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.

Marilena Di Carlo, Massimiliano Vasile and Edmondo Minisci are with the Department of Mechanical and Aerospace Engineering, University of Strathclyde, Glasgow, UK (e-mail: marilena.di-carlo@strath.ac.uk, massimiliano.vasile@strath.ac.uk, edmondo.minisci@strath.ac.uk).

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 problem 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 δ_{local} around the local minimum (*local restart*). The parameter δ_{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 δ_{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}, $\bar{\rho}$, δ_{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 $\mathbf{x}_{m,i,k_m}$ $\forall m \in [1, \ldots, N_{pop}]$ 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: A regular mesh **CRF** with $(n_D + 1)^2$ points (n_D is the dimensionality of the problem) in the space $CR \in [0.1, 0.99] \times F \in [-0.5, 1]$ is created
6: Initialize **CRF** with points of the mesh: $\mathbf{CRF}_{j,1} \leftarrow \mathbf{CR}_j$ and $\mathbf{CRF}_{j,2} \leftarrow \mathbf{F}_j$ 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 $\mathbf{CR}_{m,k_m}$ and $\mathbf{F}_{m,k_m}$ from a bi-variate distribution on the two dimensional lattice defined by the rows of **CRF**
10: **for** $i \in [1, \ldots, n_{pop}]$ **do**
11: $\mathbf{x}_{m,i,k_m+1} \leftarrow$ DE($\mathbf{x}_{m,i,k_m}$, $\mathbf{CR}_{m,k_m}$, $\mathbf{F}_{m,k_m}$)
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 $\rho_m = \max(||\mathbf{x}_{m,i,k_m} - \mathbf{x}_{m,j,k_m}||)$ $\forall \mathbf{x}_{m,i,k_m}, \mathbf{x}_{m,j,k_m} \in P_{m,k_m}$
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: $\mathbf{x}_{best,m} = argmin_{\mathbf{x}_{i,m} \in P_m} f(\mathbf{x}_{i,m})$
22: **if** $A = \emptyset$ **then**
23: $inside = 0$
24: **else**
25: **for** $j \in [1, \ldots, n_{LM}]$ where n_{LM} is the number of minima $\mathbf{x}_{LM}$ in A **do**
26: Compute $d_j = \|\mathbf{x}_{best,m} - \mathbf{x}_{LM,j}\|$
27: **if** $d_j < d_{basin,j}$ **and** $N_{LMDet,j} \geq 4$ **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 $\mathbf{x}_{best,m}$ and let $\mathbf{x}_{min}$ 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 $d_{min} = \|\mathbf{x}_{min} - \mathbf{x}_{LM,j}\|$
43: **if** $d_{min} \leq \epsilon\Delta$ **then**
44: $\mathbf{x}_{min} \leftarrow \mathbf{x}_{LM,j}$
45: $d_{basin,j} = \min(d_{basin,j}, \|\mathbf{x}_{best,m} - \mathbf{x}_{min}\|)$
46: $N_{LMDet,j} = N_{LMDet,j} + 1$
47: $detected = 1$
48: **break**
49: **end if**
50: **end for**
51: **end if**
52: **if** $detected = 0$ **then**
53: $\mathbf{x}_{LM,j+1} = \mathbf{x}_{min}$, $A = A \cup \{\mathbf{x}_{LM,j+1}\}$
54: $d_{basin,j+1} = \|\mathbf{x}_{best,m} - \mathbf{x}_{LM,j+1}\|$ is added to the archive of basin dimensions associated to the elements in A: $D_{basin} = D_{basin} \cup \{d_{basin,j+1}\}$
55: $N_{LMDet,j+1} = 1$
56: **end if**
57: $LR_m = 1$
58: **else**
59: $LR_m = 0$
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 $\mathbf{B}$ with Algorithm 3
64: **end if**
65: **if** (All populations did local search L.S. more than once) **then**
66: Update $\mathbf{B}$ (see Algorithm 3)
67: **end if**
68: $q = q + 1$, $k_m = 1\ \forall m$
69: **for** $m \in [1, \ldots, N_{pop}]$ **do**
70: **if** $LR_m = 1$ **then**
71: Sample $\delta_{local,m}$ from $\mathbf{B}$ to define the bubble D_m
72: L.R.: Initialize population $\mathbf{x}_{m,i,k_m}$ for all $i \in [1, \ldots, n_{pop}]$ in the bubble D_m
73: **else**
74: Define clusters in the archive and compute baricentre $\mathbf{x}_{c,m}$ of each cluster
75: G.R.: Initialize population $\mathbf{x}_{m,i,k_m}$ for all $i \in [1, \ldots, n_{pop}]$ such that $\forall i, j ||\mathbf{x}_{m,i,k_m} - \mathbf{x}_{m,j,k_m}|| > \delta_{global}$
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 $\bar{\rho}\rho_{max,m}$ (Alg. 1, line 18), a local optimizer is run from the best individual in the population, $\mathbf{x}_{best}$, and the resulting local minimum, $\mathbf{x}_{LM}$, 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 $2\delta_{local,m}$ around the detected local minimum $\mathbf{x}_{LM}$. The dimension $\delta_{local,m}$ is drawn from a probability distribution defined by the vector $\mathbf{B}$. The procedure to initialize $\mathbf{B}$ is described in Algorithm 2 and is analogous to the one used to generate $\mathbf{CRF}$: the distance between all the local minima in the archive A is computed and the vector $\mathbf{B}$ is initialize with values spanning the interval between the min and the mean distance among minima (Alg. 2, line 1).

Algorithm 2: Generation of $\mathbf{B}$ for the adaptation of δ_{local}

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

Algorithm 3: Updating procedure for $\mathbf{B}$

for $m \in [1, \dots, N_{pop}]$ **do**
  $p_m = ||\mathbf{x}_{LM,m,q} - \mathbf{x}_{LM,best}||$
  for $j \in [1, \dots, (n_D+1)^2]$ **do**
    if $\mathbf{dd}_{bj} < \ p_m \wedge f(\mathbf{x}_{LM,m,q}) \leq f(\mathbf{x}_{LM,best})$ **then**
      $\mathbf{B}_{j,q} \leftarrow \delta_{local,m}$
      $\mathbf{dd}_{b,j,q} \leftarrow p_m$
    end if
  end for
end for
Row sort $\mathbf{B}$ according to $\mathbf{dd}_b$ values

The updating procedure for $\mathbf{B}$, detailed in Algorithm 3, follows the same approach used for updating $\mathbf{CRF}$ in [11]: the distance p_m between the local minimum identified by population m at iteration q, $\mathbf{x}_{LM,m,q}$, and the best local minimum in A, $\mathbf{x}_{LM,best}$, is used to update $\mathbf{B}$ if $f(\mathbf{x}_{LM,m,q}) \leq f(\mathbf{x}_{LM,best})$. 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 $\delta_{local,m}$. 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 $\mathbf{x}_{LM,j}$, one can define the basin of attraction of $\mathbf{x}_{LM,j}$ for the local search operator $LS(\mathbf{x})$ as:

$$BA(\mathbf{x}_{LM,j})_j = \{\mathbf{x} : LS(\mathbf{x}) = \mathbf{x}_{LM,j}\} \tag{1}$$

If BA_j 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 $\mathbf{x}_{LM,j}$. In order to avoid rediscovering the same local minima, when the local search operator is applied to $\mathbf{x}_{best}$, the distance $d_{basin,j}$ between $\mathbf{x}_{best}$ and the local minimum $\mathbf{x}_{LM,j}$, $\mathbf{x}_{best}$ converged to, is recorded and the number of times $N_{LMDet,j}$ 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 $\mathbf{x}_{best}$ falls at a distance from local minimum j that is lower than $d_{basin,j}$, the local restart procedure is replaced by the global restart. If different $\mathbf{x}_{best}$ converge to the same local minimum, $d_{basin,j}$ is the minimum distance of all the $\mathbf{x}_{best}$ from $\mathbf{x}_{LM,j}$.

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 $N_{pop} = 4$ and the number of individuals in each population is $n_{pop} = n_D$, where n_D is the dimensionality of the problem. The non-adapted parameter of MP-AIDEA-ALR are set to $\bar{\rho} = 0.2$ and $\delta_{global} = 0.1$.

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 $n_{feval,max} = 10000 n_D$ 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.

	$n_D = 10$	$n_D = 30$	$n_D = 50$
1	**MP-AIDEA-ALR**	**MP-AIDEA-ALR**	**MP-AIDEA-ALR**
2	G-CMA-ES [19]	G-CMA-ES	G-CMA-ES
3	L-SaDE [20]	L-CMA-ES	L-CMA-ES
4	DMS-L-PSO [21]	K-PCX	flexGA
5	L-CMA-ES [22]	BLX-GL50	
6	BLX-GL50 [23]	SPC-PNX	
7	DE (Ronkonnen) [24]	DE (Ronkonnen)	
8	SPC-PNX [25]	DE (Bui)	
9	EvLiv [26]	flexGA	
10	EDA [27]	CoEVO	
11	K-PCX [28]	EDA	
12	BLX-MA [29]		
13	DE (Bui) [30]		
14	CoEVO [31]		
15	flexGA [32]		
16	ES [33]		

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

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

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 $n_{feval,max} = 10000 n_D$ 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 III: CEC 2011 Ranking.

	With Function 13	Without Function 13
1	GA-MPC [34]	**MP-AIDEA-ALR**
2	**MP-AIDEA-ALR**	GA-MPC
3	SAMODE [35]	SAMODE
4	WI-DE [36]	WI-DE
5	EA-DE-MA [37]	EA-DE-MA
6	Adap. DE 171 [38]	Adap. DE 171
7	DE-Λ [39]	DE-Λ
8	ED-DE [40]	ED-DE
9	Mod-DE-LS [41]	Mod-DE-Ls
10	RGA [42]	RGA
11	DE-RHC [43]	DE-RHC
12	mSBX-GA [44]	mSBX-GA
13	ENSML-DE [45]	ENSML-DE
14	CDASA [46]	CDASA

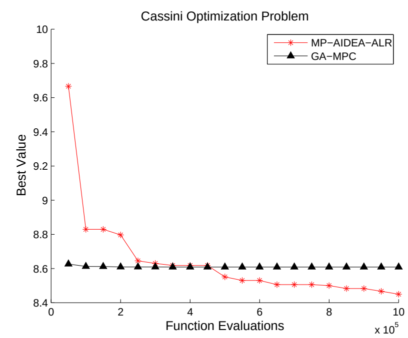

Fig. 1: F12: Cassini Optimization Problem

TABLE IV: CEC 2014 Ranking.

	$n_D = 10$	$n_D = 30$	$n_D = 50$	$n_D = 100$
1	UMOEAs [47]	L-SHADE	UMOEAs	UMOEAs
2	**MP-A.-ALR**	UMOEAs	**MP-A.-ALR**	L-SHADE
3	L-SHADE [48]	GaAPADE	MVMO	**MP-A.-ALR**
4	MVMO [49]	**MP-A.-ALR**	L-SHADE	rmalshcma
5	DE-b6e6rl [50]	CMLSP	rmalshcma	POBL-ADE
6	rmalschma [51]	rmalshcma	b3e3pbest	b3e3pbest
7	GaAPADE [52]	MVMO	FERDE	OptBees
8	FERDE [53]	DE-b6e6rl	DE-b6e6rl	DE-b6e6rl
9	CMLSP [54]	b3e3pbest	RSDE	RSDE
10	b3e3pbest [55]	FERDE	POBL-ADE	FWA-DE
11	RSDE [56]	RSDE	OptBees	
12	FWA-DE [57]	FWA-DE	FWA-DE	
13	POBL-ADE [58]	POBL-ADE	SOO	
14	OptBees [59]	OptBees		
15	SOO-BOBYQA [60]	SOO-BOBYQA		
16	FCDE [61]	NRGA		
17	NRGA [62]	FCDE		
18	SOO [60]	SOO		

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, $n_D = 10$.

Algorithm	Func. 9	Func. 10	Func. 11	Func. 15
b3e3pbest	2.60e+00	0.00e+00	9.50e+01	5.70e-01
CMLSP	0.00e+00	2.50e-01	3.60e+00	4.50e-01
DE-b6e6rl	2.50e+00	0.00e+00	3.60e+01	4.90e-01
FCDE	8.00e+00	3.10e-01	1.40e+02	6.50e-01
FERDE	3.00e+00	0.00e+00	3.80e-01	3.50e-01
FWA-DM	2.00e+00	9.10e-13	4.00e+01	3.20e-01
GaAPADE	1.90e+00	2.40e-02	2.40e+00	3.80e-01
L-SHADE	2.20e-03	0.00e+00	3.90e-01	2.10e-01
MVMO	9.90e-01	6.20e-02	3.40e+00	2.10e-01
NRGA	9.90e-01	3.70e+00	1.90e+01	3.70e-01
OptBees	2.00e+00	3.50e+00	1.30e+02	6.30e-01
POBL-ADE	1.00e+00	2.20e+01	3.60e+00	1.70e-01
rmalschma	9.90e-01	6.20e-02	1.90e-01	3.10e-01
RSDE	2.00e+00	3.50e+00	1.90e+01	3.60e-01
SOO	9.00e+00	1.30e+02	3.50e+02	4.40e-01
SOO-BOBYQA	9.00e+00	1.30e+02	3.50e+02	4.20e-01
UMOEAs	9.90e-01	6.20e-02	3.50e+00	3.20e-01
MP-AIDEA-ALR	**0.00e+00**	**0.00e+00**	**0.00e+00**	**2.00e-02**

TABLE VI: CEC 2014 Ranking, $n_D = 30$, $\rho = 0.3$.

	Case A	Case B	Case C	Case D
	$\bar{\rho} = 0.1$	$\bar{\rho} = 0.3$	$\bar{\rho} = 0.2$	$\bar{\rho} = 0.2$
	$\delta_{global} = 0.1$	$\delta_{global} = 0.1$	$\delta_{global} = 0.2$	$\delta_{global} = 0.3$
1	L-SHADE	L-SHADE	L-SHADE	L-SHADE
2	UMOEAs	UMOEAs	UMOEAs	UMOEAs
3	GaAPADE	**MP-A.-ALR**	GaAPADE	GaAPADE
4	**MP-A.-ALR**	GaAPADE	**MP-A.-ALR**	**MP-A.-ALR**
5	CMLSP	CMLSP	CMLSP	CMLSP
6	rmalschma	rmalschma	rmalschma	rmalschma
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

1) Sensitivity to $\bar{\rho}$ and δ_{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 δ_{global}. Table VI shows the ranking obtained when varying $\bar{\rho}$ and δ_{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 δ_{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.

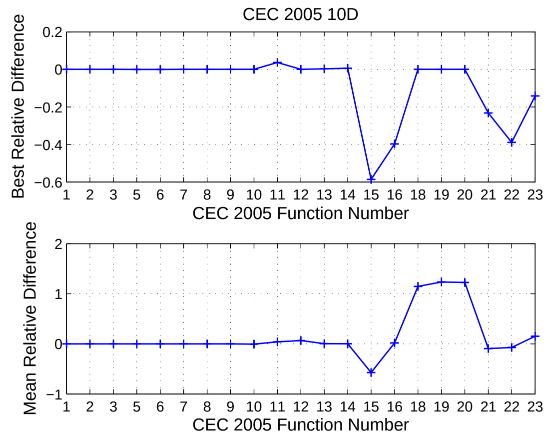

Fig. 2: Relative difference between MP-AIDEA-ALR and the best performing algorithm for each of the selected functions of the CEC2005 competition at 10D

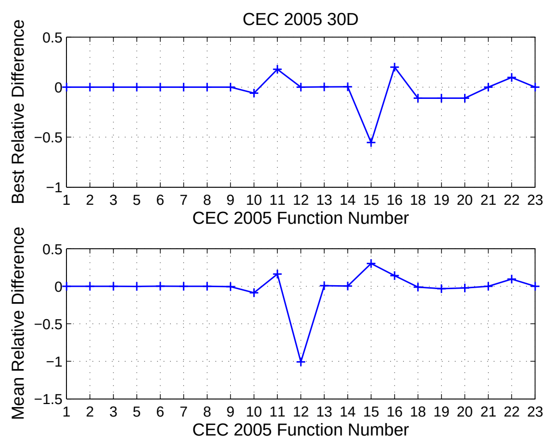

Fig. 3: Relative difference between MP-AIDEA-ALR and the best performing algorithm for each of the selected functions of the CEC2005 competition at 30D

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.

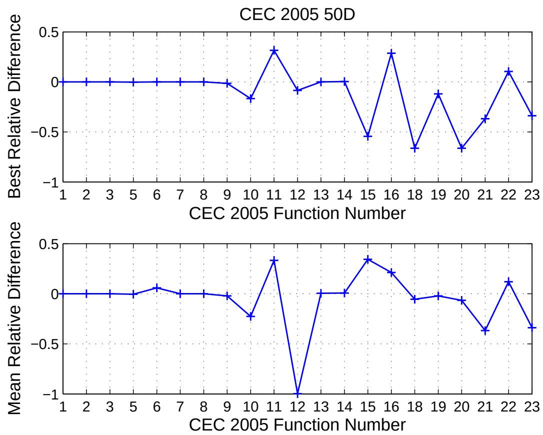

Fig. 4: Relative difference between MP-AIDEA-ALR and the best performing algorithm for each of the selected functions of the CEC2005 competition at 50D

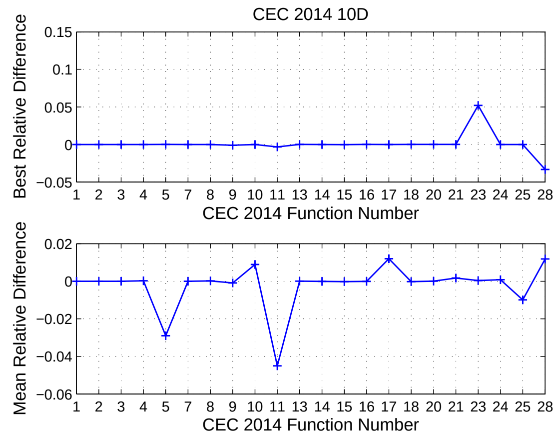

Fig. 6: Relative difference between MP-AIDEA-ALR and the best performing algorithm for each of the selected functions of the CEC2014 competition at 10D

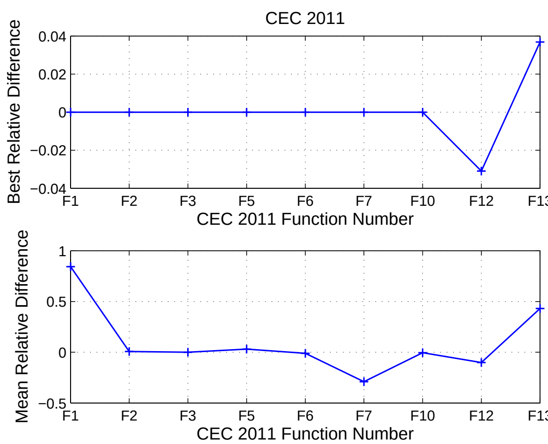

Fig. 5: Relative difference between MP-AIDEA-ALR and the best performing algorithm for each of the selected functions of the CEC2011 competition

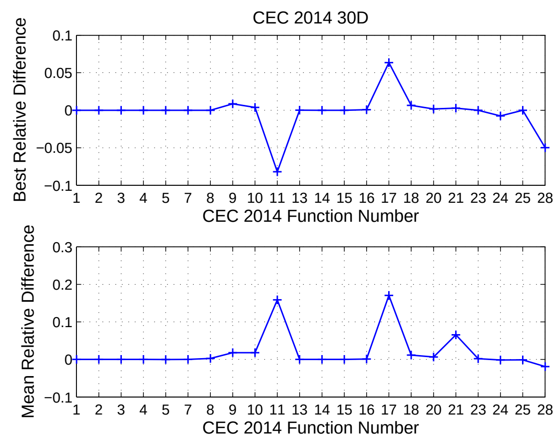

Fig. 7: Relative difference between MP-AIDEA-ALR and the best performing algorithm for each of the selected functions of the CEC2014 competition at 30D

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

[1] K. V. Price, R. M. Storn, and J. A. Lampinen. *Differential Evolution. A Practical Approach to Global Optimization*. Natural Computing Series, Springer, 2005.

[2] D. Swagatam and P. N. Suganthan. *Differential Evolution: A Survey of the State-of-the-Art*. IEEE Transaction on Evolutionary Computation, Vol. 15, No.1, pp. 4-31, Febraury 2011.

[3] A. Qinq. *Differential Evolution: Fundamentals and Applications in Electrical Engineering*. John Wiley and Sons, 2009.

[4] M. Vasile, E. Minisci, and M. Locatelli. *An Inflationary Differential Evolution Algorithm for Space Trajectory Optimization*. IEEE Transaction on Evolutionary Computation, Vol. 15, No. , pp. 267-281, 2011.

[5] M. Locatelli, M. Vasile.*(Non) convergence results for the differential evolution method*. Optimization Letters October 2014.DOI 10.1007/s11590-014-0816-9.

[6] D.J. Wales, and J.P.K. Doye. *Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms*. Journal of Physycal Chemistry, Vol. 10, No. , pp. 5111–5116, 1997.

[7] R. Gamperle, S.D. Muller, and P. Koumoutsakos. *A Parameter Study for Differential Evolution*. Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation. pp. 293-298, 2002.

[8] J. Liu and J. Lampinen. *A Fuzzy Adaptive Differential Evolution Algorithm*. Journal of Soft Computing, Vol. 9, pp. 448-462, 2005.

[9] J. Brest, S. Greiner, B. Boskovic, M. Mernick and V. Zumer. *Self-adapting Control Parameters in Differential Evolution: a Comparative*

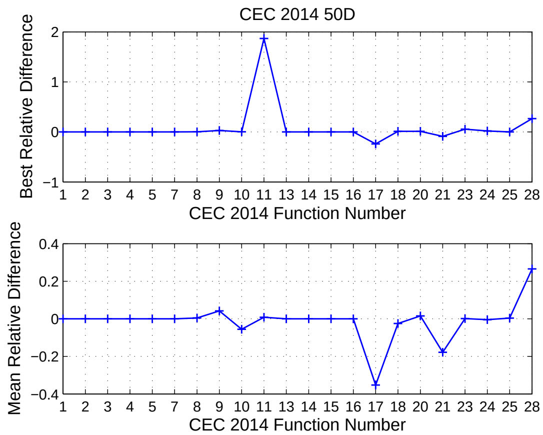

Fig. 8: Relative difference between MP-AIDEA-ALR and the best performing algorithm for each of the selected functions of the CEC2014 competition at 50D

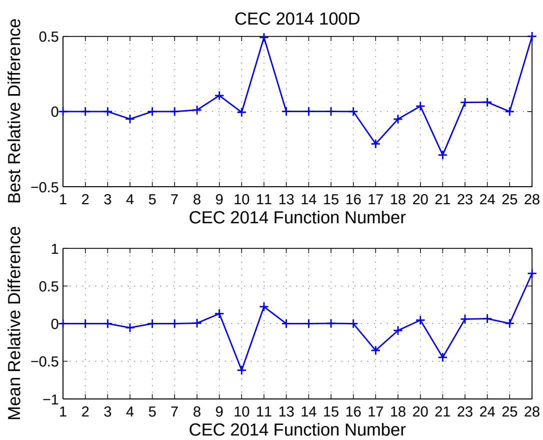

Fig. 9: Relative difference between MP-AIDEA-ALR and the best performing algorithm for each of the selected functions of the CEC2014 competition at 100D

Study on Numerical Benchmark Problems. IEEE Transaction of Evolutionary Computation, Vol. 10, pp. 646-657. 2006.

[10] M. G. H. Omran, A. Salman and A. P. Engelbrecht. *Self-adaptive Differnetial Evolution*. Lecture Notes in Artificial Intelligence, pp. 192-199, 2005.

[11] E. Minisci, and M. Vasile. *Adaptive Inflationary Differential Evolution*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, China, July 2014.

[12] M. Di Carlo, M. Vasile and E. Minisci. *Mutli-Population Adaptive Inflationary Differential Evolution*. Bio-inspired Optimization Methods and their Applications, BIOMA 2014, Ljubljana, September 2014.

[13] R. Storn. *On the Usage of Differential Evolution for Function Optimization*. Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS), IEEE, pp. 519-523, Berkeley, 1996.

[14] D. H. Wolpert and W; G. Macready. *No Free Lunch Theorems for Optimization*. IEEE Transaction on Evolutionary Computation, Vol. 1, No. 1, pp. 67-83, April 1997.

[15] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari. *Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization*. Technical Report, 2005.

[16] S. Das and P. N. Suganthan. *Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems*. Technical Report, December 2010.

[17] J. J. Liang, B. Y. Qu, P. N. Suganthan. *Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization*. Technical Report, December 2013.

[18] P. N. Suganthan. *Testing Evolutionary Algorithms on Real World Numerical Optimization Problems*.

[19] A. Auger, S. Kern and N. Hansen. *A Restart CMA Evolution Strategy with Increasing Population Size*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, pp. 1769-1776, Edinburgh, September 2005.

[20] A. K. Qin and P. N. Suganthan. *Self-adaptive Differential Evolution Algorithm for Numerical Optimization*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, pp. 1785-1791, Edinburgh, September 2005.

[21] J. J. Liang and P. N. Suganthan. *Dynamic Multi-Swarm Particle Swarm Optimizer with Local Search*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, pp. 522-528, Edinburgh, September 2005.

[22] A. Auger, S. Kern and N. Hansen. *Performance Evaluation of an Advanced Local Search Evolutionary Algorithm*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, pp. 1777-1784, Edinburgh, September 2005.

[23] C. Garcia-Martinez and M. Lozano. *Hybrid Real-Coded Genetic Algorithms with Female and Male Differentiation*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, pp. 896-903, Edinburgh, September 2005.

[24] J. Ronkkonen, S. Kukkonen and K. V. Price. *Real-Parameter Optimization with Differential Evolution*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, pp. 506-513, Edinburgh, September 2005.

[25] P. J. Ballester, J. Stephenson, J. N. Carter, K. Gallagher. *Real-Parameter Optimization Performance Study on the CEC-2005 benchmark with SPC-PNX*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, Edinburgh, September 2005.

[26] W. Becker, X. Yu and J. Tu. *EvLib: A Parameterless Self-Adaptive Real-Valued Optimisation Library*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, pp. 506-515, Edinburgh, September 2005.

[27] B. Yuan and M. Gallagher. *Experimental Results for the Special Session on Real-Parameter Optimization at CEC 2005: A Simple, Continuous EDA*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, pp. 1096-1103, Edinburgh, September 2005.

[28] A. Sinha, S. Tiwari and K. Deb. *A Population-Based, Steady-State Procedure for Real-Parameter Optimization*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, pp.514-521, Edinburgh, September 2005.

[29] D. Molina, F. Herrera, M. Lozano. *Adaptive Local Search Parameters for Real-Coded Memetic Algorithms*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005,pp. 888-895, Edinburgh, September 2005.

[30] L. T. Bui, Y. Shan, F. Qi, and H. A. Abbass. *Comparing Two Versions of Differential Evolution in Real Parameter Optimization*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, pp. ,Edinburgh, September 2005.

[31] P. Posik. *Real Parameter Optimisation Using Mutation Step Co-evolution*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, pp. 872-879, Edinburgh, September 2005.

[32] S. Alonso, J. Jimenez, H. Carmona, B. Galvan and G. Winter. *Performance of a Flexible Evolutionary algorithm*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, pp. Edinburgh, September 2005.

[33] L. Costa. *A Parameterless Evolution Strategy for Global Optimization*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2005, Edinburgh, September 2005.

[34] S. Elsayed, R. Sarker and D. Essam. *GA with a New Multi-Parent Crossover for Solving IEEE-CEC2011 Competition Problems*. Proceeding of the IEEE Congress on Evolutionary Computation, CEC 2011, pp. 1034-1040, New Orleans, June 2011.

[35] S. Elsayed, R. Sarker and D. Essam. *Differential Evolution with Multiple Strategies for Solving CEC2011 Real-world Numerical Optimization Problems*. Proceeding of the IEEE Congress on Evolutionary Computation, CEC 2011, pp. 1034-1040, New Orleans, June 2011.

[36] U. Haider, S. Das, D. Maity, A. Abraham and P. Dasgupta. *Self Adaptive Cluster Based and Weed Inspired Differential Evolution Algorithm For Real World Optimization*. Proceeding of the IEEE Congress on Evolutionary Computation, CEC 2011, pp. 750-756, New Orleans, June 2011.

[37] H. K. Singh and T. Ray. *Performance of a Hybrid EA-DE-Memetic Algorithm on CEC 2011 Real World Optimization Problems*. Proceeding of the IEEE Congress on Evolutionary Computation, CEC 2011, pp. 1322-1326, New Orleans, June 2011.

[38] M. Asafuddoula, T. Ray and R. Sarker. *An Adaptive Differential Evolution Algorithm and its Performance on Real World Optimization Problems*. Proceeding of the IEEE Congress on Evolutionary Computation, CEC 2011, pp. 1057-1062, New Orleans, June 2011.

[39] G. Reynoso-Meza, J. Sanchis, X. Blasco and J. M. Herrero. *Hybrid DE Algorithm With Adaptive Crossover Operator For Solving Real-World Numerical Optimization Problems*.Proceeding of the IEEE Congress on Evolutionary Computation, CEC 2011, pp. 1551-1556, New Orleans, June 2011.

[40] Y. Wang, B. Li and K. Zhang. *Estimation of Distribution and Differential Evolution Cooperation for Real-world Numerical Optimization Problems*. Proceeding of the IEEE Congress on Evolutionary Computation, CEC 2011, pp. 1315-1321, New Orleans, June 2011.

[41] M. Ankush, A. K. Das, P. Mukherjee, S. Das and P. N. Suganthan. *Modified Differential Evolution with Local Search Algorithm for Real World Optimization*. Proceeding of the IEEE Congress on Evolutionary Computation, CEC 2011, pp. 1565-1572, New Orleans, June 2011.

[42] A. Saha and T. Ray. *How does the good old Genetic Algorithm fare at Real World Optimization?*. Proceeding of the IEEE Congress on Evolutionary Computation, CEC 2011, pp. 1049-1056, New Orleans, June 2011.

[43] A. La Torre, S. Muelas and J. Pena. *Benchmarking a Hybrid DE-RHC Algorithm on Real World Problems*. Proceeding of the IEEE Congress on Evolutionary Computation, CEC 2011, pp. 1027-1033, New Orleans, June 2011.

[44] S. Bandaru, R. Tulshyan and K. Deb. *Modified SBX and Adaptive Mutation for Real World Single Objective Optimization*. Proceeding of the IEEE Congress on Evolutionary Computation, CEC 2011, pp. 1335-1342, New Orleans, June 2011.

[45] R. Mallipeddi and P. N. Suganthan. *Ensemble Differential Evolution Algorithm for CEC2011 Problems*. Proceeding of the IEEE Congress on Evolutionary Computation, CEC 2011, pp. 1557-1564, New Orleans, June 2011.

[46] P. Korosec and J. Silc. *The Continuous Differential Ant-Stigmergy Algorithm Applied to Real-World Optimization Problems*. Proceeding of the IEEE Congress on Evolutionary Computation, CEC 2011, pp. 1327-1334, New Orleans, June 2011.

[47] S. M. Elsayed, R. A. Sarker, D. L. Essam and N. M. Hamza. *Testing United Multi-Operator Evolutionary Algorithms on the CEC2014 Real-Parameter Numerical Optimization*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, pp. 1650-1657, Beijing, China, July 2014.

[48] R. Tanabe and A. S. Fukunaga. *Improving the Search Performance of SHADE Using Linear Population Size Reduction*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, pp. 1658-1665, Beijing, China, July 2014.

[49] I. Erlich, J. L. Rueda and S. Wildenhues. *Evaluating the Mean-Variance Mapping Optimization on the IEEE-CEC 2014 Test Suite*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, pp. 1625-1632, Beijing, China, July 2014.

[50] R. Polakova, J. Tvrdk and P. Bujok. *Controlled Restart in Differential Evolution Applied to CEC2014 Benchmark Functions*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, pp. 2230-2236, China, July 2014.

[51] D. Molina, B. Lacroix and F. Herrera. *Influence of regions on the memetic algorithm for the CEC2014 Special Session on Real-Parameter Single Objective Optimisation*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, pp. 1633-1640, Beijing, China, July 2014.

[52] R. Mallipeddi, G. Wu, M. Lee and P. N. Suganthan. *Gaussian Adaptation based Parameter Adaptation for Differential Evolution*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, pp. 1760-1767, Beijing, China, July 2014.

[53] B. Y. Qu, J. J. Liang, J. M. Xiao and Z. G. Shang. *Memetic Differential Evolution Based on Fitness Euclidean-Distance Ratio*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, pp. 2266-2273, Beijing, China, July 2014.

[54] L. Chen, H. L. Liu, Z. Zheng and S. Xie. *An Evolutionary Algorithm Based on Covariance Matrix Leaning and Searching Preference for Solving CEC 2014 Benchmark Problems*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, pp. 2672-2677, Beijing, China, July 2014.

[55] P. Bujok, J. Tvrdk and R. Polakov. *Differential Evolution with Rotation-Invariant Mutation and Competing-Strategies Adaptation*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, pp. 2253-2258, China, July 2014.

[56] C. J. Xu, H. Huang and S. J. Ye. *A Differential Evolution with Replacement Strategy for Real-Parameter Numerical Optimization*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, pp. 1617-1624, Beijing, China, July 2014.

[57] C. Yu, L. Kelley, S. Zheng and Y. Tan. *Fireworks Algorithm with Differential Mutation for Solving the CEC 2014 Competition Problems*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, pp. 3238-3245, Beijing, China, July 2014.

[58] Z. Hu, Y. Bao and T. Xiong. *Partial Opposition-Based Adaptive Differential Evolution Algorithms: Evaluation on the CEC 2014 Benchmark Set for Real-parameter Optimization*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, pp. 2259-2265, Beijing, China, July 2014.

[59] R. D. Maia, L. N. de Castro and W. M. Caminhas. *Real-Parameter Optimization with OptBees*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, pp. 2649-2655, Beijing, China, July 2014.

[60] P. Preux, R. Munos and M. Valko. *Bandits attack function optimization*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, pp. 2245-2252, Beijing, China, July 2014.

[61] Z. Li, Z. Shang, B. Y. Qu and J. J. Liang. *Differential Evolution Strategy based on the Constraint of Fitness Values Classification*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, pp. 1454-1460, Beijing, China, July 2014.

[62] D. Yashesh, K. Deb and S. Bandaru. *Non-Uniform Mapping in Real-Coded Genetic Algorithms*. Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, pp. 2237-2244, Beijing, China, July 2014.