e−02	3.50e+00	3.20e−01
MP-AIDEA	0.00e+00	0.00e+00	0.00e+00	1.97e−02

Table 12 CEC 2005 algorithms ranking

Rank	$n_D = 10$	$n_D = 30$	$n_D = 50$
1	**MP-AIDEA**	**MP-AIDEA**	**MP-AIDEA**, $n_{pop} = 1$
2	**MP-AIDEA**, $n_{pop} = 1$	**MP-AIDEA**, $n_{pop} = 1$	**MP-AIDEA**
3	G-CMA-ES (Auger and Hansen 2005b)	G-CMA-ES	G-CMA-ES
4	L-SaDE (Qin and Suganthan 2005)	L-CMA-ES	L-CMA-ES
5	DMS-L-PSO (Liang and Suganthan 2005)	K-PCX	flexGA
6	L-CMA-ES (Auger and Hansen 2005a)	BLX-GL50	
7	BLX-GL50 (García-Martínez and Lozano 2005)	SPC-PNX	
8	DE (Ronkkonen et al. 2005)	DE (Ronkonnen)	
9	SPC-PNX (Ballester et al. 2005)	DE (Bui)	
10	EvLiv (Becker 2005)	flexGA	
11	EDA (Yuan and Gallagher 2005)	CoEVO	
12	K-PCX (Sinha et al. 2005)	EDA	
13	BLX-MA (Molina et al. 2005)		
14	DE (Bui et al. 2005)		
15	CoEVO (Pošik 2005)		
16	flexGA (Alonso et al. 2005)		
17	ES (Costa 2005)		

6.2.2 CEC 2011 test set

The results obtained on the CEC 2011 test set are reported in Table 13. MP-AIDEA ranks first if problem 13 (the Cassini 2 Spacecraft Trajectory Optimisation Problem) is excluded from the test set and second if it is included.

The reason can be found in Fig. 2. Figure 2 shows the convergence profile of the best solutions found by MP-AIDEA and GA-MPC, the best algorithm of the competition, on function 13 for an increasing number of function evaluations (greater than the limit prescribed by the CEC 2011 competition). The results for GA-MPC are obtained using the code available online (http://www3.ntu.edu.sg/home/epnsugan/index_files/CEC11-RWP/CEC11-RWP.htm).

On this test problem, GA-MPC converges very rapidly to a local minimum but then stagnates. On the contrary,

Table 13 CEC 2011 algorithms ranking

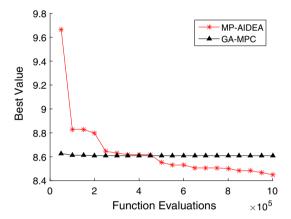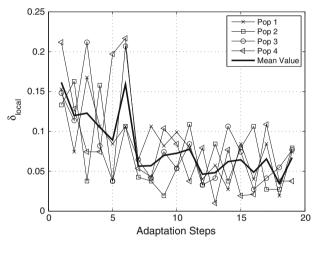| Rank	With function 13	Without function 13
1	GA-MPC (Elsayed et al. 2011b)	**MP-AIDEA**
3	**MP-AIDEA**	GA-MPC
4	SAMODE (Elsayed et al. 2011a)	EA-DE-MA
5	EA-DE-MA (Singh and Ray 2011)	SAMODE
6	WI-DE (Haider et al. 2011)	WI-DE
7	Adap. DE 171 (Asafuddoula et al. 2011)	**MP-AIDEA**, $n_{\mathrm{pop}} = 1$
8	**MP-AIDEA**, $n_{\mathrm{pop}} = 1$	ED-DE
10	DE-Λ (Reynoso-Meza et al. 2011)	DE-Λ
11	ED-DE (Wang et al. 2011)	Adapt. DE 171
12	DE-RHC (LaTorre et al. 2011)	DE-RHC
13	RGA (Saha and Ray 2011)	RGA
14	Mod-DE-LS (Mandal et al. 2011)	Mod-DE-LS
15	mSBX-GA (Bandaru 2011)	mSBX-GA
16	ENSML-DE (Mallipeddi and Suganthan 2011)	CDASA
17	CDASA (Korošec and Šilc 2011)	ENSML-DE

Fig. 2 Best values of MP-AIDEA and GA-MPC for Function 13, CEC2011

MP-AIDEA has a slower convergence for the first 200,000 function evaluations but then progressively finds better and better minima as the number of function evaluations increases. This demonstrates that in a realistic scenario in which function evaluations are not arbitrarily limited, MP-AIDEA would provide better results than the algorithm that won the competition.

Results in Table 13 shows that MP-AIDEA with adaptation of δ_{local} and n_{LR} performs better than MP-AIDEA with fixed values of δ_{local} and n_{LR}. The adaptation history of δ_{local} is shown in Fig. 3 for each of the four populations on test functions 12 and 13 and for 600,000 function evaluations.

6.2.3 CEC 2014 test set

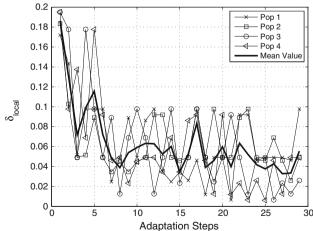The ranking results for the CEC 2014 test set are reported in Table 14. MP-AIDEA with one population is tested in this

Fig. 3 δ_{local} for the four populations of MP-AIDEA for functions 12 (top) and 13 (bottom), CEC 2011
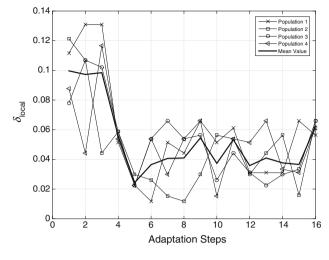
Table 14 CEC 2014 algorithms ranking

Rank	$n_D = 10$	$n_D = 30$
1	UMOEAs (Elsayed et al. 2014)	L-SHADE
2	**MP-AIDEA**	UMOEAs
3	L-SHADE (Tanabe and Fukunaga 2014)	GaPADE
4	MVMO (Erlich et al. 2014)	**MP-AIDEA** $n_{pop} = 1, \delta_{local} = 0.1$
5	**MP-AIDEA,** $n_{pop} = 1, \delta_{local} = 0.1$	**MP-AIDEA**
6	DE-b6e6rl (Polakova et al. 2014)	CMLSP
7	rmalschma (Molina et al. 2014)	**MP-AIDEA** $n_{pop} = 1, \delta_{local} = 0.3$
8	**MP-AIDEA,** $n_{pop} = 1, \delta_{local} = 0.3$	rmalshcma
9	GaAPADE (Mallipeddi et al. 2014)	MVMO
10	FERDE (Qu et al. 2014)	DE-b6e6rl
11	CMLSP (Chen et al. 2014)	b3e3pbest
12	b3e3pbest (Bujok et al. 2014)	FERDE
13	RSDE (Xu et al. 2014)	RSDE
14	FWA-DE (Yu et al. 2014)	FWA-DE
15	POBL-ADE (Hu et al. 2014)	POBL-ADE
16	OptBees (2014 b)	OptBees
17	SOO-BOBYQA (Preux et al. 2014)	SOO-BOBYQA
18	FCDE (Li et al. 2014)	NRGA
19	NRGA (Yashesh et al. 2014)	FCDE
20	SOO (Preux et al. 2014)	SOO

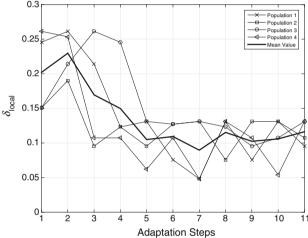| Rank	$n_D = 50$	$n_D = 100$
1	**MP-AIDEA** $n_{pop} = 1, \delta_{local} = 0.1$	**MP-AIDEA** $n_{pop} = 1, \delta_{local} = 0.1$
2	UMOEAs	UMOEAs
3	MVMO	L-SHADE
4	**MP-AIDEA**	**MP-AIDEA**
5	L-SHADE	rmalshcma
6	**MP-AIDEA** $n_{pop} = 1, \delta_{local} = 0.3$	**MP-AIDEA** $n_{pop} = 1, \delta_{local} = 0.3$
7	rmalshcma	POBL-ADE
8	b3e3pbest	b3e3pbest
9	FERDE	OptBees
10	DE-b6e6rl	DE-b6e6rl
11	RSDE	RSDE
12	POBL-ADE	FWA-DE
13	OptBees	
14	FWA-DE	
15	SOO	

case with $\delta_{local} = 0.1$ and $\delta_{local} = 0.3$. For $n_D = 10$ the results of MP-AIDEA with adaptation of δ_{local} and n_{LR} are better than those of MP-AIDEA with fixed values of δ_{local} and n_{LR}, for both $\delta_{local} = 0.1$ and $\delta_{local} = 0.3$. In the other cases MP-AIDEA with fixed values of δ_{local} and n_{LR} outperforms MP-AIDEA with adaptation of δ_{local} and n_{LR} when $\delta_{local} = 0.1$ but not when $\delta_{local} = 0.3$. These results show the strong influence of this parameter on the results obtained by MP-AIDEA. The adaptation history of δ_{local} for test functions 9,

17 and 25 at $n_D = 30$ and 300,000 functions evaluations is shown in Fig. 4.

These figures show how the adaptation of δ_{local} is effective when a sufficient number of adaptation steps can be performed within the limit of the maximum number of function evaluation (300,000 in this case). For function 25, for example, the adaptation steps are only 7, while they are 11 for function 17 and 18 for function 9. In these two cases δ_{local} converges to 0.1 and 0.04, respectively.
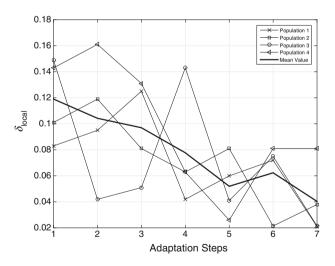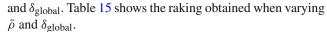
Fig. 4 δ_{local} for the four populations of MP-AIDEA for functions 9 (top), 17 (middle) and 25 (bottom), $n_{\text{D}} = 30$, CEC 2014

The performance of MP-AIDEA for the 30D functions of the CEC 2014 test set is further investigated to test the dependence of the results upon the two non-adapted parameters, $\bar{\rho}$

and δ_{global}. Table 15 shows the raking obtained when varying $\bar{\rho}$ and δ_{global}.

Case B of Table 15 shows the ranking obtained when using $\bar{\rho} = 0.3$ instead than $\bar{\rho} = 0.2$. Comparing the results in Table 15 with those in Table 14, it is possible to see that MP-AIDEA performs better using $\bar{\rho} = 0.3$ rather than $\bar{\rho} = 0.2$, moving from the fourth to the third position in the ranking. At the same time, there is no significant dependence upon the value of δ_{global}, as shown by Cases C and D in Table 15, where δ_{global} is changed from its nominal value of 0.1 to 0.2 and 0.3.

6.3 Wilcoxon test

The Wilcoxon rank sum test is a nonparametric test for two populations when samples are independent. In this case, the two populations of samples are, for each problem, the n_{runs} values of the objective function obtained by MP-AIDEA and by another algorithms participating in the CEC 2011 and CEC 2014 competitions. No test is performed for the CEC2005 test set, since for no one of the algorithms participating in the CEC 2005 competition the code is available on-line.

The Wilcoxon test is realised using the Matlab® function *ranksum*. *ranksum* tests the null hypothesis that data from two entries x and y are samples from continuous distributions with equal medians. Results from *ranksum* are presented in the following as values of p and h. p, ranging from 0 to 1, is the probability of observing a test statistic as or more extreme than the observed value under the null hypothesis. h is a logical value, where $h = 1$ indicates rejection of the null hypothesis at the 100α % significance level while $h = 0$ indicates a failure to reject the null hypothesis at the 100α % significance level, where α is 0.05. When $h = 1$, the null hypothesis that distributions x and y have equal medians is rejected, and additional test are conducted to assess which one of the two distributions has lower median. In order to do so, three types of tests are realised using *ranksum* for the two distributions x and y:

– Two-sided hypothesis test: the alternative hypothesis states that x and y have different medians. Two distributions with equal medians will give as results $p_B = 1$ and $h_B = 0$ (failure to reject the null hypothesis that x and y have equal medians), while two distributions with different medians will give as results $p_B = 0$ and $h_B = 1$ (rejection of the null hypothesis that x and y have equal medians). If the two-sided hypothesis test finds that the two distributions have equal medians ($p_B = 1$ and $h_B = 0$), no further test is conducted. Otherwise, the left-tailed and right-tailed hypothesis test are conducted.
– Left-tailed hypothesis test: the alternative hypothesis states that the median of x is lower than the median of y.

Table 15 CEC 2014 algorithms ranking, 30D, $\bar{\rho} = 0.1$ and $\bar{\rho} = 0.3$

Rank	Case A $\bar{\rho} = 0.1$ $\delta_{\text{global}} = 0.1$	Case B $\bar{\rho} = 0.3$ $\delta_{\text{global}} = 0.1$	Case C $\bar{\rho} = 0.2$ $\delta_{\text{global}} = 0.2$	Case D $\bar{\rho} = 0.2$ $\delta_{\text{global}} = 0.3$
1	L-SHADE	L-SHADE	L-SHADE	L-SHADE
2	UMOEAs	UMOEAs	UMOEAs	UMOEAs
3	GaAPADE	**MP-AIDEA**	GaAPADE	GaAPADE
4	**MP-AIDEA**	GaAPADE	**MP-AIDEA**	**MP-AIDEA**
5	CMLSP	CMLSP	CMLSP	CMLSP
6	rmalshcma	rmalschma	rmalshcma	rmalshcma
7	MVMO	MVMO	MVMO	MVMO
8	DE-b6e6rl	DE-b6e6rl	DE-b6e6rl	DE-b6e6rl
9	b3e3pbest	b3e3pbest	b3e3pbest	b3e3pbest
10	FERDE	FERDE	FERDE	FERDE
11	RSDE	RSDE	RSDE	RSDE
12	FWA-DE	FWA-DE	FWA-DE	FWA-DE
13	POBL-ADE	POBL-ADE	POBL-ADE	POBL-ADE
14	OptBees	OptBees	OptBees	OptBees
15	SOO-BOBYQA	SOO-BOBYQA	SOO-BOBYQA	SOO-BOBYQA
16	NRGA	NRGA	NRGA	NRGA
17	FCDE	FCDE	FCDE	FCDE
18	SOO	SOO	SOO	SOO

If x has median greater than the median of y, results will be $p_L = 1$ and $h_L = 0$ (failure to reject the hypothesis that x has median greater than y) while if x has median lower than y results will be $p_L = 0$ and $h_L = 1$ (rejection of the hypothesis that x has median greater than y).

– Right-tailed hypothesis test: the alternative hypothesis states that the median of x is greater than the median of y. If x has median lower than the median of y, results will be $p_R = 1$ and $h_R = 0$ (failure to reject the hypothesis that x has median lower than y) while if x has median greater than y results will be $p_R = 0$ and $h_R = 1$ (rejection of the hypothesis that x has median lower than y).

If x is the distribution of results of MP-AIDEA and y the distribution of results given by another algorithm, the possible results obtained from the *ranksum* tests are summarised in Table 16.

Case 1 in Table 16 ($h_B = 0$) represents a situation in which the distribution of results from MP-AIDEA and a competing algorithm have equal median (failure to reject the hypothesis that x has median lower than y). Case 2 ($h_B = 1$, $h_L = 0$ and $h_R = 1$) represents a situation in which the median of MP-AIDEA is greater than the median of the other algorithm (rejection of the null hypothesis that x and y have equal medians, failure to reject the hypothesis that x has median greater than y, rejection of the hypothesis that x has median lower than y). Case 3 ($h_B = 1$, $h_L = 1$ and $h_R = 0$) represents instead a situation in which the median of MP-AIDEA is

lower than the median of the other algorithm (rejection of the null hypothesis that x and y have equal medians, rejection of the hypothesis that x has median greater than y, failure to reject the hypothesis that x has median lower than y). In the following, test functions with results corresponding to cases 1 and 3 are shown in bold (MP-AIDEA has median equal or lower than the competing algorithm). For case 3 results with $p_B < 5 \cdot 10^{-2}$, $p_L < 5 \cdot 10^{-2}$ and $p_R > 9.5 \cdot 10^{-1}$ are considered significant. Analogously, the competing algorithm has median lower than MP-AIDEA if $p_B < 5 \cdot 10^{-2}$, $p_L > 9.5 \cdot 10^{-1}$ and $p_R < 5 \cdot 10^{-2}$.

6.3.1 CEC 2011 test set

For the CEC 2011 test set, we limited the comparison against the two top algorithms GA-MPC and DE-Λ, for which the code is available online (http://www3.ntu.edu.sg/home/epnsugan/index_files/CEC11-RWP/CEC11-RWP.htm; http://uk.mathworks.com/matlabcentral/fileexchange/39217-hybrid-differential-evolution-algorithm-with-adaptive-crossover-mechanism/content/DE_TCRparam.m). The outcome of the Wilcoxon test for the comparison of MP-AIDEA against GA-MPC, the winning algorithm of the CEC2011 competition, can be found in Table 17 for all the functions in the test set in Table 2.

The comparison of MP-AIDEA with GA-MPC shows that the median of MP-AIDEA is lower than the median of GA-MPC (Case 3) for functions 2, 5, 6 and 7, while it is higher

Table 16 Wilcoxon test: possible outcomes

	Both		Left		Right	
	h_B	p_B	h_L	p_L	h_R	p_R
Case 1: equal medians	0	1	–	–	–	–
Case 2: median of MP-AIDEA is greater	1	0	0	1	1	0
Case 3: median of MP-AIDEA is lower	1	0	1	0	0	1

Table 17 Outcome of the Wilcoxon test on the CEC 2011 test set: MP-AIDEA versus GA-MPC

Func.	Both		Left		Right		Result type (Table 16)
	h	p	h	p	h	p	
1	1	1.28e−04	0	1.00e+00	1	6.40e−05	Case 2
2	**1**	**1.43e-04**	**1**	**7.14e-05**	**0**	**1.00e+00**	**Case 3**
3	1	1.10e−05	0	1.00e+00	1	5.49e−06	Case 2
5	**1**	**5.12e-06**	**1**	**2.56e-06**	**0**	**1.00e+00**	**Case 3**
6	**1**	**4.78e-02**	**1**	**2.39e-02**	**0**	**9.77e-01**	**Case 3**
7	**1**	**3.01e-09**	**1**	**1.50e-09**	**0**	**1.00e+00**	**Case 3**
10	0	3.62e−01	0	8.24e−01	0	1.81e−01	Not significant
12	0	4.85e−01	0	2.42e−01	0	7.64e−01	Not significant
13	1	4.61e−03	0	9.98e−01	1	2.31e−03	Case 2

Table 18 Outcome of the Wilcoxon test on the CEC 2011 test set: MP-AIDEA versus DE-Λ

Func	Both		Left		Right		Result type (Table 16)
	h	p	h	p	h	p	
1	0	7.58e−02	0	9.64e−01	1	3.79e−02	Not significant
2	0	4.72e−01	0	2.36e−01	0	7.70e−01	Not significant
3	**1**	**9.73e-11**	**1**	**4.86e-11**	**0**	**1.00e+00**	**Case 3**
5	**1**	**8.52e-08**	**1**	**4.26e-08**	**0**	**1.00e+00**	**Case 3**
6	**1**	**1.41e-09**	**1**	**7.07e-10**	**0**	**1.00e+00**	**Case 3**
7	0	5.05e−02	1	2.52e−02	0	9.76e−01	Not significant
10	**1**	**1.18e-07**	**1**	**5.89e-08**	**0**	**1.00e+00**	**Case 3**
12	**1**	**2.57e-09**	**1**	**1.29e-09**	**0**	**1.00e+00**	**Case 3**
13	**1**	**2.04e-03**	**1**	**1.02e-03**	**0**	**9.99e-01**	**Case 3**

(Case 2) for functions 1, 3 and 13. Results for functions 10 and 12 are not significant enough to obtain a clear indication.

The outcome of the Wilcoxon test for the comparison of MP-AIDEA with DE-Λ is reported in Table 18.

The comparison of MP-AIDEA with DE-Λ (Table 18) shows that the median of MP-AIDEA is lower than the median of DE-Λ for functions 3, 5, 6, 10, 12 and 13. Results for the remaining functions 1, 2 and 7 are not significant enough to obtain a clear indication.

Table 19 summarises the outcome of the Wilcoxon tests for the CEC 2011 test set. The table reports the number of functions for which the median of MP-AIDEA is lower, equal or higher than the median of the competing algorithm. The results in Table 19 show that MP-AIDEA clearly outperforms DE-Λ and has median lower than GA-MPC for 4 test functions.

Table 19 Summary of Wilcoxon test results, CEC 2011 test set: MP-AIDEA versus GA-MPC and DE-Λ. The table reports the number of functions for which the median of MP-AIDEA is equal (Case 1), higher (Case 2) or lower (Case 3) than the median of the competing algorithm

	GA-MPC	DE-Λ
Case 1: equal medians	0	0
Case 2: median of MP-AIDEA is greater	3	0
Case 3: median of MP-AIDEA is lower	4	6
Not significant	2	3

6.3.2 CEC 2014 test set

Codes for the algorithms UMOEAs, CLMSP, L-SHADE and MVMO are avilable online (http://web.mysites.ntu.edu.sg/epnsugan/PublicSite/Shared%20Documents/Forms/AllItems.aspx). Wilcoxon test results for the comparison of MP-AIDEA with these algorithms at 10, 30, 50 and 100 dimen-

Table 20 Summary of Wilcoxon test results, CEC 2014. The table reports the number of functions for which the median of MP-AIDEA is equal (Case 1), higher (Case 2) or lower (Case 3) than the median of the competing algorithm

	$n_D = 10$	$n_D = 30$	$n_D = 50$	$n_D = 100$
UMOEAs				
Case 1	3	0	0	0
Case 2	4	9	7	9
Case 3	11	9	11	9
Not significant	4	4	4	4
L-SHADE				
Case 1	3	0	0	0
Case 2	8	17	13	10
Case 3	9	5	8	9
Not significant	2	0	1	3
MVMO				
Case 1	0	0	0	0
Case 2	3	7	11	9
Case 3	13	10	8	10
Not significant	6	5	3	3
CMLSP				
Case 1	0	0	0	0
Case 2	1	9	2	2
Case 3	21	13	20	20
Not significant	0	0	0	0

Table 21 Success rate: CEC2011 test set. Highest success rates for each function are shown in bold, and their total is reported at the bottom of the table. MP-AIDEA* represents MP-AIDEA with settings $n_{pop} = 1$, $\delta_{local} = 0.1$ and $n_{LR} = 10$

	tol_f	MP-AIDEA	MP-AIDEA*	GA-MPC	DE-Λ
1	1.0e−01	**0.92**	0.48	0.80	0.64
2	1.0e−01	**0.40**	0.20	0.12	**0.40**
3	1.0e−06	**1.00**	**1.00**	**1.00**	**1.00**
5	1.0e−01	**0.44**	0.20	0.16	0.04
6	1.0e+01	0.76	0.76	**0.84**	0.00
7	1.0e−01	**0.92**	0.64	0.04	0.72
10	1.0e−01	**0.36**	0.16	0.24	0.00
12	2.0e+00	**0.24**	0.00	0.20	0.00
13	1.0e+00	0.16	0.04	**0.52**	0.00
Total		7	1	3	2

In all the cases, MP-AIDEA has median lower than CMLSP for the majority of the tested functions.

Summarizing, results of the Wilcoxon test show that MP-AIDEA clearly outperforms CMLSP for all the values of n_D, gives similar or slightly better results than UMOEAs and MVMO while is outperformed by L-SHADE for $n_D = 30$, $n_D = 50$ and $n_D = 100$.

6.4 Success rate

In this section, we present the success rate of MP-AIDEA and the top performing algorithms on the test sets CEC 2011 and CEC 2014. As for the Wilcoxon test no algorithm participating in the CEC 2005 was included in the comparison due to the lack of availability of the source code.

The computation of the success rate SR is reported in Algorithm 5 for a generic algorithm AG and a generic problem $\min_f$ where n is the number of runs (Vasile et al. 2011). In Algorithm 5, $\bar{\mathbf{x}}(AG, i)$ denotes the lowest minimum observed during the i-th run of the algorithm AG. The quantity f_{global} is the known global minimum of the function and tol_f is a prescribed tolerance with respect to f_{global}. The index j_{sr} represents the number of times algorithm AG generates values lower or equal than $f_{global} + tol_f$. For each test set, we report also the total number of problems in which each of the tested algorithms has the best success rate.

6.4.1 CEC 2011 test set

For the calculation of the success rate on the test set CEC 2011, we consider the following algorithms: MP-AIDEA with 4 populations (MP-AIDEA), adaptive δ_{local} and local restart, MP-AIDEA with one population, $n_{LR} = 10$ and $\delta_{local} = 0.1$ (MP-AIDEA*), GA-MPC and DE-Λ. Table 21 shows the obtained values of SR and the value of tol_f used for

sions are reported in Appendix A (Tables 24, 25, 26, 27, 28, 29, 30, 31).

A summary of the obtained results is given in Table 20. Table 20 shows the number of function for which Case 1, 2 or 3 in Table 16 are verified and the number of functions for which the results are not significant enough to judge, for n_D equal to 10, 30, 50 and 100.

For $n_D = 10$, the median of MP-AIDEA is lower than the one of UMOEAs in 11 cases, while in 3 cases the medians are equal and in 4 cases the median of UMOEAs is lower than the median of MP-AIDEA. In 4 cases (functions 10, 17, 20 and 21), the results are not significant enough. For $n_D = 30$ and $n_D = 100$ the median of MP-AIDEA is lower than the median of UMOEAs in 9 cases and the median of UMOEAs is lower than the one of MP-AIDEA for other 9 functions. For 4 functions, the results are not significant enough to obtain a clear indication. The median of MP-AIDEA is lower than the one of UMOEAs in 11 cases for $n_D = 50$.

As regards the comparison with L-SHADE, MP-AIDEA has lower median for a number of functions greater than L-SHADE only for $n_D = 10$ (9 functions).

In all dimension but $n_D = 50$, the number of functions for which the median of MP-AIDEA is lower than the median of MVMO is greater than the number of functions for which the median of MVMO is lower than the median of MP-AIDEA.

Table 22 Success rate: CEC 2014, 10D and 30D

	tol$_f$	MP-AIDEA	MP-AIDEA*	UMOEAs	L-SHADE	MVMO	CMLSP
$n_D = 10$							
1	1.0e−06	**1.00**	**1.00**	**1.00**	**1.00**	0.00	0.00
2	1.0e−06	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.00
3	1.0e−06	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.00
4	1.0e−01	**0.98**	0.78	0.51	0.18	0.67	0.53
5	1.0e−01	**0.75**	0.33	0.06	0.04	0.12	0.00
7	1.0e−02	0.98	0.92	**1.00**	0.90	0.39	0.16
8	1.0e−01	0.86	**1.00**	0.98	**1.00**	**1.00**	0.00
9	1.0e+00	**0.37**	0.20	0.27	0.06	0.14	0.00
10	1.0e−01	0.33	0.00	0.18	**0.98**	0.00	0.00
11	1.0e+00	**0.06**	**0.06**	0.00	0.00	0.00	0.00
13	1.0e−01	0.92	0.53	**1.00**	**1.00**	**1.00**	0.24
14	1.0e−01	**1.00**	**1.00**	0.55	0.84	0.69	0.06
15	1.0e−01	**0.04**	0.00	0.00	0.00	0.00	0.00
16	1.0e+00	0.24	0.20	0.14	**0.29**	0.16	0.00
17	1.0e+01	0.53	0.45	0.37	**0.98**	0.57	0.10
18	1.0e+00	0.71	0.39	0.45	**0.96**	0.33	0.18
20	1.0e+00	0.78	0.98	0.88	**1.00**	0.96	0.18
21	1.0e+01	0.92	0.98	0.90	**1.00**	0.92	0.63
23	2.0e+02	0.00	0.00	0.00	0.00	0.00	0.00
24	2.0e+02	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
25	2.0e+02	**1.00**	0.98	0.86	0.75	0.96	0.96
28	2.0e+02	**0.06**	0.00	0.04	0.00	0.00	0.02
Total		12	7	6	12	5	1
$n_D = 30$							
1	1.0e−06	**1.00**	**1.00**	0.96	**1.00**	0.00	0.00
2	1.0e−06	**1.00**	**1.00**	**1.00**	**1.00**	0.18	0.00
3	1.0e−06	**1.00**	**1.00**	**1.00**	**1.00**	0.02	0.00
4	1.0e−06	**1.00**	**1.00**	0.86	**1.00**	**1.00**	0.00
5	2.0e+01	**1.00**	**1.00**	0.37	0.00	**1.00**	0.00
7	1.0e−04	**1.00**	**1.00**	**1.00**	**1.00**	0.73	0.86
8	1.0e−01	0.04	**1.00**	0.10	**1.00**	**1.00**	0.00
9	1.0e+01	0.00	0.04	0.47	**0.94**	0.00	0.02
10	1.0e−01	0.00	0.98	0.00	**1.00**	0.00	0.00
11	1.0e+03	0.08	**0.22**	**0.22**	0.12	0.02	0.00
13	1.0e−01	0.00	0.67	**0.92**	0.12	0.00	0.00
14	2.0e−01	**0.96**	0.53	0.41	0.12	0.76	0.00
15	2.0e+00	**0.47**	0.39	0.04	0.24	0.10	0.00
16	1.0e+01	0.47	0.65	0.14	**1.00**	0.61	0.00
17	5.0e+02	0.47	0.94	0.00	**1.00**	0.98	**1.00**
18	2.0e+01	0.25	0.33	0.43	**1.00**	0.75	0.75
20	2.0e+01	0.76	**1.00**	0.90	**1.00**	0.96	**1.00**
21	1.0e+01	0.02	**0.98**	0.00	0.25	0.04	0.04

Table 22 continued

	tol$_f$	MP-AIDEA	MP-AIDEA*	UMOEAs	L-SHADE	MVMO	CMLSP
23	2.0e+02	0.00	0.00	0.00	0.00	0.00	0.00
24	2.2e+02	0.18	**0.76**	0.04	0.00	0.00	0.41
25	2.0e+02	0.00	0.00	0.00	0.00	0.00	0.00
28	7.0e+02	0.20	0.10	0.04	0.00	0.00	**0.63**
Total		8	11	5	12	3	3

Highest success rates for each function are shown in bold. and their total is reported at the bottom of the table for each value of n_D. MP-AIDEA* represents MP-AIDEA with settings $n_{pop} = 1$ and $\delta_{local} = 0.1$

Table 23 Success rate: CEC 2014, 50D and 100D

	tol$_f$	MP-AIDEA	MP-AIDEA*	UMOEAs	L-SHADE	MVMO	CMLSP
$n_D = 50$							
1	1.0e−06	**1.00**	**1.00**	**1.00**	0.00	0.00	0.00
2	1.0e−06	**1.00**	**1.00**	**1.00**	**1.00**	0.00	0.00
3	1.0e−06	**1.00**	**1.00**	0.80	**1.00**	0.00	0.00
4	1.0e−06	**1.00**	**1.00**	0.18	0.00	0.98	0.00
5	2.0e+01	**1.00**	**1.00**	0.35	0.00	**1.00**	0.00
7	1.0e−04	**1.00**	**1.00**	**1.00**	**1.00**	0.41	0.00
8	5.0e+00	0.18	**1.00**	0.51	**1.00**	0.94	0.00
9	5.0e+01	0.18	0.69	**1.00**	**1.00**	0.10	0.00
10	1.0e+01	0.12	0.98	0.16	**1.00**	0.10	0.00
11	3.0e+03	0.06	**0.78**	0.10	0.16	0.12	0.00
13	3.0e−01	0.49	0.65	**1.00**	**1.00**	0.94	0.20
14	2.0e−01	**0.10**	0.04	0.02	0.00	0.08	0.00
15	5.0e+00	0.57	**0.94**	0.14	0.39	0.49	0.00
16	1.8e+01	0.02	0.16	0.06	**1.00**	0.08	0.00
17	1.0e+03	0.41	**0.98**	0.00	0.12	0.88	0.04
18	1.0e+02	0.94	**0.96**	0.45	0.63	**0.96**	0.22
20	1.0e+02	0.51	0.96	0.63	**1.00**	**1.00**	**1.00**
21	1.0e+03	0.96	**1.00**	0.10	**1.00**	**1.00**	0.22
23	3.4e+02	0.00	0.00	0.00	0.00	0.00	0.00
24	2.6e+02	**0.90**	**0.90**	0.02	0.00	0.86	0.00
25	2.1e+02	0.02	**0.18**	0.02	0.12	0.08	0.02
28	1.1e+03	0.20	**0.37**	0.04	0.27	0.00	0.33
Total		8	15	5	10	4	1
$n_D = 100$							
1	1.0e−06	**1.00**	**1.00**	**1.00**	0.00	0.00	0.00
2	1.0e−06	**1.00**	**1.00**	**1.00**	**1.00**	0.00	0.00
3	1.0e−06	**1.00**	**1.00**	0.41	1.00	0.00	0.00
4	1.0e−06	**0.92**	0.78	0.00	0.00	0.35	0.00
5	2.0e+01	**1.00**	**1.00**	0.63	0.00	**1.00**	0.00
7	1.0e−04	**1.00**	**1.00**	**1.00**	**1.00**	0.82	0.00
8	5.0e+00	0.00	0.98	0.00	**1.00**	0.04	0.00
9	2.0e+02	0.88	**1.00**	**1.00**	**1.00**	0.61	0.00
10	5.0e+02	0.51	**1.00**	0.20	**1.00**	0.08	0.00

Table 23 continued

	tol_f	MP-AIDEA	MP-AIDEA*	UMOEAs	L-SHADE	MVMO	CMLSP
11	1.0e+04	0.55	**1.00**	0.59	0.06	0.57	0.00
13	4.0e−01	0.16	0.12	**1.00**	**1.00**	**1.00**	0.00
14	3.0e−01	0.45	0.16	**1.00**	**1.00**	**1.00**	0.90
15	2.0e+01	0.92	**1.00**	**1.00**	**1.00**	0.90	0.00
16	4.0e+01	0.02	0.12	0.00	**0.96**	0.00	0.00
17	3.0e+03	0.67	**0.84**	0.00	0.04	0.39	0.00
18	2.0e+02	0.41	**1.00**	0.00	0.10	**1.00**	0.00
20	3.0e+02	0.08	0.98	0.37	**1.00**	**1.00**	**1.00**
21	1.0e+03	0.02	**0.98**	0.00	0.00	0.18	0.00
23	3.7e+02	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.98
24	3.7e+02	0.57	0.04	0.00	0.00	**0.98**	0.00
25	2.5e+02	0.96	**1.00**	**1.00**	**1.00**	0.22	**1.00**
28	2.0e+03	**0.27**	**0.27**	0.00	0.00	0.00	0.04
Total		7	15	9	12	7	2

Highest success rates for each function are shown in bold and their total is reported at the bottom of the table for each value of n_D. MP-AIDEA* represents MP-AIDEA with settings $n_{pop} = 1$ and $\delta_{local} = 0.1$

Algorithm 8 Calculation of the success rate

1: Apply AG to min_f for n_{runs} times and set $j_{sr} = 0$
2: **for** $i \in [1, \ldots n_{runs}]$ **do**
3:   Compute $\delta_f = \| f_{global} - f(\bar{\mathbf{x}}(AG, i)) \|$
4:   **if** $\delta_f < tol_f$ **then**
5:     $j_{sr} = j_{sr} + 1$
6:   **end if**
7: **end for**
8: $SR = j_{sr}/n_{runs}$

---

each function and shows that MP-AIDEA outperforms all the other algorithms on most of the functions. The result against GA-MPC would be even better if a higher number of function evaluation was considered, as explained in Sect. 6.2.2.

6.4.2 CEC 2014 test set

For the comparison on the test set CEC 2014, we considered the following algorithms: MP-AIDEA, MP-AIDEA*, UMOEAs, CLMSP, L-SHADE and MVMO (http://web. mysites.ntu.edu.sg/epnsugan/PublicSite/Shared%20Docum ents/Forms/AllItems.aspx). The values of the success rates for all tested algorithms are shown in Tables 22 and 23, together with the associated values of tol_f. The total number of problems for which an algorithm yields the best success rate is also reported.

For all dimensions, MP-AIDEA compares very well against the other algorithms. In low dimension, the full adaptive settings is the most competitive while as the number of dimensions increases the single population version with $\delta_{local} = 0.1$ results the most successful algorithm. These results are in line with the results in Sect. 6.2.3 and confirm the position of MP-AIDEA in the ranking.

7 Conclusions

This paper presented MP-AIDEA, an adaptive version of inflationary differential evolution which automatically adapts the two key parameters of differential evolution, CR, F, the size of the restart bubble δ_{local} and the number of local restarts n_{LR}. The adaptation of the number of local restarts is implemented through a mechanism that mitigates the possibility to detect the same local minimum multiple times. This mechanism allows MP-AIDEA to automatically identify when to switch from a local to a global restart of the population.

MP-AIDEA was tested on a total of 51 problems, taken from three CEC competitions, grouped in three test sets (named CEC 2015, CEC 2011 and CEC 2014) and compared against 53 algorithms that participated in those three competitions. Four different metrics were presented to assess the performance of MP-AIDEA. Results demonstrated that MP-AIDEA ranks first in the CEC 2015 outperforming all the other algorithms for all problem dimensionalities. On the CEC 2011 test set, MP-AIDEA ranks second, after GA-MPC, if we restrict the number of function evaluations to the one prescribed by the competition. However, it was demonstrated that, in problem 13, an increase in the number of function evaluations does not provide any improvement of the objective value returned by GA-MPC but greatly improves the result of MP-AIDEA. It was noted, in fact, that GA-MPC has a fast convergence but then tends to stagnate. On the contrary, the convergence profile of MP-AIDEA is slower but, thanks to the restart mechanism, achieves better objective values. In this test set, in particular, the adaptation of the local restart neighbourhood was shown to be effective providing compet-

itive results compared to the settings of MP-AIDEA with a single population and predefined values of δ_{local} and number of restarts. This is confirmed by the Wilcoxon test, and the success rate.

On the test set CEC 2014, results are not equally satisfactory for all dimensions. MP-AIDEA is in the top three algorithms except in dimension 30. When the number of populations is reduced to one and $\delta_{\text{local}} = 1$, MP-AIDEA outperforms all other algorithms in dimension 50 and 100.

One part of the problem is the extra effort required by the multi-population adaptive algorithm to identify the correct value of δ_{local}. However, another part of the problem was found in the contraction limit. This is in line with the theoretical findings by the authors who demonstrated that DE can converge to a level set in the general case. Furthermore, it was noted that the populations can naturally partition and form clusters that independently converge to separate points. This slow rate of convergence affects the restart and local search mechanisms and the associated adaptation machinery. Since the current implementation uses a synchronous restart and adaptation of δ_{local} and n_{LR}, the number of restarts might be limited by the fact that the evolution of all populations has to come to a stop before any of them can be restarted. Future work will be dedicated to improve these aspects of the algorithm.

Compliance with ethical standards

A Wilcoxon test results

The following tables report the results of the Wilcoxon test for the comparison of MP-AIDEA with UMOEAs (Tables 24, 25), L-SHADE (Tables 26, 27), MVMO (Tables 28, 29) and CMLSP (Tables 30, 31).

Table 24 Wilcoxon test results, CEC 2014: MP-AIDEA versus UMOEAs, 10D and 30D

	Both		Left		Right		Result type
	h	p	h	p	h	p	
$n_{\text{D}} = 10$							
1	**0**	**1.00e+00**	–	–	–	–	**Case 1**
2	**0**	**1.00e+00**	–	–	–	–	**Case 1**
3	**0**	**1.00e+00**	–	–	–	–	**Case 1**
4	**1**	**2.18e−02**	**1**	**1.09e−02**	**0**	**9.89e−01**	**Case 3**
5	**1**	**3.74e−14**	**1**	**1.87e−14**	**0**	**1.00e+00**	**Case 3**
7	1	9.89e−16	0	1.00e+00	1	4.95e−16	Case 2
8	1	1.79e−02	0	9.91e−01	1	8.94e−03	Case 2
9	**1**	**7.80e−04**	**1**	**3.90e−04**	**0**	**1.00e+00**	**Case 3**
10	0	3.17e−01	0	8.43e−01	0	1.59e−01	Not sign
11	**1**	**1.63e−05**	**1**	**8.16e−06**	**0**	**1.00e+00**	**Case 3**
13	1	8.19e−16	0	1.00e+00	1	4.10e−16	Case 2
14	**1**	**1.20e−17**	**1**	**5.98e−18**	**0**	**1.00e+00**	**Case 3**
15	**1**	**3.30e−18**	**1**	**1.65e−18**	**0**	**1.00e+00**	**Case 3**
16	**1**	**2.27e−03**	**1**	**1.14e−03**	**0**	**9.99e−01**	**Case 3**
17	0	5.60e−01	0	2.80e−01	0	7.22e−01	Not sign
18	**1**	**5.36e−07**	**1**	**2.68e−07**	**0**	**1.00e+00**	**Case 3**
20	0	2.06e−01	0	8.98e−01	0	1.03e−01	Not sign
21	0	2.11e−01	0	1.05e−01	0	8.96e−01	Not sign
23	1	1.34e−20	0	1.00e+00	1	6.68e−21	Case 2
24	**1**	**1.54e−07**	**1**	**7.71e−08**	**0**	**1.00e+00**	**Case 3**
25	**1**	**2.39e−17**	**1**	**1.19e−17**	**0**	**1.00e+00**	**Case 3**
28	**1**	**2.22e−04**	**1**	**1.11e−04**	**0**	**1.00e+00**	**Case 3**
$n_{\text{D}} = 30$							
1	1	1.12e−05	0	1.00e+00	1	5.60e−06	Case 2
2	1	9.47e−16	0	1.00e+00	1	4.74e−16	Case 2
3	1	6.52e−03	0	9.97e−01	1	3.26e−03	Case 2
4	1	1.10e−10	0	1.00e+00	1	5.51e−11	Case 2
5	**1**	**8.62e−16**	**1**	**4.31e−16**	**0**	**1.00e+00**	**Case 3**
7	1	3.25e−18	0	1.00e+00	1	1.63e−18	Case 2
8	0	6.99e−01	0	3.49e−01	0	6.53e−01	Not sign
9	1	2.38e−16	0	1.00e+00	1	1.19e−16	Case 2
10	0	2.55e−01	0	8.74e−01	0	1.28e−01	Not sign
11	0	6.67e−02	1	3.33e−02	0	9.67e−01	Not sign
13	1	3.50e−18	0	1.00e+00	1	1.75e−18	Case 2
14	**1**	**1.64e−13**	**1**	**8.19e−14**	**0**	**1.00e+00**	**Case 3**
15	**1**	**3.64e−13**	**1**	**1.82e−13**	**0**	**1.00e+00**	**Case 3**
16	**1**	**1.05e−08**	**1**	**5.26e−09**	**0**	**1.00e+00**	**Case 3**
17	**1**	**1.56e−16**	**1**	**7.78e−17**	**0**	**1.00e+00**	**Case 3**
18	0	1.48e−01	0	9.27e−01	0	7.41e−02	Not sign
20	1	2.84e−03	0	9.99e−01	1	1.42e−03	Case 2
21	**1**	**5.93e−05**	**1**	**2.96e−05**	**0**	**1.00e+00**	**Case 3**
23	1	1.39e−20	0	1.00e+00	1	6.95e−21	Case 2
24	**1**	**5.44e−12**	**1**	**2.72e−12**	**0**	**1.00e+00**	**Case 3**
25	**1**	**9.60e−03**	**1**	**4.80e−03**	**0**	**9.95e−01**	**Case 3**
28	**1**	**3.87e−09**	**1**	**1.94e−09**	**0**	**1.00e+00**	**Case 3**

Table 25 Wilcoxon test results, CEC 2014: MP-AIDEA versus UMOEAs, 50D and 100D

	Both		Left		Right		Result type
	h	p	h	p	h	p	
$n_D = 50$							
1	0	1.59e−01	0	9.24e−01	0	7.97e−02	Not sign
2	1	1.95e−19	0	1.00e+00	1	9.77e−20	Case 2
3	1	8.85e−08	0	1.00e+00	1	4.43e−08	Case 2
4	**1**	**1.27e−08**	**1**	**6.37e−09**	**0**	**1.00e+00**	**Case 3**
5	**1**	**1.11e−12**	**1**	**5.57e−13**	**0**	**1.00e+00**	**Case 3**
7	1	1.33e−20	0	1.00e+00	1	6.66e−21	Case 2
8	0	7.14e−02	0	9.65e−01	1	3.57e−02	Not sign
9	1	3.45e−18	0	1.00e+00	1	1.73e−18	Case 2
10	**1**	**5.37e−03**	**1**	**2.68e−03**	**0**	**9.97e−01**	**Case 3**
11	**1**	**1.43e−02**	**1**	**7.15e−03**	**0**	**9.93e−01**	**Case 3**
13	1	3.30e−18	0	1.00e+00	1	1.65e−18	Case 2
14	**1**	**3.72e−15**	**1**	**1.86e−15**	**0**	**1.00e+00**	**Case 3**
15	**1**	**2.06e−07**	**1**	**1.03e−07**	**0**	**1.00e+00**	**Case 3**
16	0	2.28e−01	0	1.14e−01	0	8.87e−01	Not sign
17	**1**	**7.51e−18**	**1**	**3.76e−18**	**0**	**1.00e+00**	**Case 3**
18	**1**	**4.45e−05**	**1**	**2.23e−05**	**0**	**1.00e+00**	**Case 3**
20	1	2.77e−02	0	9.86e−01	1	1.38e−02	Case 2
21	**1**	**8.87e−17**	**1**	**4.43e−17**	**0**	**1.00e+00**	**Case 3**
23	1	1.39e−20	0	1.00e+00	1	6.95e−21	Case 2
24	**1**	**1.79e−17**	**1**	**8.95e−18**	**0**	**1.00e+00**	**Case 3**
25	0	5.03e−01	0	2.52e−01	0	7.50e−01	Not sign
28	**1**	**3.16e−09**	**1**	**1.58e−09**	**0**	**1.00e+00**	**Case 3**
$n_D = 100$							
1	0	8.22e−02	0	9.61e−01	1	4.11e−02	Not sign
2	1	5.25e−20	0	1.00e+00	1	2.63e−20	Case 2
3	0	1.24e−01	0	6.19e−02	0	9.39e−01	Not sign
4	**1**	**2.91e−18**	**1**	**1.46e−18**	**0**	**1.00e+00**	**Case 3**
5	0	4.99e−01	0	2.50e−01	0	7.53e−01	Not sign
7	1	1.36e−20	0	1.00e+00	1	6.81e−21	Case 2
8	1	3.57e−08	0	1.00e+00	1	1.78e−08	Case 2
9	1	3.30e−18	0	1.00e+00	1	1.65e−18	Case 2
10	**1**	**5.36e−07**	**1**	**2.68e−07**	**0**	**1.00e+00**	**Case 3**
11	0	9.79e−01	0	5.13e−01	0	4.89e−01	Not sign
13	1	3.30e−18	0	1.00e+00	1	1.65e−18	Case 2
14	1	3.94e−18	0	1.00e+00	1	1.97e−18	Case 2
15	1	1.57e−15	0	1.00e+00	1	7.87e−16	Case 2
16	**1**	**3.34e−04**	**1**	**1.67e−04**	**0**	**1.00e+00**	**Case 3**
17	**1**	**4.70e−18**	**1**	**2.35e−18**	**0**	**1.00e+00**	**Case 3**
18	**1**	**7.96e−18**	**1**	**3.98e−18**	**0**	**1.00e+00**	**Case 3**
20	1	1.35e−03	0	9.99e−01	1	6.73e−04	Case 2
21	**1**	**5.29e−18**	**1**	**2.64e−18**	**0**	**1.00e+00**	**Case 3**
23	1	1.39e−20	0	1.00e+00	1	6.95e−21	Case 2
24	**1**	**1.90e−17**	**1**	**9.50e−18**	**0**	**1.00e+00**	**Case 3**
25	**1**	**2.65e−04**	**1**	**1.32e−04**	**0**	**1.00e+00**	**Case 3**
28	**1**	**5.01e−06**	**1**	**2.50e−06**	**0**	**1.00e+00**	**Case 3**

Table 26 Wilcoxon test results, CEC 2014: MP-AIDEA versus L-SHADE, 10D and 30D

	Both		Left		Right		Result type
	h	p	h	p	h	p	
$n_D = 10$							
1	**0**	**1.00e+00**	**–**	**–**	**–**	**–**	**Case 1**
2	**0**	**1.00e+00**	**–**	**–**	**–**	**–**	**Case 1**
3	**0**	**1.00e+00**	**–**	**–**	**–**	**–**	**Case 1**
4	**1**	**5.69e−12**	**1**	**2.85e−12**	**0**	**1.00e+00**	**Case 3**
5	**1**	**5.12e−15**	**1**	**2.56e−15**	**0**	**1.00e+00**	**Case 3**
7	1	6.14e−06	0	1.00e+00	1	3.07e−06	Case 2
8	1	3.47e−03	0	9.98e−01	1	1.73e−03	Case 2
9	**1**	**8.86e−09**	**1**	**4.43e−09**	**0**	**1.00e+00**	**Case 3**
10	1	4.95e−17	0	1.00e+00	1	2.47e−17	Case 2
11	1	1.85e−02	0	9.91e−01	1	9.24e−03	Case 2
13	1	1.85e−07	0	1.00e+00	1	9.26e−08	Case 2
14	**1**	**6.31e−17**	**1**	**3.16e−17**	**0**	**1.00e+00**	**Case 3**
15	**1**	**3.43e−05**	**1**	**1.72e−05**	**0**	**1.00e+00**	**Case 3**
16	1	2.91e−02	0	9.86e−01	1	1.46e−02	Case 2
17	1	5.73e−05	0	1.00e+00	1	2.87e−05	Case 2
18	0	8.10e−01	0	5.98e−01	0	4.05e−01	Not sign
20	1	4.20e−05	0	1.00e+00	1	2.10e−05	Case 2
21	0	1.01e−01	0	9.50e−01	0	5.05e−02	Not sign
23	**1**	**1.34e−20**	**1**	**6.68e−21**	**0**	**1.00e+00**	**Case 3**
24	**1**	**1.40e−05**	**1**	**7.00e−06**	**0**	**1.00e+00**	**Case 3**
25	**1**	**7.22e−13**	**1**	**3.61e−13**	**0**	**1.00e+00**	**Case 3**
28	**1**	**8.47e−11**	**1**	**4.23e−11**	**0**	**1.00e+00**	**Case 3**
$n_D = 30$							
1	1	1.79e−07	0	1.00e+00	1	8.94e−08	Case 2
2	1	9.47e−16	0	1.00e+00	1	4.74e−16	Case 2
3	1	6.52e−03	0	9.97e−01	1	3.26e−03	Case 2
4	1	4.92e−20	0	1.00e+00	1	2.46e−20	Case 2
5	**1**	**3.30e−18**	**1**	**1.65e−18**	**0**	**1.00e+00**	**Case 3**
7	1	3.25e−18	0	1.00e+00	1	1.63e−18	Case 2
8	1	4.21e−20	0	1.00e+00	1	2.10e−20	Case 2
9	1	3.30e−18	0	1.00e+00	1	1.65e−18	Case 2
10	1	1.62e−18	0	1.00e+00	1	8.12e−19	Case 2
11	1	3.40e−07	0	1.00e+00	1	1.70e−07	Case 2
13	1	1.27e−17	0	1.00e+00	1	6.34e−18	Case 2
14	**1**	**5.03e−17**	**1**	**2.52e−17**	**0**	**1.00e+00**	**Case 3**
15	**1**	**4.26e−02**	**1**	**2.13e−02**	**0**	**9.79e−01**	**Case 3**
16	1	2.30e−16	0	1.00e+00	1	1.15e−16	Case 2
17	1	2.48e−14	0	1.00e+00	1	1.24e−14	Case 2
18	1	4.18e−18	0	1.00e+00	1	2.09e−18	Case 2
20	1	4.43e−18	0	1.00e+00	1	2.22e−18	Case 2
21	1	9.72e−13	0	1.00e+00	1	4.86e−13	Case 2
23	1	1.39e−20	0	1.00e+00	1	6.95e−21	Case 2
24	**1**	**1.64e−10**	**1**	**8.21e−11**	**0**	**1.00e+00**	**Case 3**
25	1	7.97e−13	0	1.00e+00	1	3.99e−13	Case 2
28	**1**	**2.19e−11**	**1**	**1.09e−11**	**0**	**1.00e+00**	**Case 3**

Table 27 Wilcoxon test results, CEC 2014: MP-AIDEA versus L-SHADE, 50D and 100D

	Both		Left		Right		Result type
	h	p	h	p	h	p	
$n_D = 50$							
1	**1**	**2.79e−20**	**1**	**1.39e−20**	**0**	**1.00e+00**	**Case 3**
2	1	1.95e−19	0	1.00e+00	1	9.77e−20	Case 2
3	1	4.53e−20	0	1.00e+00	1	2.26e−20	Case 2
4	**1**	**1.99e−18**	**1**	**9.93e−19**	**0**	**1.00e+00**	**Case 3**
5	**1**	**3.30e−18**	**1**	**1.65e−18**	**0**	**1.00e+00**	**Case 3**
7	1	1.33e−20	0	1.00e+00	1	6.66e−21	Case 2
8	1	4.62e−19	0	1.00e+00	1	2.31e−19	Case 2
9	1	3.30e−18	0	1.00e+00	1	1.65e−18	Case 2
10	1	3.30e−18	0	1.00e+00	1	1.65e−18	Case 2
11	1	2.57e−08	0	1.00e+00	1	1.29e−08	Case 2
13	1	3.30e−18	0	1.00e+00	1	1.65e−18	Case 2
14	**1**	**4.01e−17**	**1**	**2.00e−17**	**0**	**1.00e+00**	**Case 3**
15	0	5.31e−02	1	2.65e−02	0	9.74e−01	Not sign
16	1	3.30e−18	0	1.00e+00	1	1.65e−18	Case 2
17	**1**	**2.37e−09**	**1**	**1.19e−09**	**0**	**1.00e+00**	**Case 3**
18	**1**	**8.86e−11**	**1**	**4.43e−11**	**0**	**1.00e+00**	**Case 3**
20	1	3.30e−18	0	1.00e+00	1	1.65e−18	Case 2
21	1	6.57e−12	0	1.00e+00	1	3.28e−12	Case 2
23	**1**	**2.76e−05**	**1**	**1.38e−05**	**0**	**1.00e+00**	**Case 3**
24	**1**	**3.24e−18**	**1**	**1.62e−18**	**0**	**1.00e+00**	**Case 3**
25	1	3.75e−14	0	1.00e+00	1	1.87e−14	Case 2
28	1	1.62e−03	0	9.99e−01	1	8.10e−04	Case 2
$n_D = 100$							
1	0	8.22e−02	0	9.61e−01	1	4.11e−02	Not sign
2	1	5.25e−20	0	1.00e+00	1	2.63e−20	Case 2
3	0	1.24e−01	0	6.19e−02	0	9.39e−01	Not sign
4	**1**	**2.91e−18**	**1**	**1.46e−18**	**0**	**1.00e+00**	**Case 3**
5	0	4.99e−01	0	2.50e−01	0	7.53e−01	Case 2
7	1	1.36e−20	0	1.00e+00	1	6.81e−21	Case 2
8	1	3.57e−08	0	1.00e+00	1	1.78e−08	Case 2
9	1	3.30e−18	0	1.00e+00	1	1.65e−18	Case 2
10	**1**	**5.36e−07**	**1**	**2.68e−07**	**0**	**1.00e+00**	**Case 3**
11	0	9.79e−01	0	5.13e−01	0	4.89e−01	Not sign
13	1	3.30e−18	0	1.00e+00	1	1.65e−18	Case 2
14	1	3.94e−18	0	1.00e+00	1	1.97e−18	Case 2
15	1	1.57e−15	0	1.00e+00	1	7.87e−16	Case 2
16	**1**	**3.34e−04**	**1**	**1.67e−04**	**0**	**1.00e+00**	**Case 3**
17	**1**	**4.70e−18**	**1**	**2.35e−18**	**0**	**1.00e+00**	**Case 3**
18	**1**	**7.96e−18**	**1**	**3.98e−18**	**0**	**1.00e+00**	**Case 3**
20	1	1.35e−03	0	9.99e−01	1	6.73e−04	Case 2
21	**1**	**5.29e−18**	**1**	**2.64e−18**	**0**	**1.00e+00**	**Case 3**
23	1	1.39e−20	0	1.00e+00	1	6.95e−21	Case 2
24	**1**	**1.90e−17**	**1**	**9.50e−18**	**0**	**1.00e+00**	**Case 3**
25	**1**	**2.65e−04**	**1**	**1.32e−04**	**0**	**1.00e+00**	**Case 3**
28	**1**	**5.01e−06**	**1**	**2.50e−06**	**0**	**1.00e+00**	**Case 3**

Table 28 Wilcoxon test results, CEC 2014: MP-AIDEA versus MVMO, 10D and 30D

	Both		Left		Right		Result type
	h	p	h	p	h	p	
$n_D = 10$							
1	**1**	**1.39e−20**	**1**	**6.95e−21**	**0**	**1.00e+00**	**Case 3**
2	**1**	**1.39e−20**	**1**	**6.95e−21**	**0**	**1.00e+00**	**Case 3**
3	**1**	**1.35e−20**	**1**	**6.73e−21**	**0**	**1.00e+00**	**Case 3**
4	**1**	**6.66e−08**	**1**	**3.33e−08**	**0**	**1.00e+00**	**Case 3**
5	**1**	**3.64e−13**	**1**	**1.82e−13**	**0**	**1.00e+00**	**Case 3**
7	**1**	**2.31e−07**	**1**	**1.16e−07**	**0**	**1.00e+00**	**Case 3**
8	1	1.35e−02	0	9.93e−01	1	6.75e−03	Case 2
9	**1**	**5.49e−07**	**1**	**2.75e−07**	**0**	**1.00e+00**	**Case 3**
10	**1**	**2.32e−02**	**1**	**1.16e−02**	**0**	**9.89e−01**	**Case 3**
11	0	9.15e−01	0	4.57e−01	0	5.45e−01	Not sign
13	1	1.75e−15	0	1.00e+00	1	8.77e−16	Case 2
14	**1**	**8.38e−17**	**1**	**4.19e−17**	**0**	**1.00e+00**	**Case 3**
15	**1**	**1.60e−04**	**1**	**8.01e−05**	**0**	**1.00e+00**	**Case 3**
16	0	4.14e−01	0	2.07e−01	0	7.95e−01	Not sign
17	0	1.37e−01	0	9.32e−01	0	6.87e−02	Not sign
18	**1**	**1.54e−05**	**1**	**7.68e−06**	**0**	**1.00e+00**	**Case 3**
20	0	7.72e−02	0	9.62e−01	1	3.86e−02	Not sign
21	0	7.08e−02	0	9.65e−01	1	3.54e−02	Not sign
23	1	1.34e−20	0	1.00e+00	1	6.68e−21	Case 2
24	**1**	**1.88e−06**	**1**	**9.41e−07**	**0**	**1.00e+00**	**Case 3**
25	**1**	**1.36e−12**	**1**	**6.81e−13**	**0**	**1.00e+00**	**Case 3**
28	0	1.95e−01	0	9.03e−01	0	9.77e−02	Not sign
$n_D = 30$							
1	**1**	**1.28e−18**	**1**	**6.41e−19**	**0**	**1.00e+00**	**Case 3**
2	**1**	**3.08e−18**	**1**	**1.54e−18**	**0**	**1.00e+00**	**Case 3**
3	**1**	**1.15e−19**	**1**	**5.75e−20**	**0**	**1.00e+00**	**Case 3**
4	0	1.22e−01	0	6.09e−02	0	9.40e−01	Not sign
5	0	7.72e−02	0	9.62e−01	1	3.86e−02	Not sign
7	**1**	**1.43e−18**	**1**	**7.17e−19**	**0**	**1.00e+00**	**Case 3**
8	1	2.73e−16	0	1.00e+00	1	1.36e−16	Case 2
9	**1**	**2.42e−15**	**1**	**1.21e−15**	**0**	**1.00e+00**	**Case 3**
10	1	4.33e−02	0	9.79e−01	1	2.16e−02	Case 2
11	0	2.11e−01	0	1.05e−01	0	8.96e−01	Not sign
13	0	3.81e−01	0	8.11e−01	0	1.90e−01	Not sign
14	**1**	**4.33e−05**	**1**	**2.16e−05**	**0**	**1.00e+00**	**Case 3**
15	**1**	**3.91e−08**	**1**	**1.96e−08**	**0**	**1.00e+00**	**Case 3**
16	0	1.25e−01	0	9.38e−01	0	6.27e−02	Not sign
17	1	6.57e−12	0	1.00e+00	1	3.28e−12	Case 2
18	1	1.62e−08	0	1.00e+00	1	8.09e−09	Case 2
20	1	8.56e−06	0	1.00e+00	1	4.28e−06	Case 2
21	1	4.53e−04	0	1.00e+00	1	2.27e−04	Case 2
23	1	1.39e−20	0	1.00e+00	1	6.95e−21	Case 2
24	**1**	**3.34e−15**	**1**	**1.67e−15**	**0**	**1.00e+00**	**Case 3**
25	**1**	**3.29e−10**	**1**	**1.65e−10**	**0**	**1.00e+00**	**Case 3**
28	**1**	**3.64e−13**	**1**	**1.82e−13**	**0**	**1.00e+00**	**Case 3**

Table 29 Wilcoxon test results, CEC 2014: MP-AIDEA versus MVMO, 50D and 100D

	Both		Left		Right		Result type
	h	p	h	p	h	p	
$n_D = 50$							
1	**1**	**2.79e−20**	**1**	**1.39e−20**	**0**	**1.00e+00**	**Case 3**
2	**1**	**3.29e−18**	**1**	**1.65e−18**	**0**	**1.00e+00**	**Case 3**
3	**1**	**2.92e−18**	**1**	**1.46e−18**	**0**	**1.00e+00**	**Case 3**
4	**1**	**9.96e−03**	**1**	**4.98e−03**	**0**	**9.95e−01**	**Case 3**
5	1	1.24e−05	0	1.00e+00	1	6.20e−06	Case 2
7	**1**	**3.01e−18**	**1**	**1.51e−18**	**0**	**1.00e+00**	**Case 3**
8	1	1.15e−15	0	1.00e+00	1	5.75e−16	Case 2
9	**1**	**6.06e−06**	**1**	**3.03e−06**	**0**	**1.00e+00**	**Case 3**
10	0	6.06e−01	0	3.03e−01	0	6.99e−01	Not sign
11	1	2.65e−04	0	1.00e+00	1	1.32e−04	Case 2
13	1	1.79e−07	0	1.00e+00	1	8.93e−08	Case 2
14	0	6.98e−01	0	3.49e−01	0	6.54e−01	Not sign
15	0	9.79e−01	0	4.89e−01	0	5.13e−01	Not sign
16	1	2.63e−02	0	9.87e−01	1	1.31e−02	Case 2
17	1	2.20e−08	0	1.00e+00	1	1.10e−08	Case 2
18	1	6.68e−15	0	1.00e+00	1	3.34e−15	Case 2
20	1	3.50e−18	0	1.00e+00	1	1.75e−18	Case 2
21	1	9.14e−10	0	1.00e+00	1	4.57e−10	Case 2
23	1	1.39e−20	0	1.00e+00	1	6.95e−21	Case 2
24	**1**	**8.28e−08**	**1**	**4.14e−08**	**0**	**1.00e+00**	**Case 3**
25	1	1.93e−09	0	1.00e+00	1	9.66e−10	Case 2
28	**1**	**6.34e−15**	**1**	**3.17e−15**	**0**	**1.00e+00**	**Case 3**
$n_D = 100$							
1	**1**	**3.83e−20**	**1**	**1.92e−20**	**0**	**1.00e+00**	**Case 3**
2	**1**	**3.28e−18**	**1**	**1.64e−18**	**0**	**1.00e+00**	**Case 3**
3	**1**	**3.30e−18**	**1**	**1.65e−18**	**0**	**1.00e+00**	**Case 3**
4	**1**	**2.03e−16**	**1**	**1.01e−16**	**0**	**1.00e+00**	**Case 3**
5	1	2.27e−12	0	1.00e+00	1	1.14e−12	Case 2
7	**1**	**6.75e−18**	**1**	**3.37e−18**	**0**	**1.00e+00**	**Case 3**
8	1	2.26e−17	0	1.00e+00	1	1.13e−17	Case 2
9	**1**	**5.01e−04**	**1**	**2.51e−04**	**0**	**1.00e+00**	**Case 3**
10	**1**	**7.25e−13**	**1**	**3.63e−13**	**0**	**1.00e+00**	**Case 3**
11	0	7.79e−01	0	3.89e−01	0	6.13e−01	Not sign
13	1	1.65e−16	0	1.00e+00	1	8.23e−17	Case 2
14	1	3.30e−18	0	1.00e+00	1	1.65e−18	Case 2
15	0	6.06e−01	0	6.99e−01	0	3.03e−01	Not sign
16	0	9.20e−01	0	5.43e−01	0	4.60e−01	Not sign
17	**1**	**2.04e−04**	**1**	**1.02e−04**	**0**	**1.00e+00**	**Case 3**
18	1	3.30e−18	0	1.00e+00	1	1.65e−18	Case 2
20	1	3.30e−18	0	1.00e+00	1	1.65e−18	Case 2
21	1	8.37e−03	0	9.96e−01	1	4.18e−03	Case 2
23	1	1.39e−20	0	1.00e+00	1	6.95e−21	Case 2
24	1	5.32e−17	0	1.00e+00	1	2.66e−17	Case 2
25	**1**	**6.58e−16**	**1**	**3.29e−16**	**0**	**1.00e+00**	**Case 3**
28	**1**	**1.41e−15**	**1**	**7.06e−16**	**0**	**1.00e+00**	**Case 3**

Table 30 Wilcoxon test results, CEC 2014: MP-AIDEA versus CMLSP, 10D and 30D

	Both		Left		Right		Result type
	h	p	h	p	h	p	
$n_D = 10$							
1	**1**	**1.39e−20**	**1**	**6.95e−21**	**0**	**1.00e+00**	**Case 3**
2	**1**	**1.39e−20**	**1**	**6.95e−21**	**0**	**1.00e+00**	**Case 3**
3	**1**	**1.39e−20**	**1**	**6.95e−21**	**0**	**1.00e+00**	**Case 3**
4	**1**	**4.38e−18**	**1**	**2.19e−18**	**0**	**1.00e+00**	**Case 3**
5	**1**	**3.30e−18**	**1**	**1.65e−18**	**0**	**1.00e+00**	**Case 3**
7	**1**	**1.71e−15**	**1**	**8.56e−16**	**0**	**1.00e+00**	**Case 3**
8	**1**	**1.47e−19**	**1**	**7.36e−20**	**0**	**1.00e+00**	**Case 3**
9	**1**	**2.05e−18**	**1**	**1.02e−18**	**0**	**1.00e+00**	**Case 3**
10	**1**	**3.28e−18**	**1**	**1.64e−18**	**0**	**1.00e+00**	**Case 3**
11	**1**	**3.30e−18**	**1**	**1.65e−18**	**0**	**1.00e+00**	**Case 3**
13	**1**	**6.01e−15**	**1**	**3.01e−15**	**0**	**1.00e+00**	**Case 3**
14	**1**	**3.30e−18**	**1**	**1.65e−18**	**0**	**1.00e+00**	**Case 3**
15	**1**	**3.30e−18**	**1**	**1.65e−18**	**0**	**1.00e+00**	**Case 3**
16	**1**	**1.01e−17**	**1**	**5.03e−18**	**0**	**1.00e+00**	**Case 3**
17	**1**	**5.09e−08**	**1**	**2.55e−08**	**0**	**1.00e+00**	**Case 3**
18	**1**	**2.57e−13**	**1**	**1.29e−13**	**0**	**1.00e+00**	**Case 3**
20	**1**	**9.67e−15**	**1**	**4.84e−15**	**0**	**1.00e+00**	**Case 3**
21	**1**	**1.14e−06**	**1**	**5.71e−07**	**0**	**1.00e+00**	**Case 3**
23	1	1.34e−20	0	1.00e+00	1	6.68e−21	Case 2
24	**1**	**3.29e−18**	**1**	**1.64e−18**	**0**	**1.00e+00**	**Case 3**
25	**1**	**8.30e−07**	**1**	**4.15e−07**	**0**	**1.00e+00**	**Case 3**
28	**1**	**1.13e−04**	**1**	**5.64e−05**	**0**	**1.00e+00**	**Case 3**
30D							
1	**1**	**1.28e−18**	**1**	**6.41e−19**	**0**	**1.00e+00**	**Case 3**
2	**1**	**3.08e−18**	**1**	**1.54e−18**	**0**	**1.00e+00**	**Case 3**
3	**1**	**1.15e−19**	**1**	**5.75e−20**	**0**	**1.00e+00**	**Case 3**
4	**1**	**3.12e−18**	**1**	**1.56e−18**	**0**	**1.00e+00**	**Case 3**
5	**1**	**3.30e−18**	**1**	**1.65e−18**	**0**	**1.00e+00**	**Case 3**
7	**1**	**1.42e−18**	**1**	**7.11e−19**	**0**	**1.00e+00**	**Case 3**
8	**1**	**2.76e−18**	**1**	**1.38e−18**	**0**	**1.00e+00**	**Case 3**
9	1	3.86e−02	0	9.81e−01	1	1.93e−02	Case 2
10	**1**	**3.30e−18**	**1**	**1.65e−18**	**0**	**1.00e+00**	**Case 3**
11	**1**	**3.30e−18**	**1**	**1.65e−18**	**0**	**1.00e+00**	**Case 3**
13	**1**	**6.32e−03**	**1**	**3.16e−03**	**0**	**9.97e−01**	**Case 3**
14	**1**	**3.30e−18**	**1**	**1.65e−18**	**0**	**1.00e+00**	**Case 3**
15	**1**	**3.30e−18**	**1**	**1.65e−18**	**0**	**1.00e+00**	**Case 3**
16	**1**	**3.30e−18**	**1**	**1.65e−18**	**0**	**1.00e+00**	**Case 3**
17	1	3.19e−17	0	1.00e+00	1	1.59e−17	Case 2
18	1	4.55e−09	0	1.00e+00	1	2.28e−09	Case 2
20	1	3.21e−14	0	1.00e+00	1	1.61e−14	Case 2
21	1	4.99e−02	0	9.75e−01	1	2.49e−02	Case 2
23	1	1.85e−17	0	1.00e+00	1	9.24e−18	Case 2
24	1	1.40e−02	0	9.93e−01	1	7.02e−03	Case 2
25	1	3.58e−10	0	1.00e+00	1	1.79e−10	Case 2
28	1	5.36e−07	0	1.00e+00	1	2.68e−07	Case 2

Table 31 Wilcoxon test results, CEC 2014: MP-AIDEA versus CMLSP, 50D and 100D

	Both		Left		Right		Result type
	h	p	h	p	h	p	
$n_D = 50$							
1	1	2.79e−20	1	1.39e−20	0	1.00e+00	Case 3
2	1	3.29e−18	1	1.65e−18	0	1.00e+00	Case 3
3	1	2.92e−18	1	1.46e−18	0	1.00e+00	Case 3
4	1	3.29e−18	1	1.65e−18	0	1.00e+00	Case 3
5	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
7	1	3.20e−18	1	1.60e−18	0	1.00e+00	Case 3
8	1	3.16e−18	1	1.58e−18	0	1.00e+00	Case 3
9	1	1.69e−17	1	8.46e−18	0	1.00e+00	Case 3
10	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
11	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
13	1	1.07e−04	1	5.33e−05	0	1.00e+00	Case 3
14	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
15	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
16	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
17	1	5.00e−16	1	2.50e−16	0	1.00e+00	Case 3
18	1	4.14e−15	1	2.07e−15	0	1.00e+00	Case 3
20	1	1.27e−17	0	1.00e+00	1	6.34e−18	Case 2
21	1	1.58e−12	1	7.89e−13	0	1.00e+00	Case 3
23	1	7.50e−19	1	3.75e−19	0	1.00e+00	Case 3
24	1	2.13e−17	1	1.07e−17	0	1.00e+00	Case 3
25	1	5.67e−04	0	1.00e+00	1	2.84e−04	Case 2
28	1	9.60e−03	1	4.80e−03	0	9.95e−01	Case 3
$n_D = 100$							
1	1	3.83e−20	1	1.92e−20	0	1.00e+00	Case 3
2	1	3.28e−18	1	1.64e−18	0	1.00e+00	Case 3
3	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
4	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
5	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
7	1	3.25e−18	1	1.63e−18	0	1.00e+00	Case 3
8	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
9	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
10	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
11	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
13	1	2.70e−13	1	1.35e−13	0	1.00e+00	Case 3
14	1	7.72e−10	0	1.00e+00	1	3.86e−10	Case 2
15	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
16	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
17	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
18	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
20	1	4.70e−18	0	1.00e+00	1	2.35e−18	Case 2
21	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
23	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
24	1	3.30e−18	1	1.65e−18	0	1.00e+00	Case 3
25	1	1.36e−04	1	6.81e−05	0	1.00e+00	Case 3
28	1	9.90e−14	1	4.95e−14	0	1.00e+00	Case 3

References

Alonso S, Jimenez J, Carmona H, Galvan B, Winter G (2005) Performance of a flexible evolutionary algorithm. In: IEEE congress on evolutionary computation (CEC)

Asafuddoula M, Ray T, Sarker R (2011) An adaptive differential evolution algorithm and its performance on real world optimization problems. In: IEEE congress on evolutionary computation (CEC), pp 1057–1062. https://doi.org/10.1109/CEC.2011.5949734

Auger A, Hansen, N (2005a) Performance evaluation of an advanced local search evolutionary algorithm. In: IEEE congress on evolutionary computation (CEC), pp 1777–1784. https://doi.org/10.1109/CEC.2005.1554903

Auger A, Hansen N (2005b) A restart CMA evolution strategy with increasing population size. In: IEEE congress on evolutionary computation (CEC), pp 1769–1776. https://doi.org/10.1109/CEC.2005.1554902

Ballester PJ, Stephenson J, Carter JN, Gallagher K (2005) Real-parameter optimization performance study on the CEC-2005 benchmark with SPC-PNX. In: IEEE congress on evolutionary computation (CEC), pp 498–505. https://doi.org/10.1109/CEC.2005.1554724

Bandaru S, Tulshyan R, Deb K (2011) Modified SBX and adaptive mutation for real world single objective optimization. In: IEEE congress on evolutionary computation (CEC), pp 1335–1342. https://doi.org/10.1109/CEC.2011.5949771

Becker W, Yu X, Tu J (2005) EvLib: a parameterless self-adaptive real-valued optimisation library. In: IEEE congress on evolutionary computation (CEC), pp 498–505

Bezdek JC (1981) Pattern recognition with fuzzy objective function algorithms. Springer, New York. https://doi.org/10.1007/978-1-4757-0450-1

Brest J, Greiner S, Bošković B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans Evol Comput 10(6):646–657. https://doi.org/10.1109/TEVC.2006.872133

Brest J, Korošec P, Silc J, Zamuda A, Boškovič B, Maučec MS (2013) Differential evolution and differential ant-stigmergy on dynamic optimization problems. Int J Syst Sci 44(4):663–679. https://doi.org/10.1080/00207721.2011.617899

Bui LT, Shan Y, Qi F, Abbass HA (2005) Comparing two versions of differential evolution in real parameter optimization. In: IEEE congress on evolutionary computation (CEC)

Bujok P, Tvrdik J, Polakova R (2014) Differential evolution with rotation-invariant mutation and competing-strategies adaptation. In: IEEE congress on evolutionary computation (CEC), pp 2253–2258. https://doi.org/10.1109/CEC.2014.6900626

Chen L, Liu Hl, Zheng Z, Xie S (2014) An evolutionary algorithm based on covariance matrix leaning and searching preference for solving CEC 2014 benchmark problems. In: IEEE congress on evolutionary computation (CEC), pp 2672–2677. https://doi.org/10.1109/CEC.2014.6900594

Competition on testing evolutionary algorithms on real-world numerical optimization problems @ CEC11, New Orleans, USA, June 2011. http://www3.ntu.edu.sg/home/epnsugan/index_files/CEC11-RWP/CEC11-RWP.htm. Accessed Apr 2015

Costa L (2005) A parameter-less evolution strategy for global optimization. In: IEEE congress on evolutionary computation (CEC)

Das S, Suganthan PN (2010) Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. Technical report, Jadavpur University, Nanyang Technological University

Das S, Suganthan PN (2011) Differential evolution: a survey of the state-of-the-art. IEEE Trans Evol Comput 15(1):4–31. https://doi.org/10.1109/TEVC.2010.2059031

Di Carlo M, Vasile M, Minisci E (2015) Multi-population inflationary differential evolution algorithm with adaptive local restart. In: IEEE congress on evolutionary computation (CEC), pp 632–639. https://doi.org/10.1109/CEC.2015.7256950

Dourado Maia R, Nunes de Castro L, Matos Caminhas W (2014) Real-parameter optimization with optbees. In: IEEE congress on evolutionary computation (CEC), pp 2649–2655. https://doi.org/10.1109/CEC.2014.6900549

Elsayed SM, Sarker RA, Essam DL (2011a) Differential evolution with multiple strategies for solving CEC2011 real-world numerical optimization problems. In: IEEE congress on evolutionary computation (CEC), pp 1041–1048. https://doi.org/10.1109/CEC.2011.5949732

Elsayed SM, Sarker RA, Essam DL (2011b) GA with a new multi-parent crossover for solving IEEE-CEC2011 competition problems. In: IEEE congress on evolutionary computation (CEC), pp 1034–1040. https://doi.org/10.1109/CEC.2011.5949731

Erlich I, Rueda JL, Wildenhues S, Shewarega F (2014) Evaluating the mean-variance mapping optimization on the IEEE-CEC 2014 test suite. In: IEEE congress on evolutionary computation (CEC), pp 1625–1632. https://doi.org/10.1109/CEC.2014.6900516

Elsayed SM, Sarker RA, Essam DL, Hamza NM (2014) Testing united multi-operator evolutionary algorithms on the CEC2014 real-parameter numerical optimization. In: IEEE congress on evolutionary computation (CEC), pp 1650–1657. https://doi.org/10.1109/CEC.2014.6900308

García-Martínez C, Lozano M (2005) Hybrid real-coded genetic algorithms with female and male differentiation. In: IEEE congress on evolutionary computation (CEC), pp 896–903. https://doi.org/10.1109/CEC.2005.1554778

Haider U, Das S, Maity D, Abraham A, Dasgupta P (2011) Self adaptive cluster based and weed inspired differential evolution algorithm for real world optimization. In: IEEE congress on evolutionary computation (CEC), pp 750–756. https://doi.org/10.1109/CEC.2011.5949694

Hu Z, Bao Y, Xiong T (2014) Partial opposition-based adaptive differential evolution algorithms: evaluation on the CEC 2014 benchmark set for real-parameter optimization. In: IEEE congress on evolutionary computation (CEC), pp 2259–2265. https://doi.org/10.1109/CEC.2014.6900489

Hybrid differential evolution algorithm with adaptive crossover mechanism. http://uk.mathworks.com/matlabcentral/fileexchange/39217-hybrid-differential-evolution-algorithm-with-adaptive-crossover-mechanism/content/DE_TCRparam.m. Accessed Apr 2015

Korošec P, Šilc J (2011) The continuous differential ant-stigmergy algorithm applied to real-world optimization problems. In: IEEE congress on evolutionary computation (CEC), pp 1327–1334. https://doi.org/10.1109/CEC.2011.5949770

LaTorre A, Muelas S, Peña JM (2011) Benchmarking a hybrid DE-RHC algorithm on real world problems. In: IEEE congress on evolutionary computation (CEC), pp 1027–1033. https://doi.org/10.1109/CEC.2011.5949730

Liang JJ, Suganthan PN (2005) Dynamic multi-swarm particle swarm optimizer with local search. In: IEEE congress on evolutionary computation (CEC), pp 522–528. https://doi.org/10.1109/CEC.2005.1554727

Liang J, Qu B, Suganthan P (2013) Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Technical report

Li Z, Shang Z, Qu BY, Liang JJ (2014) Differential evolution strategy based on the constraint of fitness values classification. In:

IEEE congress on evolutionary computation (CEC), pp 1454–1460. https://doi.org/10.1109/CEC.2014.6900507

Liu J, Lampinen J (2005) A fuzzy adaptive differential evolution algorithm. Soft Comput 9(6):448–462. https://doi.org/10.1007/s00500-004-0363-x

Locatelli M, Vasile M (2015) (Non) convergence results for the differential evolution method. Optim Lett 9(3):413–425. https://doi.org/10.1007/s11590-014-0816-9

Mallipeddi R, Suganthan PN (2011) Ensemble differential evolution algorithm for CEC2011 problems. In: IEEE congress on evolutionary computation (CEC), pp 1557–1564. https://doi.org/10.1109/CEC.2011.5949801

Mallipeddi R, Wu G, Lee M, Suganthan P (2014) Gaussian adaptation based parameter adaptation for differential evolution. In: IEEE congress on evolutionary computation (CEC), pp 1760–1767. https://doi.org/10.1109/CEC.2014.6900601

Mandal A, Das AK, Mukherjee P, Das S, Suganthan PN (2011) Modified differential evolution with local search algorithm for real world optimization. In: IEEE congress on evolutionary computation (CEC), pp 1565–1572. https://doi.org/10.1109/CEC.2011.5949802

Minisci E, Vasile M (2014) Adaptive inflationary differential evolution. In: IEEE congress on evolutionary computation (CEC), pp 1792–1799. https://doi.org/10.1109/CEC.2014.6900587

Molina D, Herrera F, Lozano M (2005) Adaptive local search parameters for real-coded memetic algorithms. In: IEEE congress on evolutionary computation (CEC), pp 888–895. https://doi.org/10.1109/CEC.2005.1554777

Molina D, Lacroix B, Herrera F (2014) Influence of regions on the memetic algorithm for the cec'2014 special session on real-parameter single objective optimisation. In: IEEE congress on evolutionary computation (CEC), pp 1633–1640. https://doi.org/10.1109/CEC.2014.6900536

Omran MG, Salman A, Engelbrecht AP (2005) Self-adaptive differential evolution. In: Hao Y, Liu J, Wang Y, Cheung Y-M, Yin H, Jiao L, Ma J, Jiao Y-C (eds) Computational intelligence and security. Springer, Berlin, pp 192–199. https://doi.org/10.1007/11596448_28

Papa G (2013) Parameter-less algorithm for evolutionary-based optimization. Comput Optim Appl 56(1):209–229. https://doi.org/10.1007/s10589-013-9565-4

Polakova R, Tvrdik J, Bujok P (2014) Controlled restart in differential evolution applied to CEC2014 benchmark functions. In: IEEE congress on evolutionary computation (CEC), pp 2230–2236. https://doi.org/10.1109/CEC.2014.6900632

Ponnuthurai nagaratnam sugantha shared documents. http://web.mysites.ntu.edu.sg/epnsugan/PublicSite/Shared%20Documents/Forms/AllItems.aspx. Accessed Apr 2015

Pošik P (2005) Real-parameter optimization using the mutation step co-evolution. In: IEEE congress on evolutionary computation (CEC), pp 872–879. https://doi.org/10.1109/CEC.2005.1554775

Preux P, Munos R, Valko M (2014) Bandits attack function optimization. In: IEEE congress on evolutionary computation (CEC), pp 2245–2252. https://doi.org/10.1109/CEC.2014.6900558

Price K, Storn RM, Lampinen JA (2006) Differential evolution: a practical approach to global optimization. Springer, Berlin. https://doi.org/10.1007/3-540-31306-0

Qin AK, Suganthan PN (2005) Self-adaptive differential evolution algorithm for numerical optimization. In: IEEE congress on evolutionary computation (CEC), pp 1785–1791. https://doi.org/10.1109/CEC.2005.1554904

Qing A (2010) Fundamentals of differential evolution. Differential evolution: fundamentals and applications in electrical engineering. Wiley, New York, pp 41–60. https://doi.org/10.1002/9780470823941.ch2

Qu B, Liang J, Xiao J, Shang Z (2014) Memetic differential evolution based on fitness Euclidean-distance ratio. In: IEEE congress on evolutionary computation (CEC), pp 2266–2273. https://doi.org/10.1109/CEC.2014.6900476

Reynoso-Meza G, Sanchis J, Ferragud XB, Durá JMH (2011) Hybrid de algorithm with adaptive crossover operator for solving real-world numerical optimization problems. In: IEEE congress on evolutionary computation (CEC), pp 1551–1556. https://doi.org/10.1109/CEC.2011.5949800

Ronkkonen J, Kukkonen S, Price KV (2005) Real-parameter optimization with differential evolution. In: IEEE congress on evolutionary computation (CEC), pp 506–513. https://doi.org/10.1109/CEC.2005.1554725

Saha A, Ray T (2011) How does the good old genetic algorithm fare at real world optimization? In: IEEE congress on evolutionary computation (CEC), pp 1049–1056. https://doi.org/10.1109/CEC.2011.5949733

Singh HK, Ray T (2011) Performance of a hybrid EA-DE-memetic algorithm on CEC 2011 real world optimization problems. In: IEEE congress on evolutionary computation (CEC), pp 1322–1326. https://doi.org/10.1109/CEC.2011.5949769

Sinha A, Tiwari S, Deb K (2005) A population-based, steady-state procedure for real-parameter optimization. In: IEEE congress on evolutionary computation (CEC), pp 514–521. https://doi.org/10.1109/CEC.2005.1554726

Suganthan P (2011) Testing evolutionary algorithms on real-world numerical optimization problems. Nanyang Technological University, School of Electrical and Electronics Engineering, Singapore

Suganthan PN, Hansen N, Liang JJ, Deb K, Chen YP, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical report, Nanyang Technological University

Tanabe R, Fukunaga AS (2014) Improving the search performance of shade using linear population size reduction. In: IEEE congress on evolutionary computation (CEC), pp 1658–1665. https://doi.org/10.1109/CEC.2014.6900380

Vasile M, Minisci E, Locatelli M (2011) An inflatory differential evolution algorithm for space trajectory optimization. IEEE Trans Evol Comput 15(2):267–281. https://doi.org/10.1109/TEVC.2010.2087026

Wales DJ, Doye JP (1997) Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. J Phys Chem A 101(28):5111–5116. https://doi.org/10.1021/jp970984n

Wang Y, Li B, Zhang K (2011) Estimation of distribution and differential evolution cooperation for real-world numerical optimization problems. In: IEEE congress on evolutionary computation (CEC), pp 1315–1321. https://doi.org/10.1109/CEC.2011.5949768

Xu C, Huang H, Ye S (2014) A differential evolution with replacement strategy for real-parameter numerical optimization. In: IEEE congress on evolutionary computation (CEC), pp 1617–1624. https://doi.org/10.1109/CEC.2014.6900468

Yashesh D, Deb K, Bandaru S (2014) Non-uniform mapping in real-coded genetic algorithms. In: IEEE congress on evolutionary computation (CEC), pp 2237–2244. https://doi.org/10.1109/CEC.2014.6900621

Yu C, Kelley L, Zheng S, Tan Y (2014) Fireworks algorithm with differential mutation for solving the CEC 2014 competition problems. In: IEEE congress on evolutionary computation (CEC), pp 3238–3245. https://doi.org/10.1109/CEC.2014.6900590

Yuan B, Gallagher M (2005) Experimental results for the special session on real-parameter optimization at cec 2005: a simple, continuous EDA. In: IEEE congress on evolutionary computation (CEC), pp 1792–1799. https://doi.org/10.1109/CEC.2005.1554905

Zhang J, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. IEEE Trans Evol Comput 13(5):945–958. https://doi.org/10.1109/TEVC.2009.2014613