
Strathprints is designed to allow users to access the research output of the University of Strathclyde. Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. You may not engage in further distribution of the material for any profiting activities or any commercial gain. You may freely distribute both the url (http://strathprints.strath.ac.uk/) and the content of this paper for research or study, educational, or not-for-profit purposes without prior permission or charge.

Any correspondence concerning this service should be sent to Strathprints administrator: strathprints@strath.ac.uk
Multi-Population Inflationary Differential Evolution Algorithm with Adaptive Local Restart

Marilena Di Carlo, Massimiliano Vasile, Member, IEEE, and Edmondo Minisci, Member, IEEE

Abstract—In this paper a Multi-Population Inflationary Differential Evolution algorithm with Adaptive Local Restart is presented and extensively tested over more than fifty test functions from the CEC 2005, CEC 2011 and CEC 2014 competitions. The algorithm combines a multi-population adaptive Differential Evolution with local search and local and global restart procedures. 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 local restart of the population, which follows the local search, is, therefore, automatically adapted.

Keywords—Global optimization, differential evolution, multi-population algorithm, adaptive algorithm.

I. INTRODUCTION

Differential Evolution (DE) is a population-based global optimization technique over continuous spaces [1]. Existing literature indicates that DE exhibits very good performance over a wide variety of optimization problems [2]. However, although being a very efficient optimizer, its local search ability has long been questioned, [3], and work has been done to improve its local convergence by combining DE with local optimization strategies. Furthermore, stagnation due to the collapse of the population to a fixed point or to a level set has been theoretically demonstrated [4][5]. In [4], Inflationary Differential Evolution Algorithm (IDEA) was introduced. IDEA is based on the hybridization of DE with the restarting procedure of Monotonic Basin Hopping (MBH) [6]. IDEA gives better results than a simple DE but its performance is dependent upon the parameters controlling both the DE and MBH heuristics [4]. In particular, the DE heuristic, for a given population size, depends on two main control parameters, the crossover probability CR, and the differential weight F, whose best settings are problem dependent, [2], [7]. Different adaptive mechanisms for adjusting the control parameters during the search process can be found in the literature [8], [9], [10]. In [11] the authors proposed an Adaptive Inflationary Differential Evolution Algorithm (AIDEA) that uses a probabilistic kernel based approach to automatically adapt the values of both CR and F. In [12] a further improvement of AIDEA, the Multi-Population Adaptive Inflationary Differential Evolution Algorithm (MP-AIDEA), was introduced, which automatically adapts the neighborhood of a local minimum, within which the search is restarted.

This paper presents a new implementation of MP-AIDEA, the Multi-Population Adaptive Inflationary Differential Evolution Algorithm with Adaptive Local Restart (MP-AIDEA-ALR), in which the number of local restarts is adapted and integrated with the restart neighborhood adaptation. In order to assess the performance of MP-AIDEA-ALR, the new algorithm was extensively tested over a great number of test problems from the past competitions of the Congress on Evolutionary Computation. The paper starts with a section that introduces MP-AIDEA with Adaptive Local Restart. Then the test cases and the obtained results are presented.

II. MULTI-POPULATION INFLATIONARY DIFFERENTIAL EVOLUTION WITH ADAPTIVE LOCAL RESTART

The algorithm presented in this paper is a further extension of MP-AIDEA, [12]. MP-AIDEA starts with the initialization of multiple populations in the search space. For each population, a DE process is run in which each individual is associated to a different value of CR and F. During the evolution of the populations from parents to children the values of CR and F are automatically adapted [11]. The DE heuristics is iterated until a population contracts below a given threshold, identified by a contraction parameter $\bar{\rho}$. When the contraction condition is satisfied a local search is run from the best individual in the population. The resulting local minimum is archived in an archive of minima $A$, common to all the populations, and the population is restarted in a bubble of dimension $\delta_{local}$ around the local minimum (local restart). The parameter $\delta_{local}$ is adapted by assessing the distance between minima found at subsequent local restart. In theory, a local restart is effective if a transition from one local minimum to a local minimum with a better value of the objective function occurs. Therefore, in this paper the dimension of the bubble is deemed to be appropriate, if the populations move from a set of local minima to another set with better local minima. Local restart is iterated up to a predefined maximum number of times, identified by the value $n_{LR}$. When this value is reached the population is restarted globally, rather than locally, at a distance $\delta_{global}$ from the cluster of local minima found thus far (global restart). The algorithm stops when the maximum number of function evaluation is reached. In the new implementation of MP-AIDEA the parameter $n_{LR}$ is replaced with a mechanism that detects when the population contracts in the basin of attraction of a local minimum which has already been identified. The whole MP-AIDEA with Adaptive Local Restart is described in Algorithm 1. The first step of the optimization process is the initialization of $N_p$ populations, composed of $n_p$ individuals, in the search space (Alg. 1, line 1-3). Then, a joint PDF...
over the values of $CR$ and $F$, $CRF$, is initialized with a uniform distribution (Alg. 1, line 5-8). The values of $CR$ and $F$ for each individual of each population are drawn from $CRF$. Each population evolves, independently of the others, by implementing the DE heuristics: the mutant vectors are generated from existing population members by applying either the DE/rand or the DE/current-to-best strategy [13]. The probability of applying one strategy or another is set to 0.5. During the advancement from parents to offspring, $CR$ and $F$ are adapted in each population according to the Algorithm developed in [11].

**Algorithm 1: MP-AIDEA-ALR**

1: Set values for $n_{pop}$, $N_{pop}$, $\bar{\rho}$, $\delta_{global}$, $q = 1$
2: Set $n_{feval,m} = 0$ and $k_m = 1$ (generation number) for each populations $m \in \{1, \ldots, N_{pop}\}$
3: Initialize population $P_{m,k_m}$ with individuals $x_{m,i,k_m}$ $\forall m \in \{1, \ldots, N_{pop}\}$ and $\forall i \in \{1, \ldots, n_{pop}\}$
4: Compute $\Delta = \|x_{upper} - x_{lower}\|$ where $x_{lower}$ and $x_{upper}$ are the lower and upper boundaries of the search space
5: A regular mesh $CRF$ with $(n_D + 1)^2$ points $(n_D$ is the dimensionality of the problem) in the space $\mathbf{CR} \in [0,1,0.99] \times \mathbf{F} \in [-0.5,1]$ is created
6: Initialize $CRF$ with points of the mesh: $CRF_{j,1} \leftarrow CR_j$ and $CRF_{j,2} \leftarrow F_j$ for all $j \in \{1, \ldots, (n_D + 1)^2\}$
7: Associate to each row of $CRF$ an element $dd_j$ $= 0$ for all $j \in \{1, \ldots, (n_D + 1)^2\}$
8: for $m \in \{1, \ldots, N_{pop}\}$ do
9: Sample $CR_{m,k_m}$ and $F_{m,k_m}$ from a bi-variate distribution on the two dimensional lattice defined by the rows of $CRF$
10: for $i \in \{1, \ldots, n_{pop}\}$ do
11: $x_{m,i,k_m + 1} \leftarrow \text{DE}(x_{m,i,k_m}, CR_{m,k_m}, F_{m,k_m})$
12: $n_{feval,m} = n_{feval,m} + 1$
13: Update $CRF$ (see Ref.[11])
14: end for
15: $k_m = k_m + 1$
16: Row sort $CRF$ in terms of $dd$ values
17: Compute $\rho_m = \max (\|x_{m,i,k_m} - x_{m,j,k_m}\|) \quad \forall x_{m,i,k_m}, x_{m,j,k_m} \in P_{m,k_m}$
18: Until $\rho_m \leq \bar{\rho} \cdot \rho_{max,m}$ goto (9)
19: end for
20: for $m \in \{1, \ldots, N_{pop}\}$ do
21: $x_{best,m} = \arg\min_{x_i,m \in P_m} f(x_{i,m})$
22: if $A = 0$ then
23: $inside = 0$
24: else
25: for $j \in \{1, \ldots, n_{LM}\}$ where $n_{LM}$ is the number of minima $x_{j,M}$ in $A$ do
26: Compute $d_{j} = \|x_{best,m} - x_{LM,j}\|$
27: if $d_{j} < d_{basin,j}$ and $N_{LM Det,j} \geq 4$ then
28: $inside = 1$
29: break
30: else
31: $inside = 0$
32: end if
33: end for
34: end if
35: if $inside = 0$ then
36: L.S.: Run local optimizer from $x_{best,m}$ and let $x_{min}$ be the resulting local minimum
37: if $A = \emptyset$ then
38: detected = 0
39: else
40: detected = 0
41: for $j \in \{1, \ldots, n_{LM}\}$ do
42: Compute $d_{min} = \|x_{min} - x_{LM,j}\|$
43: if $d_{min} \leq \epsilon \Delta$ then
44: $x_{min} \leftarrow x_{LM,j}$
45: $d_{basin,j} = \min (d_{basin,j}, \|x_{best,m} - x_{min}\|)$
46: $N_{LM Det,j} = N_{LM Det,j} + 1$
47: break
48: end if
49: end for
50: end if
51: if detected = 0 then
52: $x_{LM,j+1} = x_{min}, A = A \cup \{x_{LM,j+1}\}$
53: $d_{basin,j+1} = \|x_{best,m} - x_{LM,j+1}\|$ is added to the archive of basin dimensions associated to the elements in $A$: $D_{basin} = D_{basin} \cup \{d_{basin,j+1}\}$
54: $N_{LM Det,j+1} = 1$
55: end if
56: $LR_m = 1$
57: else
58: $LR_m = 0$
59: end if
60: end if
61: end for
62: if All populations did local search L.S. (line 35) for the 1st time or all populations did global restart G.R. (line 75) then
63: Create vector $B$ with Algorithm 3
64: end if
65: if (All populations did local search L.S. more than once) then
66: Update $B$ (see Algorithm 3)
67: end if
68: if $q = q + 1, k_m = 1 \forall m$ then
69: for $m \in \{1, \ldots, N_{pop}\}$ do
70: if $LR_m = 1$ then
71: Sample $\delta_{local,m}$ from $B$ to define the bubble $D_m$
72: L.R.: Initialize population $x_{m,i,k_m}$ for all $i \in \{1, \ldots, n_{pop}\}$ in the bubble $D_m$
73: else
74: Define clusters in the archive and compute baricentre $x_{c,m}$ of each cluster
75: G.R.: Initialize population $x_{m,i,k_m}$ for all $i \in \{1, \ldots, n_{pop}\}$ such that $\forall i,j \|x_{m,i,k_m} - x_{m,j,k_m}\| > \delta_{global}$
76: end if
77: end for
78: Termination If $\sum (n_{feval,m}) \geq n_{feval,max}$ goto (5)
A. Adaptive Local Restart

In MP-AIDEA, when a population contracts below a threshold \( \rho_{\text{max,min}} \) (Alg. 1, line 18), a local optimizer is run from the best individual in the population, \( x_{\text{best}} \), and the resulting local minimum, \( x_{LM,j} \), is saved in an archive of local minima, \( A \), common to all the populations. The population \( m \) is then restarted in a hypercube with edge \( 2\delta_{\text{local,m}} \) around the detected local minimum \( x_{LM,j} \). The dimension \( \delta_{\text{local,m}} \) is drawn from a probability distribution defined by the vector \( B \).

The procedure to initialize \( B \) is described in Algorithm 2 and is analogous to the one used to generate CRF: the distance between all the local minima in the archive \( A \) is computed and the vector \( B \) is initialized with values spanning the interval between the min and the mean distance among minima (Alg. 2, line 1).

**Algorithm 2:** Generation of \( B \) for the adaptation of \( \delta_{\text{local}} 

1. Compute mean and minimum distance between all local minima in \( A \): \( d_{\text{minMIN}} \) and \( d_{\text{minMEAN}} \)
2. Create 1-dimensional regular grid with \((n_D + 1)^2\) points in the interval \([d_{\text{minMIN}}, d_{\text{minMEAN}}]\)
3. Initialize \( B \) with points of the mesh
4. Associate to \( B \) a vector \( \delta d \) with element \( \delta d_{b,j} = 0 \) for all \( j \in [1, \ldots, (n_D + 1)^2] \)

**Algorithm 3:** Updating procedure for \( B \)

1. for \( m \in [1, \ldots, N_{\text{pop}}] \) do
2. \( p_m = \|x_{LM,m,q} - x_{LM,\text{best}}\| \)
3. for \( j \in [1, \ldots, (n_D + 1)^2] \) do
4. if \( \delta d_{b,j} < p_m \wedge f(x_{LM,m,q}) \leq f(x_{LM,\text{best}}) \) then
5. \( B_{j,q} \leftarrow \delta d_{\text{local,m}} \)
6. \( \delta d_{b,j,q} \leftarrow p_m \)
7. end if
8. end for
9. end for
10. Row sort \( B \) according to \( \delta d \) values

The updating procedure for \( B \), detailed in Algorithm 3, follows the same approach used for updating CRF in [11]: the distance \( p_m \) between the local minimum identified by population \( m \) at iteration \( q \), \( x_{LM,m,q} \), and the best local minimum in \( A \), \( x_{LM,\text{best}} \), is used to update \( B \) if \( f(x_{LM,m,q}) \leq f(x_{LM,\text{best}}) \). In MP-AIDEA the local restart following the local search was iterated up to a user-defined number of times \( n_{LR} \). The novelty in MP-AIDEA-ALR is that the parameter \( n_{LR} \) is removed and replaced by a procedure that locally restart the optimization until no new local minimum is found. The idea is that the local restart should enable a transition from the current minimum to a neighboring minimum that falls within the local restart bubble \( \delta_{\text{local,m}} \). If the transition repeatedly leads to the same local minima, the local restart procedure is deemed to be not effective and a global restart becomes necessary. Given a local minimum \( x_{LM,j} \), one can define the basin of attraction of \( x_{LM,j} \) for the local search operator \( LS(x) \) as:

\[
BA(x_{LM,j}) = \{x : LS(x) = x_{LM,j}\}
\]  

If \( BA \) is included in the restart bubble a transition can occur to a new basin of attraction and a new local minimum is saved in \( A \). However, if the restart bubble includes multiple basins of attraction, then a local restart might lead to converging to a local minimum already recorded in the archive \( A \). Furthermore, it is desirable to avoid having the local search operator applied to multiple points belonging to the basin of attraction of the same local minimum \( x_{LM,j} \). In order to avoid rediscovering the same local minima, when the local search operator is applied to \( x_{\text{best}} \), the distance \( d_{\text{basin,j}} \) between \( x_{\text{best}} \) and the local minimum \( x_{LM,j} \), \( x_{\text{best}} \) converged to, is recorded and the number of times \( N_{LM,\text{Det,j}} \) convergence to local minimum \( j \) is achieved is increased by 1. If the same local minimum is revisited more than four times, the local restart procedure is stopped and the global restart is activated. Likewise if \( x_{\text{best}} \) falls at a distance from local minimum \( j \) that is lower than \( \delta_{\text{basin,j}} \), the local restart procedure is replaced by the global restart. If different \( x_{\text{best}} \) converge to the same local minimum, \( d_{\text{basin,j}} \) is the minimum distance of all the \( x_{\text{best}} \) from \( x_{LM,j} \).

III. Case Studies

The effectiveness of the new MP-AIDEA-ALR was tested on some of the functions taken from three past competitions of the Congress on Evolutionary Computation (CEC): CEC 2005 [15], CEC 2011 [16] and CEC 2014 [17]. MP-AIDEA-ALR was compared against those algorithms, competing in each of the CEC competitions, that reported their best result in a paper. The ranking method used to assess the performance of MP-AIDEA-ALR follows the rules of the CEC 2011 Competition, [18]. All algorithms are ranked on the basis of the best and mean values of the objective function criteria according to the following criteria: for each function, algorithms are ranked according to the obtained best objective value; for each function, algorithms are ranked according to the obtained 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. For MP-AIDEA-ALR the number of populations is \( N_{\text{pop}} = 4 \) and the number of individuals in each population is \( n_{\text{pop}} = n_D \), where \( n_D \) is the dimensionality of the problem. The non-adapted parameter of MP-AIDEA-ALR are set to \( \rho = 0.2 \) and \( \delta_{\text{global}} = 0.1 \).

A. CEC 2005 Competition

The functions of the CEC 2005 competition were tested at dimension \( n_D = 10, 30 \) and \( 50 \), with a maximum number of functions evaluation equal to \( n_{\text{eval,max}} = 10000n_D \) and considering 25 independent runs for each function, [15]. All the functions but the noisy ones (functions 4, 17, 24 and 25) were used in this test. The obtained results are reported in Table I, which shows that, for all the considered dimensions, MP-AIDEA-ALR is ranked first. Table II reports the best objective function error values obtained by all the algorithms for functions 13 and 16 at 10 dimensions. According to the CEC 2005 specifications, the accuracy level for the detection of the global minimum is \( 10^{-2} \) for these functions. Note that MP-AIDEA-ALR is able to identify the global minimum of both functions 13 and 16. Previously only EvLib succeeded in identifying the global minimum of function 13 and no other algorithm managed to find the global minimum of function 16.
TABLE I: CEC 2005 Ranking.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Algorithm</th>
<th>Function 13</th>
<th>Function 16</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MP-AIDEA-ALR</td>
<td>7.20e+01</td>
<td>7.20e+01</td>
</tr>
<tr>
<td>2</td>
<td>G-CMA-ES [19]</td>
<td>7.90e+01</td>
<td>7.90e+01</td>
</tr>
<tr>
<td>3</td>
<td>L-SaDE [20]</td>
<td>8.60e+01</td>
<td>8.60e+01</td>
</tr>
<tr>
<td>4</td>
<td>DMS-L-PSO [21]</td>
<td>9.00e+01</td>
<td>9.00e+01</td>
</tr>
<tr>
<td>5</td>
<td>L-CMA-ES [22]</td>
<td>9.00e+01</td>
<td>9.00e+01</td>
</tr>
<tr>
<td>6</td>
<td>BLX-GL50</td>
<td>1.20e+02</td>
<td>1.20e+02</td>
</tr>
<tr>
<td>7</td>
<td>DE (Ronkonnen) [24]</td>
<td>1.50e+02</td>
<td>1.50e+02</td>
</tr>
<tr>
<td>8</td>
<td>DE (Bui)</td>
<td>1.60e+02</td>
<td>1.60e+02</td>
</tr>
<tr>
<td>9</td>
<td>EvLiv [26]</td>
<td>1.90e+02</td>
<td>1.90e+02</td>
</tr>
<tr>
<td>10</td>
<td>CMLSP</td>
<td>2.20e+02</td>
<td>2.20e+02</td>
</tr>
<tr>
<td>11</td>
<td>DE-b6e6rl</td>
<td>2.50e+02</td>
<td>2.50e+02</td>
</tr>
<tr>
<td>12</td>
<td>L-CMA-ES</td>
<td>2.50e+02</td>
<td>2.50e+02</td>
</tr>
<tr>
<td>13</td>
<td>L-SaDE</td>
<td>2.50e+02</td>
<td>2.50e+02</td>
</tr>
<tr>
<td>14</td>
<td>CMLSP</td>
<td>2.50e+02</td>
<td>2.50e+02</td>
</tr>
<tr>
<td>15</td>
<td>L-CMA-ES</td>
<td>2.50e+02</td>
<td>2.50e+02</td>
</tr>
<tr>
<td>16</td>
<td>L-SaDE</td>
<td>2.50e+02</td>
<td>2.50e+02</td>
</tr>
</tbody>
</table>

TABLE II: CEC 2005 Best Objective Function Error Values for Functions 13 and 16, \(n_D = 10\).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Function 13</th>
<th>Function 16</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLX-GL50</td>
<td>3.70e-01</td>
<td>7.20e+01</td>
</tr>
<tr>
<td>BLX-MA</td>
<td>3.80e-01</td>
<td>9.00e+01</td>
</tr>
<tr>
<td>CoEVO</td>
<td>4.70e-01</td>
<td>1.20e+02</td>
</tr>
<tr>
<td>DE (Ronkonnen)</td>
<td>4.60e-01</td>
<td>1.50e+02</td>
</tr>
<tr>
<td>DE (Bui)</td>
<td>2.70e-01</td>
<td>1.00e+02</td>
</tr>
<tr>
<td>DMS-L-PSO</td>
<td>2.50e-01</td>
<td>5.20e+01</td>
</tr>
<tr>
<td>EDA</td>
<td>1.60e+00</td>
<td>1.30e+02</td>
</tr>
<tr>
<td>ES</td>
<td>7.90e-01</td>
<td>9.70e+01</td>
</tr>
<tr>
<td>EvLiv</td>
<td>9.90e-03</td>
<td>1.20e+02</td>
</tr>
<tr>
<td>flexGA</td>
<td>4.20e-02</td>
<td>1.10e+02</td>
</tr>
<tr>
<td>G-CMA-ES</td>
<td>4.10e-01</td>
<td>7.90e+01</td>
</tr>
<tr>
<td>K-PCX</td>
<td>3.30e-01</td>
<td>8.80e+01</td>
</tr>
<tr>
<td>L-CMA-ES</td>
<td>1.90e-01</td>
<td>6.10e+01</td>
</tr>
<tr>
<td>L-SaDE</td>
<td>1.20e-01</td>
<td>8.60e+01</td>
</tr>
<tr>
<td>SPC-PNX</td>
<td>3.50e-01</td>
<td>9.10e+01</td>
</tr>
<tr>
<td>MP-AIDEA-ALR</td>
<td>9.87e-03</td>
<td>0.00e+00</td>
</tr>
</tbody>
</table>

B. CEC 2011 Competition

For the CEC 2011 competition the functions were tested using \(n_{feval,max} = 150000\) function evaluations and considering 25 runs per function, [16]. All problems but functions 4, 8, 9 and 11 were used in this test (functions 8, 9 and 11 are characterized by equality and inequality constraints). For this competition only, the number of individuals in each population was set to \(n_{pop} = 30\) regardless of the dimensionality of the problem. The obtained results are reported in Table III. MP-AIDEA-ALR is ranked in first place if problem 13 (the Cassini 2 Spacecraft Trajectory Optimization Problem) is excluded from the ranking. Figure 1 shows the typical convergence profile of MP-AIDEA-ALR and GA-MPC, the best algorithm of the competition, on function 13 for a number of function evaluations greater than the limit prescribed by the CEC 2011 competition. It has to be noted that GA-MPC converges very rapidly to a local minimum but then seems to stagnate. On the contrary, MP-AIDEA-ALR has a slower convergence for the first 200000 function evaluations but then progressively finds better minima as the number of function evaluations increases.

C. CEC 2014 Competition

For the CEC 2014 competition the functions were tested at dimensions \(n_D = 10, 30, 50\) and 100, with maximum number of function evaluations \(n_{feval,max} = 10000n_D\) and 51 repeated runs per function [17]. Non-differentiable functions 6, 12, 19, 22, 26, 27, 29 and 30 were not included in this test. Results are reported in Table IV.

Table V reports the best objective function values obtained by all the algorithms for functions 9, 10, 11 and 15 in 10 dimensions. MP-AIDEA-ALR detects the global minimum of function 11, unlike all the other competing algorithms, and gives good results for the other functions.
TABLE V: CEC 2014 Best Objective Function Error Values for Functions 9, 10, 11 and 15, \( n_D = 10 \).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Func. 9</th>
<th>Func. 10</th>
<th>Func. 11</th>
<th>Func. 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>b3c3pbest</td>
<td>2.00e+00</td>
<td>0.00e+00</td>
<td>9.50e+00</td>
<td>5.70e+01</td>
</tr>
<tr>
<td>CMLSP</td>
<td>0.00e+00</td>
<td>2.50e-01</td>
<td>3.60e+00</td>
<td>4.50e+00</td>
</tr>
<tr>
<td>DE-bf6erb</td>
<td>2.50e-00</td>
<td>0.00e+00</td>
<td>3.60e+00</td>
<td>4.90e+00</td>
</tr>
<tr>
<td>FCDE</td>
<td>8.00e-00</td>
<td>3.10e-01</td>
<td>1.40e+02</td>
<td>6.50e+01</td>
</tr>
<tr>
<td>FERDE</td>
<td>3.00e+00</td>
<td>0.00e+00</td>
<td>3.80e-01</td>
<td>3.50e-01</td>
</tr>
<tr>
<td>FWA-DM</td>
<td>2.00e+00</td>
<td>9.10e-13</td>
<td>4.00e+01</td>
<td>3.20e-01</td>
</tr>
<tr>
<td>GaAPADE</td>
<td>1.90e+00</td>
<td>2.40e-02</td>
<td>2.40e+00</td>
<td>3.80e+00</td>
</tr>
<tr>
<td>L-SHADE</td>
<td>2.20e+03</td>
<td>0.00e+00</td>
<td>3.90e-01</td>
<td>3.50e-01</td>
</tr>
<tr>
<td>MVMO</td>
<td>9.90e-01</td>
<td>6.20e-02</td>
<td>3.40e+00</td>
<td>2.10e-01</td>
</tr>
<tr>
<td>NRGA</td>
<td>9.90e-01</td>
<td>3.70e+00</td>
<td>1.90e+01</td>
<td>3.70e-01</td>
</tr>
<tr>
<td>OptBees</td>
<td>2.90e+00</td>
<td>3.50e+00</td>
<td>1.30e+02</td>
<td>6.30e-01</td>
</tr>
<tr>
<td>POBL-DAE</td>
<td>1.00e+00</td>
<td>2.20e+01</td>
<td>3.60e+00</td>
<td>1.70e-01</td>
</tr>
<tr>
<td>rmalschma</td>
<td>9.90e-01</td>
<td>6.20e-02</td>
<td>1.90e-01</td>
<td>3.10e-01</td>
</tr>
<tr>
<td>RSDE</td>
<td>2.00e+00</td>
<td>3.50e+00</td>
<td>1.90e+01</td>
<td>3.60e-01</td>
</tr>
<tr>
<td>SOO</td>
<td>9.00e+00</td>
<td>1.30e+02</td>
<td>3.50e+02</td>
<td>4.20e-02</td>
</tr>
<tr>
<td>SOO-BOBYQA</td>
<td>9.00e-01</td>
<td>1.30e+02</td>
<td>3.50e+02</td>
<td>4.20e-02</td>
</tr>
<tr>
<td>UMOEAs</td>
<td>9.90e-01</td>
<td>6.20e-02</td>
<td>3.50e+00</td>
<td>3.20e-01</td>
</tr>
<tr>
<td>MP-AIDEA-ALR</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
<td>0.00e+00</td>
<td>2.60e+00</td>
</tr>
</tbody>
</table>

TABLE VI: CEC 2015 Ranking, \( n_D = 30, \rho = 0.3 \).

<table>
<thead>
<tr>
<th>Case A</th>
<th>( \delta_{\rho = 0.1} )</th>
<th>Case B</th>
<th>( \delta_{\rho = 0.3} )</th>
<th>Case C</th>
<th>( \delta_{\rho = 0.2} )</th>
<th>Case D</th>
<th>( \delta_{\rho = 0.3} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>F_WA-DE</td>
<td>GS-DE</td>
<td>MP-AIDEA-ALR</td>
<td>L-SHADE</td>
<td>MP-AIDEA-ALR</td>
<td>L-SHADE</td>
<td>MP-AIDEA-ALR</td>
</tr>
<tr>
<td>2</td>
<td>L-SHADE</td>
<td>GS-DE</td>
<td>MP-AIDEA-ALR</td>
<td>L-SHADE</td>
<td>MP-AIDEA-ALR</td>
<td>L-SHADE</td>
<td>MP-AIDEA-ALR</td>
</tr>
<tr>
<td>3</td>
<td>GS-DE</td>
<td>BM-DE</td>
<td>MP-AIDEA-ALR</td>
<td>GS-DE</td>
<td>MP-AIDEA-ALR</td>
<td>GS-DE</td>
<td>MP-AIDEA-ALR</td>
</tr>
<tr>
<td>4</td>
<td>BM-DE</td>
<td>GS-DES</td>
<td>MP-AIDEA-ALR</td>
<td>BM-DE</td>
<td>MP-AIDEA-ALR</td>
<td>BM-DE</td>
<td>MP-AIDEA-ALR</td>
</tr>
<tr>
<td>5</td>
<td>GS-DES</td>
<td>BM-DE</td>
<td>MP-AIDEA-ALR</td>
<td>GS-DES</td>
<td>MP-AIDEA-ALR</td>
<td>GS-DES</td>
<td>MP-AIDEA-ALR</td>
</tr>
</tbody>
</table>

1) Sensitivity to \( \bar{\rho} \) and \( \delta_{\text{global}} \): This subsection is devoted to a preliminary analysis of the sensitivity of the performance of MP-AIDEA-ALR to the two parameters that are not automatically adapted: \( \bar{\rho} \) and \( \delta_{\text{global}} \). Table VI shows the ranking obtained when varying \( \bar{\rho} \) and \( \delta_{\text{global}} \) for the 30 dimension test case of the CEC2014 competition, the one in which the performance of MP-AIDEA-ALR were poorest. In particular, case B shows the ranking obtained when using \( \bar{\rho} = 0.3 \) instead of \( \bar{\rho} = 0.1 \). Comparing the results with table IV one can see that with \( \bar{\rho} = 0.3 \) MP-AIDEA-ALR performs better and moves from the fourth to the third position in the ranking. At the same time there seems to be a reduced sensitivity of MP-AIDEA-ALR to the settings of \( \delta_{\text{global}} \).

D. Distance from the Best

Figures from 2 to 9 show the relative difference between the result of MP-AIDEA-ALR and the result of the best performing algorithm for each of the functions used in the tests. Both the difference in best and mean value are reported in each figure.

IV. Conclusion

This paper has introduced Multi-Population Adaptive Inflationary Differential Evolution Algorithm with Adaptive Local Restart, an algorithm based on the hybridization of Differential Evolution with Monotonic Basin Hopping. MP-AIDEA-ALR automatically adapts \( CR \) and \( F \) for the DE heuristics and two of the four parameters controlling the heuristic of the MBH. In particular, in this paper, a mechanism to avoid the multiple detection of the same local minima has been presented. MP-AIDEA-ALR has been tested over more than fifty functions from the CEC competitions. Results showed that the algorithm is averagely very efficient over a large number of function tested on different dimensions. MP-AIDEA-ALR was indeed always in the first three position of the algorithm ranking, except for the 30 dimension test cases of CEC 2014.
Fig. 4: Relative difference between MP-AIDEA-ALR and the best performing algorithm for each of the selected functions of the CEC2005 competition at 50D

Fig. 5: Relative difference between MP-AIDEA-ALR and the best performing algorithm for each of the selected functions of the CEC2011 competition

Fig. 6: Relative difference between MP-AIDEA-ALR and the best performing algorithm for each of the selected functions of the CEC2014 competition at 10D

Fig. 7: Relative difference between MP-AIDEA-ALR and the best performing algorithm for each of the selected functions of the CEC2014 competition at 30D

ACKNOWLEDGMENT

This research was partially supported by a grant by Airbus Defence & Space. The authors would like to thank Mr. Stephen Kemble for his support and contribution.

REFERENCES


Fig. 8: Relative difference between MP-AIDEA-ALR and the best performing algorithm for each of the selected functions of the CEC2014 competition at 50D

Fig. 9: Relative difference between MP-AIDEA-ALR and the best performing algorithm for each of the selected functions of the CEC2014 competition at 100D


