View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of Birmingham Research Portal

UNIVERSITY OF BIRMINGHAM

Research at Birmingham

Opposition-based Magnetic Optimization Algorithm with parameter adaptation strategy

Aziz, Mahdi; Tayaraninajaran, Mohammad

DOI:
10.1016/j.swevo.2015.09.001

License:
Creative Commons: Attribution-NonCommercial-NoDerivs (CC BY-NC-ND)

Document Version
Peer reviewed version

Citation for published version (Harvard):
Aziz, M & Tayaraninajaran, M 2015, 'Opposition-based Magnetic Optimization Algorithm with parameter adaptation strategy', Swarm and Evolutionary Computation. https://doi.org/10.1016/j.swevo.2015.09.001

Link to publication on Research at Birmingham portal

Publisher Rights Statement:
Eligibility for repository: Checked on 29/10/2015

General rights
Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

• Users may freely distribute the URL that is used to identify this publication.
• Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
• User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
• Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy
While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

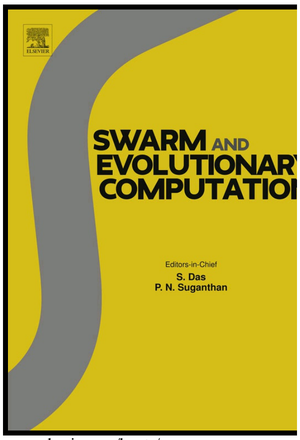
If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

Download date: 01. Feb. 2019

Author's Accepted Manuscript

Opposition-Based Magnetic Optimization Algorithm With Parameter Adaptation Strategy

Mahdi Aziz, Mohammad - H. Tayarani- N.



www.elsevier.com/locate/swevo

PII: S2210-6502(15)00068-1
DOI: http://dx.doi.org/10.1016/j.swevo.2015.09.001
Reference: SWEVO168

To appear in: *Swarm and Evolutionary Computation*

Received date: 7 December 2014
Revised date: 16 July 2015
Accepted date: 3 September 2015

Cite this article as: Mahdi Aziz and Mohammad - H. Tayarani- N., Opposition Based Magnetic Optimization Algorithm With Parameter Adaptation Strategy *Swarm and Evolutionary Computation* http://dx.doi.org/10.1016/j.swevo.2015.09.001

This is a PDF file of an unedited manuscript that has been accepted fo publication. As a service to our customers we are providing this early version o the manuscript. The manuscript will undergo copyediting, typesetting, an review of the resulting galley proof before it is published in its final citable form Please note that during the production process errors may be discovered whic could affect the content, and all legal disclaimers that apply to the journal pertain

ACCEPTED MANUSCRIPT

Available online at www.sciencedirect.com

ScienceDirect

Science Direct

Swarm and Evolutionary Computation xx (2015) 1–34

Opposition-Based Magnetic Optimization Algorithm With Parameter Adaptation Strategy

Mahdi Aziz[a,*], Mohammad - H. Tayarani- N.[b]

[a] *Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran*
[b] *Department of Electrical and Computer Science, University of Birmingham, Birmingham, Uk*

Accepted manuscript

Abstract

Magnetic Optimization Algorithm (MOA) has emerged as a promising optimization algorithm that is inspired by the principles of magnetic field theory. In this paper we improve the performance of the algorithm in two aspects. First an Opposition-Based Learning (OBL) approach is proposed for the algorithm which is applied to the movement operator of the algorithm. Second, by learning from the algorithm's past experience, an adaptive parameter control strategy which dynamically sets the parameters of the algorithm during the optimization is proposed. To show the significance of the proposed parameter adaptation strategy, we compare the algorithm with two well-known parameter setting techniques on a number of benchmark problems. The results indicate that although the proposed algorithm with the adaptation strategy does not require to set the parameters of the algorithm prior to the optimization process, it outperforms MOA with other parameter setting strategies in most large-scale optimization problems. We also study the algorithm while employing the OBL by comparing it with the original version of MOA. Furthermore, the proposed algorithm is tested and compared with seven traditional population-based algorithms and eight state-of-the-art optimization algorithms. The comparisons demonstrate that the proposed algorithm outperforms the traditional algorithms in most benchmark problems, and its results is comparative to those obtained by the state-of-the-art algorithms.

Keywords: Parameter Adaptation Strategy, Opposition-Based Learning, Magnetic Optimization Algorithm, Numerical Optimization Problems.

1. Introduction

Inspired by the principles of attraction among magnetic particles, MOA is a population based algorithm that belongs to the group of swarm intelligence algorithms. In MOA, the candidate solutions are some magnetic particles that are scattered across the search space. In this respect, each magnetic particle has a measure of mass and magnetic field according to its fitness. In this scheme, the fitter magnetic particles have higher magnetic field and greater mass. In terms of interaction, these particles are located in a lattice-like population and apply a long range force of attraction to their neighbours. Unlike Particle Swarm Optimization (PSO) algorithm in which each particle utilizes only the best experience of the best neighboring particle(s) or the best particle in the population, in MOA each magnetic particle uses the best experience

*Corresponding author. AmirKabir University of Technology, Tehran, Vali Asr St., ,Iran. Phone: +989156523782

ACCEPTED MANUSCRIPT

of all its neighboring particles, including the inferior ones. In order to improve the performance of the algorithm, an OBL [1] approach is proposed in this paper in which by calculating the opposite population of the current population at each iteration, the algorithm tries to find fitter solutions. OBL has been used to solve many optimization problems [2, 3, 4] and has been employed in several population based algorithms [5, 6, 7, 8, 9].

MOA has shown promising results when applied to numerical benchmark functions [10, 11] and to a wide range of optimization problem including travelling salesman problem [12] and multi-layer perception training [13]. Similar to other population based algorithms, the performance of the MOA depends on appropriately setting its parameters [10, 11]. Although there is a systematic way of setting the parameters of MOA [10, 11], it is computationally expensive. The parameter setting technique [14, 10, 15, 16] provides appropriate values for control parameters; however, the algorithm designer needs to set the control parameters for each problem prior to the search process.

To improve the performance of the algorithm in this aspect, several parameter setting approaches have recently been proposed. The F-Race algorithm firstly proposed for tackling the model selection problem [17] is among them. The algorithm is an automatic parameter configuration algorithm that was firstly used by [18] to automatically set the parameters of Ant Colony algorithm. Then a new version of the algorithm called Iterated F-Race was utilized in some optimization algorithms [19, 20, 21]. Iterated F-Race determines the most appropriate parameter configuration of an algorithm using the non-parametric Friedman's two-way analysis of variance by ranks. Acting like a hill climbing stochastic procedure, iterated F-Race performs a few race among the candidate configurations on a stream of instances in order to find the best candidate configuration. First a set of configurations with uniform random values are initialized. Then, at each iteration, all configurations are evaluated according to Friedman test. If the first Friedman test shows that at least one configuration is significantly different from any other configurations in the race, the second Friedman test is applied to eliminate the candidates that are remarkably worse than other configurations. The race proceeds with the surviving configurations and continues until only one candidate configuration remains in the race or the a certain number of iteration is reached. Although the method is successful in setting parameters of algorithms, specially when an algorithm has a number of parameters [19, 21], it can be prohibitively expensive for large-scale optimization problems.

Using the feedback received from the search process, parameter adaptation techniques adjust the parameters of algorithms adaptively. According to [22, 23, 24], depending on how the received feedback is used, there are three major types of parameter setting strategies: deterministic parameter control, self-adaptive parameter control and adaptive parameter control. The deterministic parameter setting approaches are those that do not receive any feedback from the optimization process and set their parameters prior to the search process via trial and error. The original version of MOA is an example of this type of strategy. Self-adaptive parameter setting strategy attempts to evolutionarily adjust the parameters of algorithms; to do so, they often adopt recombination operators such as mutation and crossover to select the optimal parameter configuration. This approach has shown remarkable success in iteratively making the individuals more adapted to the problems. For example, reference [25] proposed a new Differential Evolution (DE) algorithm that uses a self-adaptive parameter strategy for the population size, mutation rate and crossover rate. The parameter adaptation strategy refers to the parameter setting, which uses feedback received from the search process to dynamically set the parameters of the problem. Several state-of-the-art algorithms such as JADE [22], SaDE [26], jDE [27] and Memetic algorithm with adaptive local search [28] can be categorized into this group. The proposed algorithm, which dynamically adjusts its control parameters in the course of the optimization, also belongs to this category.

Being adaptable to the properties of the problem usually enhances the ability of algorithms to find good parameters without spending time on the trial and error parameter setting procedure. Therefore, parameter adaptation strategy can help the algorithm discover a good parameter value while enhancing the convergence performance. JADE as one of the powerful DE algorithms that employs the parameter adaptation strategy showed remarkable success in tackling several small-scale optimization problems [22]. JADE has two control parameters that sets them adaptively. In this paper, we develop the idea used in JADE for adaptively setting the control parameters of the proposed algorithm. The difference between the proposed algorithm and JADE is that our algorithm optimizes the control parameters individually. When the control parameters

Accepted manuscript

ACCEPTED MANUSCRIPT

are investigated and set together (similar to JADE), they may not provide high-quality results for large-scale benchmark problems. This is because it cannot be ensured which parameter is responsible for improving the quality of the solution and so unnecessary changes in the value of a parameter may occur. Instead, if separately evaluated and set, the parameters can be more appropriately adjusted which results in better performance.

The contribution of this paper is summarized into the following aspects:

- A new version of MOA using a run-time adaptation strategy for dynamically setting the parameters of the algorithm is proposed.
- A new approach that is based on the opposite number principle is developed and added to the algorithm to improve its performance.
- The proposed parameter adaptation strategy is compared with two famous parameter setting techniques including systematic parameter setting and F-Race algorithm.
- A set of most powerful optimizers including Genetic Algorithm (GA) [29], PSO [30], DE [31], Evolution Strategy (ES) [32], Fast Evolution Strategy (FES) [33], Evolutionary programming (EP) [34] and Fast Evolutionary Programming (FEP)[35], Memetic Algorithm with Solis Wet local search (MASW) [36], Memetic Algorithm with Subgrouping Solis Wet local search (MASSW) [36], Cooperatively Coevolving Particle Swarms Optimization (CCPSO2) [37], JADE [22], Three Stages Memetic Exploration (3SOME) [38], Parallel Memetic Structure (PMS) [39], Biogeography Based Optimization (BBO) [40], Opposition-based Differential Evolution (ODE) [41] and Covariance Matrix Adaptation Evolution Strategy (CMAES) [42] are used to be compared with the proposed algorithm on 27 standard benchmark functions.

The rest of this paper is organized as follows. Section 2 discusses the background of the proposed algorithm, including the OBL and the original version of MOA. Section 3 introduces the proposed algorithm. Section 4 evaluates the proposed parameter adaptation strategy, by studying the control parameters and comparing the proposed strategy with two well-known strategies. Section 5 provides a comparison between the proposed algorithm and the original version of MOA, seven popular population-based and nine state-of-the-art algorithms. Section 6 concludes this paper.

2. Background

In this section, a general overview of the key components of the proposed algorithm is presented, concentrating on the opposition-based learning scheme and the original version of MOA.

2.1. Opposition-Based Learning

Population-based algorithms often initialize the population randomly, thus the chance of sampling better regions in the search space is not higher. However, there are several ways to enhance the probability of detecting better regions. One is Opposition-Based Learning (OBL). By employing OBL at the initialization phase of algorithms, the likelihood of finding better solutions increases. Furthermore, an algorithm can employ the OBL approach during its search process to increases its chances of finding better solutions [41].

The concept of OBL was proposed by Tizhoosh in[1]. In this paper, we first explain the concept of opposition numbers. Let $x \in [a, b]$ be a real number, then the opposite number $\breve{x}$ is defined as,

$$\breve{x} = a + b - x.$$

The definition can be extended to an N-dimensional search space[1] as follows. Let $P = (x_1, x_2, ..., x_N)$ represent a point in an N-dimensional space. The opposition vector in this space is defined as,

$$\breve{x}_i = a_i + b_i - x_i.$$

Accepted manuscript

ACCEPTED MANUSCRIPT

Heretofore, the OBL has extensively been used to solve many optimization problems[2, 3, 4] and has been employed in several population-based algorithms[41, 6, 7, 9]. This encouraged us to employ the method in MOA to speed up the convergence speed and maintain population diversity simultaneously, thus reaching better solutions more swiftly.

2.2. Magnetic Optimization Algorithm

Inspired by the principles of magnetic field theory, MOA [10] was proposed to cover some weaknesses of the PSO, including premature convergence and the dictatorship of the best particles. In the traditional version of PSO algorithm, individuals tend to follow and imitate the best particle, which results in premature convergence. In newer versions of PSO like cellular PSO, in which inferior particles follow the behaviour of the best neighbouring particles, individuals usually suffer from the dictatorship of the best neighbouring particles. The dictatorship in this context means that the best neighbouring individuals always force other particles to abandon their positions and follow their lead, resulting in ignoring some important information that might be lied within low fitness particles. To overcome this weakness, MOA was proposed [10] which uses a cellular structure to surmount the premature convergence. It also utilizes a motion strategy, where each particle, even the lowest fitness one, influences other neighbouring particles.

Apart from these advantages, one disadvantage of MOA, similar to many other optimization algorithms, is that it has some parameters that should be carefully tuned before solving a problem. The original version of MOA [10, 11] has two parameters (α and ρ). Another problem is that its parameters are problem dependent, so each parameter of the algorithm needs to be set for every specific problem. The pseudo-code of MOA is presented in 2.2.

Procedure Basic MOA
begin
  $t = 0$
1. initialize X^0 with a structured population
2. while not termination condition do
  begin
    $t = t + 1$
3. evaluate the particles in X^t and store their performance
      in magnetic fields B^t
4. normalize B^t according to equation 6
5. evaluate the mass M^t for all particles according to (7)
6. for all particles x_{ij}^t in X^t do
    begin
7. F_{ij}=0
8.   find N_{ij}
9.   for all x_{uv}^t in N_{ij} do
10. $F_{ij} = F_{ij} + \frac{(x_{uv}^t - x_{ij}^t) \times B_{uv}^t}{D(x_{ij}^t, x_{uv}^t)}$
    end
11. for all particles x_{ij}^t in X^t do
    begin
12. $v_{ij,k}^{t+1} = \frac{F_{ij,k}}{M_{ij,k}} \times R(l_k, u_k)$
13. $x_{ij,k}^{t+1} = x_{ij,k}^t + v_{ij,k}^{t+1}$
    end
  end
end

A more comprehensive description of each step of the original MOA can be found in [11].

Accepted manuscript

ACCEPTED MANUSCRIPT

3. The proposed algorithm

The proposed algorithm consists of two major elements: MOA and OBL. It also employs an extended version of JADE's parameter adaptation strategy to dynamically adjust the algorithm's parameters. The proposed adaptation strategy differs from JADE's in one important aspect. Algorithms usually have some parameters that affect one another. In JADE, this effect is ignored and the parameters are optimized independently. Our idea here is to take this effect into account in the parameter adaptation process.

In addition to the proposed parameter adaptation strategy, the proposed algorithm benefits from a population-diversifying procedure called the OBL procedure. The OBL has been employed to improve the performance of some population-based algorithms [5, 6, 7, 8, 9].

In the proposed algorithm the magnetic particles interact with each other in a cellular-like structure as represented in figure 1.

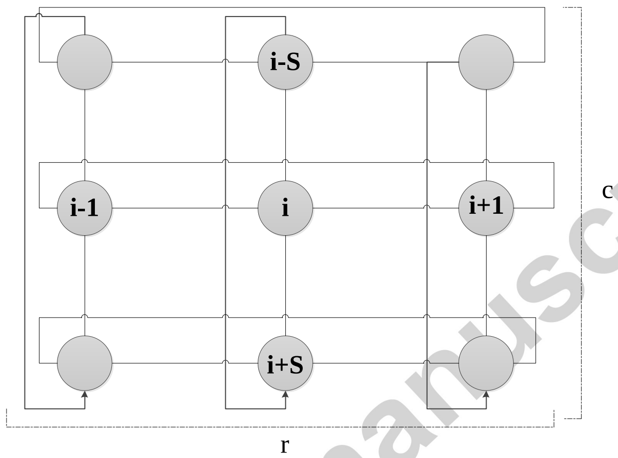


Figure 1: The interaction scheme between the i-th particle in the population and its neighboring particles in the cellular-like structure with size of (r, c).

Accepted manuscript

The pseudo-code of the proposed algorithm is represented in algorithm 3.

Procedure FMOA

1 $t = 0$, $J = 0.3$, $\rho = 0.5$, $\alpha = 0.5$
2. x^t = Initialization()
3. **while** not termination condition do
    $t = t + 1$
4. $J_A = \oslash$, $\alpha_A = \oslash$, $\rho_A = \oslash$
5. **for** i= 1 to N_p
6. J_i = randnorm(J_r, 0.01)
7. B_i^t = evaluateFitness(x_i^t)
8. **if** $B_i^t \geq B_i^{t-1}$
9. $\rho_i \rightarrow \rho_A, \alpha_i \rightarrow \alpha_A$
    end if
10. **if** $\mathrm{R}(0, 1) \leq J_i$
11. $Y_{i,k}^t$=Min_k + Max_k-$x_{i,k}^t$
12. **if** $B_i^t \leq f(Y_i^t)$
13. $x_i^t = y_i^t$
14. $J_i \rightarrow J_A$
    end if
    end if

ACCEPTED MANUSCRIPT

15. ρ_i = randnorm(ρ, 0.1), α_i = randnorm(α, 0.1)
end for
16. normalize B^t using equation (6)
17. evaluate the mass M^t for all particles according to (7)
18. **for** i= 1 to N_p do
19. F_i=0
20. N = findneighbor(i)
21. **for** j= 1 to S do
23. $F_i = F_i + \frac{(x^t_{N_j} - x^t_i) \times B^t_{N_j}}{D(x^t_i, x^t_{N_j})}$
end for
end for
24. **for** i= 1 to N_p do
25. $v^{t+1}_{i,k} = \frac{F_{i,k}}{M_{i,k}} \times R(l_k, u_k)$
26. $x^{t+1}_{i,k} = x^t_{i,k} + v^{t+1}_{i,k}$
end for
27. $\rho = (c-1) \times \rho + c \times (mean_L(\rho_A))$
28. $\alpha = (c-1) \times \alpha + c \times (mean_L(\alpha_A))$
29. $J_r = (c-1) \times J_r + c \times (mean_S(J_A))$
end while
end procedure

A more detailed description of the proposed algorithm is as follows.

1. In this step, the parameters of the algorithm α, ρ and J_r are initialized. J_r is initialized with 0.3 because it was shown in [41] that the best value for this parameter is in $[0.3 - 0.6]$.

2. Randomly initialize the particles in the population x^t. The initialization process is performed as follows:

$$x_{i,k} = R(l_k, u_k), \tag{1}$$

for $i = 1, 2, ..., N_p$ and $k = 1, 2, ..., D$ where N_p and D are the size of the population and problem respectively, l_k and u_k are the lower and the upper bounds of the search space and $R(.,.)$ is a uniform random number generator.

3. The "termination condition" is met when the maximum number of iterations (M_I) is reached.

4. In this step, the set of successful jumping rate values J_A and the set of successful parameters (α_A, ρ_A) are initialized as empty sets.

5. Steps 6-15 are applied to all the particles.

6. The jumping rate for i-th magnetic particle in the population is generated according to a normal distribution with the mean of J_r and standard division 0.01. This procedure is carried out as,

$$J_i = randnorm(J_r, 0.01). \tag{2}$$

For standard division, we use 0.01 as our studies showed that it is the best choice.

7. In this step, the objective of x_i is calculated and stored in the magnetic field B_i.

8-9. If the fitness of i-th particle has improved in the current generation, α_i and ρ_i are inserted in α_A and ρ_A respectively.

10. The OBL procedure is performed at i-th particle with the probability of jumping rate.

11. In this step, the opposite points of the current particle x_i are found and stored in Y^t_i. This is performed as,

$$Y^t_{i,k} = Min^t_k + Max^t_k - x^t_{i,k}, \quad k = 1, 2, ..., D, \tag{3}$$

Accepted manuscript

ACCEPTED MANUSCRIPT

where Max_k^t and Min_k^t are the maximum and minimum values of the variables in the k-th dimension at iteration t respectively.

12-14. If the particle x_i^t is worse than its opposite particle Y_i^t, it is replaced by its opposite, and J_i is stored in J_A.

15. In this step, α_i is generated from normal distribution with the mean parameter α and standard deviation parameter 0.1.

$$\alpha_i = randnorm(\alpha, 0.1), \tag{4}$$

Similarly, the value of ρ_i is generated from normal distribution with the mean parameter ρ and standard division parameter 0.1.

$$\rho_i = randnorm(\rho, 0.1) \tag{5}$$

For standard division, we use 0.1 as our studies demonstrated that it is the best choice.

16. To remove problem dependency, the magnetic field of each particle is normalized within the range of $[0-1]$. B_i^t is normalized as follows,

$$\frac{B_i^t - Min(B)}{Max(B) - Min(B)}, \tag{6}$$

where "Min" and "Max" are the minumum and maximum magnetic fields among all the population members.

17. The mass of all the particles is found and stored in M^t. The mass of each particle is found as,

$$M_i^t = \alpha_i + \rho_i \times B_i^t. \tag{7}$$

18. The "for" loop is applied to all the particles.

19. The magnetic force of the particle x_i^t is set to zero.

20. All the neighbors of the particle x_i^t are found (See figure 1).

21. The magnetic force applied to the particle x_i^t from its neighbors is found as,

$$F_{i,k} = \frac{(x_{u,k}^t - x_{i,k}^t \times B_u^t)}{D(x_{u,k}^t, x_{i,k}^t)}, \tag{8}$$

where $D(.,.)$ represents the distance between two neighboring particles and is calculated as,

$$Dis(x_{u,k}^t, x_{i,k}^t) = \frac{1}{n}\sum_{k=1}^{n} \left|\frac{x_{u,k}^t - x_{i,k}^t}{u_k - l_k}\right|, \tag{9}$$

and x_u is the u-th neighbor of the particle x_i.

Since the distance between two particles depends on the domain of the search space, it is normalized (see [10]).

24. In the "for" loop the location of the particles is updated.

25. The location and the velocity of the particle $x_{i,k}^{t+1}$ are updated as,

$$v_{i,k}^{t+1} = \frac{F_{i,k}}{M_i} \times R(l_k, u_k), \tag{10}$$

$$x_{i,k}^{t+1} = x_{i,k}^t + v_{i,k}^{t+1}, \tag{11}$$

where $F_{i,k}$ is the force applied to the i-th particle, $v_{i,k}^{t+1}$ and M_i are the velocity and the mass of i-th particle respectively and $\frac{F_{i,k}}{M_{i,k}}$ determines the magnitude and the direction of the particle $x_{i,k}^{t+1}$.

Accepted manuscript

ACCEPTED MANUSCRIPT

27. At the end of each iteration, ρ is updated through linear summation of the mean of α_A and the current value of ρ as,

$$\rho = (c-1) \times \rho + c \times (mean_L(\rho_A)), \tag{12}$$

where c is a constant positive value between 0 and 1 that makes a trade-off between the past experience of ρ and the successful experiences of ρ in the current iteration and $mean_L(.)$ is the Lehmer mean that is calculated as,

$$mean_L(\rho_A) = \frac{\sum_{k\in\rho_A} k^2}{\sum_{k\in\rho_A} k}. \tag{13}$$

In the proposed update strategy, the value of ρ is determined by two parts of the equation. In the equation, the parameter c controls the emphasis on each part. The first part is the current ρ that comes from the current value of the parameter. The current value of the parameter has been found in previous iterations so represents the past experience of the algorithm. The second part is $mean(\rho_a)$ which is the average of the best values of the parameter in the current iteration. Therefore, the second part represents the best value of the parameter in the current iteration.

28. Similarly α is updated as,

$$\alpha = (c-1) \times \alpha + c \times (mean_L(\alpha_A)). \tag{14}$$

29. The jumping rate value Jr is updated as,

$$J_r = (c-1) \times J_r + c \times (mean_S(\alpha_A)), \tag{15}$$

where $mean_S(.)$ is the standard arithmetic mean.

For updating α and ρ we use the Lehmer mean and for J_r the arithmetic mean, as our studies showed that these are better choices. The difference between Lehmer and arithmetic mean is that Lehmer propagates larger α and ρ values while arithmetic propagates smaller values.

4. Discussion of Parameter Setting

In this section, we first study the proposed parameter setting technique, by finding the relation between the problem size and the parameters. We then study the performance of the proposed parameter adaptation technique by comparing it with two well-known parameter setting techniques.

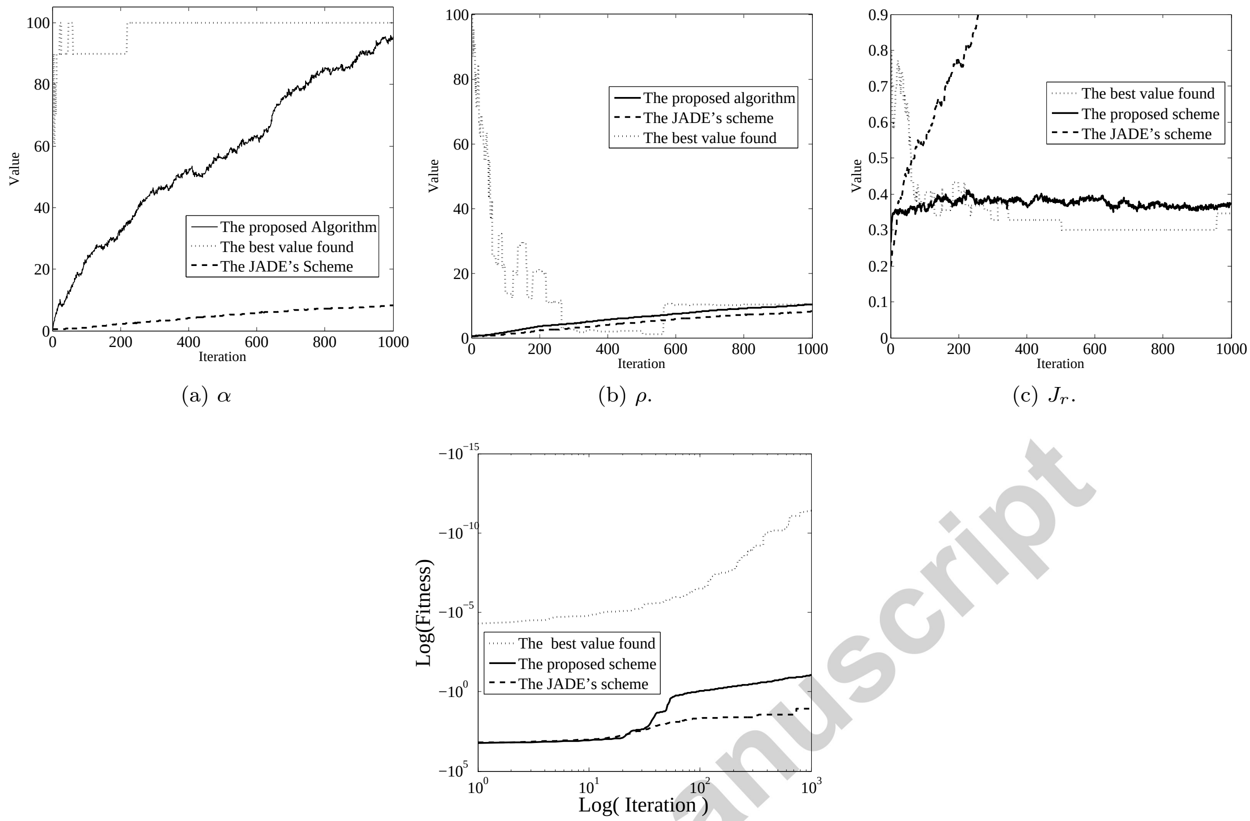
4.1. Parameter Study

The proposed algorithm uses the same strategy for storing and updating parameter values as JADE [22]. However, the major difference here is that instead of updating all the parameters in one process, our proposed scheme updates each parameter independently. The advantage of this method is that the parameters do not intervene and each parameter has the opportunity to move towards its desired optimal value. In JADE however, since the algorithm evaluates the performance of all the parameters in one process and measures the combined performance, the parameters intervene and some parameters distract the others.

In this section we analyse the performance of the proposed algorithm and that of JADE to show the advantage of our algorithm. To do so, we find the best value for each of the parameters at each iterations. We find this best value by giving different values to each parameter at each iteration and measuring the performance of the algorithm when each of the values are used. Then we pick the value that offers the best performance and move to next iteration. The set of values for the parameters are $\alpha_b = best(0.001, 0.01, 0.2, 0.6, 1, 2, 5, 10, 30, 50, 100)$, $rho_b = (0.001, 0.01, 0.2, 0.6, 1, 2, 5, 10, 30, 50, 100)$ and $J_r b = (0.3, 0.45, 0.6, 0.8, 1)$. Doing so, we can compare the best values for the parameters at each iteration and the best parameters offered by the algorithms. Figure 2 shows the result of the analysis on the parameters of FMOA where the proposed scheme is compared with JADE's scheme and the parameter value.

Accepted manuscript

ACCEPTED MANUSCRIPT



(a) α (b) ρ. (c) J_r.

(d) The fitness value obtained.

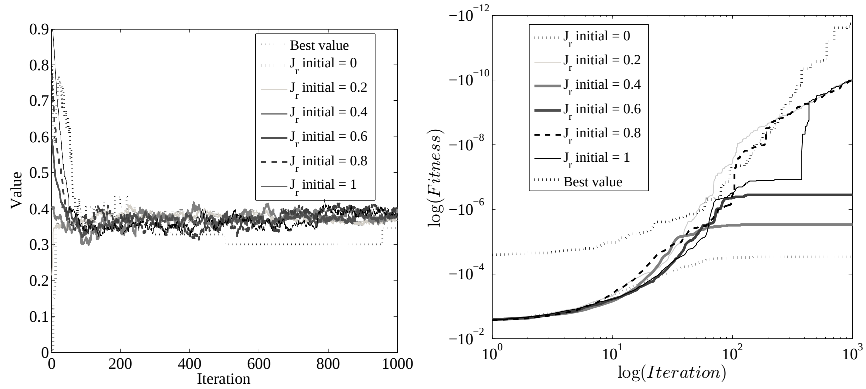
Figure 2: The best parameters at each iteration found by the proposed algorithm and JADE.

Figure 2-a, shows the best value for the parameter α found with different methods. The dotted line shows the best parameter at each iteration. As shown in this figure, the parameter offered by the proposed algorithm moves from the starting point to the best value, which means that it has been able to detect the position of the best value. In this figure, JADE has also been able to detect the position of the best parameter, but its movement is much slower than that of the proposed algorithm. We believe this could be attributed to the fact that JADE evaluates the combined performance of all the algorithms at the same time and therefore is less capable of measuring the true value of the best parameter. This results in a slow movement toward the best value. The best parameters offered by each of the algorithms for the parameter ρ is shown in figure 2-b. For this parameter, both the algorithms behave similarly and reach the best parameter rapidly.

The interesting behaviour is seen for the parameter J_r in figure 2-c, where the proposed algorithm rapidly reaches and follows the best parameter, while JADE does not even find the best parameter and moves away from it. The parameter j_r controls the opposition movement which has the role of exploring the search space. A greater j_r means an explorative and a smaller j_r means a more exploitative algorithm. As seen in figure 2-a and b, the values that JADE offers for α and ρ are smaller that the best value for the parameters. Smaller α and ρ means that the particles have smaller mass than they should have. Particles with smaller mass are more affected by other particles and move rapidly towards better particles. This means that small α and ρ result in rapid convergence around local optima. We believe the reason that JADE offers a much greater j_r is to make the algorithm more explorative to help it escape from local optima. In other words, a

Accepted manuscript

ACCEPTED MANUSCRIPT



(a) The parameter value versus the number of iteration taken by the algorithm when different initial values of J_r are used.

(b) The fitness values achieved by different initial values of J_r

Figure 3: The analysis on the proposed method when different initial values are used for J_r.

large value for j_r to some extent cancels the small values offered for α and ρ.

Figure 2-d represents the fitness of the best found solution versus the iteration of the algorithm on a log-log scale. Obviously, the mechanism that uses the best parameter archives the best performance. This is because it exhaustively checks all the values for the parameter and chooses the best one. This may offer the best performance, but it is not practical as it is extremely time consuming (Finding the best parameter at each iteration takes $N \times \delta$ instructions, where N is the number of parameters and δ is the number of instructions it takes to run the algorithm for one iteration). After that is the proposed algorithm which has reached a reasonably good performance compared to JADE.

As shown in the presented analysis, the proposed algorithm can adapt the parameters of the algorithm and move them toward the value that offers the best performance. This, in some sense, makes the algorithm parameterless, as the parameters adapt themselves and there is no need for setting the parameters before using the algorithm. These parameters, however, should start from an initial value where the proposed parameter adaptation algorithm moves them towards the best value. The question here is to what extent this initial value affects the movement of the parameters. Figure 3 shows the parameter J_r when the proposed parameter adaptation algorithm is used. The graph shows the parameter at each iteration for different initial values of $J_r = (0, 0.2, 0.4, 0.6, 0.8, 1)$. As shown in this figure, regardless of the initial value of the parameter, it always converges to the best value of the parameter. This clearly suggests that the proposed algorithm is not sensitive to the initial value of the parameters, and so the algorithm is parameterless.

4.2. Comparison with two famous parameter setting techniques

In order to study the performance of the proposed adaptive parameter approach, we compare it with two famous parameter setting techniques, the systematic parameter setting [14, 10] and the automatic algorithm configuration (iterated F-Race) [19]. We call the proposed algorithm with iterated F-Race FRFMOA and the proposed algorithm with systematic approach OMOA. First we provide a short overview of these techniques.

1. Systematic parameter setting
   (a) **Definition:** By discritizing the parameter space and evaluating its all possible values, this method systematically finds the best value for each parameter of the algorithm. In this strategy, first a large range is given to the algorithm making sure that the best parameter value is within this range. Thus the best results versus the parameter forms a "U" shape (in a minimization problem). Then knowing where the bottom of the "U" is located, we find a rough idea about the location of

Accepted manuscript

ACCEPTED MANUSCRIPT

Table 1: Best parameter for OMOA on 21 benchmark functions. The results are averaged over 50 runs.

Algorithm	Parameter	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
OMOA	α	0.4	0.4	1	0.2	0.6	1	0.6	0.4	0.01
	ρ	1	1	0.6	1	1	0.2	0.2	0.2	0.8
	J_r	0.3	0.3	0.3	0.45	0.3	0.6	0.6	0.6	0.6
Algorithm	Parameter	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
OMOA	α	0.4	0.4	0.001	0.2	1	0.8	0.2	0.01	0.6
	ρ	1	1	0.001	0.2	0.6	1	0.2	0.001	0.4
	J_r	0.3	0.45	0.6	0.3	0.3	0.3	0.45	0.6	0.45
Algorithm	Parameter	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}
OMOA	α	0.2	1	0.6	1	0.6	0.2	0.001	0.001	0.2
	ρ	1	1	1	0.2	0.001	0.01	0.2	0.2	0.01
	J_r	0.3	0.3	0.3	0.3	0.45	0.6	0.6	0.6	0.6

the best parameters. We then shrink the domain from both sides focusing on the best parameters to get a better resolution picture of the best parameters. For more information about this strategy the readers are referred to [10].

(b) **Parameter setting with systematic approach:** Table 1 represents the best parameters found over 50 runs for OMOA on 27 benchmark problems.

2. F-Race:

(a) **Definition:**
Instead of discretizing and checking all the possible values of the parameters, this strategy finds the best parameters for the algorithm by performing a search process on the parameter space. First several configurations of parameters are randomly generated. Then the configurations race against one another over a set of problem instances. At each iteration of the F-race algorithm, a set of best parameter configurations is selected for generating new candidate configurations for the next iterations. The race is finished and the best candidate configuration is reported when the tuning time budget is exhausted or when only one candidate configuration remains in the race.
The main advantage of F-Race over checking all the possible values of the parameters is when the algorithm has a large number of parameters. For such problems, it is usually very expensive or sometimes impossible to check all the possible values. The other advantage is when the algorithm is to be used for a new problem. In such a case, it is very hard to know the best parameters for the new problem unless an expensive parameter setting is performed. Since F-Race performs the parameter search on a number of problem instances and finds the best parameters that work the best on these problem, it has the ability to generalize the best parameters for a set of problems. Despite all these benefits, this algorithm still has some weaknesses. First it only can generalize the parameters according to the problem instances on which the parameter setting is performed. So the parameters may not be the best for a completely new problem that shows different properties from the previous problems. Second it finds a set of parameters that, on average, work the best for a number of problems. So this set of parameters is not necessarily the best for each of these problem individually. And finally when the parameters are set, they are constant throughout the search process. Obviously, it is better to have variable parameters that adapt during the search process.

(b) **Parameter setting with F-Race:** Table 2 shows the parameter setting via F-Race. Note that the suggested values are obtained from all 21 benchmark problems; so they will be used for all the problems.

Table 2: The best parameter configuration for OMOA on 21 benchmark functions. The results are averaged over 50 runs.

Parameter	Value
α	86.6102
ρ	93.01561
J_r	0.7958435

Comparing with F-Race and Systematic approach: The proposed adaptive parameter

Accepted manuscript

ACCEPTED MANUSCRIPT

Table 3: The experimental results for the FMOA, FRFMOA and OMOA for 9 unimodal benchmark problems. The best results are bolded.

		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
D=100										
FMOA	Mean	-1.67e-1	-3.60e-1	-1.04e-1	-9.80e+1	-1.62e-9	-6.24e-7	-5.13e-3	-1.48e+3	-1.02e+6
	Std	8.00e-2	6.38e-2	2.06e-2	3.21e-3	9.59e-10	2.16e-6	7.89e-4	3.44e+2	5.81e+6
FRFMOA	Mean	-1.57e+5	-8.43e+29	-7.14e+1	-1.21e+4	-1.56e-3	-8.44e-4	-8.20e-3	-2.79e+3	-7.57e+8
	Std	7.84e+3	2.33e+30	1.69e+0	8.60e+2	8.24e-5	5.11e-4	1.38e-3	5.14e+2	2.97e+8
OMOA	Mean	**-1.41e-1**	**-3.18e-1**	**-8.10e-2**	**-9.68e+1**	**-1.54e-9**	**-6.67e-9**	**-4.24e-3**	**-1.45e+3**	**-5.39e+3**
	Std	**6.59e-2**	**1.00e-1**	**2.55e-2**	**7.23e+0**	**8.12e-10**	**5.40e-9**	**6.30e-4**	**1.29e+2**	**2.64e+3**
D=500										
		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
FMOA	Mean	**-2.19e+0**	**-3.92e-1**	-2.53e-2	-6.31e+2	**-1.50e-6**	**-2.21e-7**	**-2.06e-2**	-1.58e+3	**-2.51e+8**
	Std	**1.52e+1**	**6.94e-2**	4.25e-3	3.54e+2	**1.06e-5**	**6.41e-7**	**1.46e-3**	4.13e+2	**9.69e+8**
FRFMOA	Mean	-1.40e+6	-2.37e+262	-9.42e+1	-1.67e+5	-1.40e-2	-1.57e-4	-3.64e-2	-3.39e+3	-6.62e+11
	Std	2.49e+4	Inf	3.39e-1	4.03e+3	2.60e-4	2.16e-4	2.35e-3	7.75e+2	1.60e+11
OMOA	Mean	-4.23e+2	-2.46e+262	**-1.98e-2**	**-5.20e+2**	-1.08e-5	-2.71e-7	-2.09e-2	**-1.07e+3**	-3.77e+9
	Std	2.13e+3	Inf	**7.47e-3**	**1.67e+2**	3.86e-5	1.60e-6	2.46e-3	**7.61e+1**	2.14e+10
D=1000										
		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
FMOA	Mean	-1.81e+3	-	-1.36e-2	**-1.03e+3**	-3.93e-5	**-5.16e-7**	**-4.44e-2**	-2.64e+3	**-1.85e+10**
	Std	1.02e+4	-	2.63e-3	**2.69e+2**	1.39e-4	**1.60e-6**	**2.79e-3**	6.23e+2	**7.89e+10**
FRFMOA	Mean	-2.99e+6	-	-9.71e+1	-3.79e+5	-3.00e-2	-1.65e-4	-7.14e-2	-3.82e+3	-1.24e+13
	Std	4.25e+4	-	1.82e-1	7.03e+3	2.92e-4	3.71e-4	3.74e-3	8.81e+2	4.36e+12
OMOA	Mean	**-1.71e+3**	-	**-1.11e-2**	-1.16e+3	**-1.79e-10**	-4.22e-6	-4.44e-2	**-9.60e+2**	-2.41e+11
	Std	**8.70e+3**	-	**4.94e-3**	5.54e+2	**1.12e-10**	1.92e-5	2.76e-3	**1.62e+2**	1.24e+12

setting is compared with F-Race and systematic approach on 21 benchmark functions over 3 different problem size, $D = 100$, 500 and 1000.

Tables 3 and 4 show the results obtained by FMOA, FRFMOA and OMOA on 21 benchmark functions, where "Mean" and "Std" are the mean and standard deviation of fitness values obtained by each algorithm over 50 runs.

As shown in Table 3, OMOA offers the best results when the dimensionality is equal to 100-D. However, as the dimensionality increases, FMOA performs the best on f_1, f_2, f_5, f_6, f_7 and f_9 for $D = 500$ and on f_4, f_6, f_7 and f_9 for $D = 1000$. As shown in Table 4, MOA and FMOA perform the best on some problems and both show high-quality results for most problems. However, the main advantage of FMOA over OMOA is that there is no need to set its parameters before running it on a problem, because as mentioned earlier, its parameters are adaptively adjusted during the optimization.

Table 6 shows the Friedman test among FMOA, FRFMOA and OMOA for different benchmark problems of different size. As shown in this table, in unimodal problems for size of the problems up to 500, FMOA achieves the best results. Then, as the problem size increases, OMOA reaches FMOA. In multimodal problems, the results are different and as the problem size grows, FMOA reaches better results. Interestingly, for all the problems, the friedman test shows that FMOA reaches similar results to those of OMOA. This indicates that FMOA offers good performance, despite the fact that the expensive parameter setting process is removed.

5. Experimental Results

In this section we first give a short description about the numerical functions used for comparison. Then, using a RLDs measurement technique, we investigate the effect of OBL on the performance of the proposed algorithm. Next, we compare FMOA with seven well-known population-based algorithms including GA [29], PSO [30], DE [31], ES [32], FES [33], EP [34] and FEP [35] on the benchmark functions. Finally, we compare FMOA with nine state-of-the-art optimization algorithms including MASW [36], MASSW [36], CCPSO2 [37], JADE [22], 3SOME [38], PMS [39], BBO [40], ODE [41] and CMAES [42].

Accepted manuscript

ACCEPTED MANUSCRIPT

Table 4: The experimental results for FMOA, FRFMOA and OMOA for 18 multimodal benchmark functions. The best results are bolded.

		f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
D=100										
FMOA	Mean	6.17e+3	-8.74e-2	**-6.58e-2**	1.15e+0	**1.61e+2**	**-1.05e-1**	**1.57e+1**	9.88e+1	3.05e+0
	Std	2.42e+3	4.93e-2	**1.34e-2**	7.98e-2	**5.10e+1**	**3.12e-3**	**2.47e+0**	5.27e+0	3.02e-1
FRFMOA	Mean	5.54e+3	-1.32e+3	-2.01e+1	-1.40e+3	-8.39e+4	-1.28e+5	6.99e+0	4.95e+1	-2.24e+6
	Std	7.57e+2	4.75e+1	8.12e-2	6.33e+1	3.78e+3	3.98e+3	5.20e-1	1.20e+0	2.07e+5
OMOA	Mean	**2.36e+4**	**-8.00e-2**	-8.03e-2	**1.22e+0**	-2.95e-2	-1.06e-1	1.31e+1	**9.99e+1**	**4.20e+1**
	Std	**8.53e+3**	**4.19e-2**	2.08e-2	**1.48e-1**	8.37e-4	3.80e-3	8.36e-1	**2.02e-2**	**3.06e+0**
D=500										
FMOA	Mean	1.41e+4	**-1.84e-2**	-2.52e-1	-1.16e+1	**3.35e+1**	-1.91e+2	1.70e+1	**4.47e+2**	**3.32e+0**
	Std	3.27e+3	**8.74e-3**	1.19e+0	4.95e+1	**1.92e+1**	1.34e+3	1.81e+1	**8.81e+1**	**3.09e-1**
FRFMOA	Mean	1.15e+4	-8.58e+3	-2.10e+1	-1.26e+4	-6.89e+5	-9.50e+5	9.93e+0	2.09e+2	-1.79e+8
	Std	1.29e+3	1.00e+2	1.80e-2	2.11e+2	1.03e+4	1.38e+4	6.05e-1	2.28e+0	4.91e+6
OMOA	Mean	**1.89e+5**	-2.35e-2	**-1.13e-2**	**9.70e-1**	2.81e+1	**-4.98e+1**	**2.60e+1**	4.29e+2	-7.10e+3
	Std	**3.32e+4**	2.11e-2	**3.47e-3**	**5.19e-2**	5.63e+0	**3.18e+2**	**1.29e+0**	9.83e+1	3.52e+4
D=1000										
FMOA	Mean	3.26e+4	-6.63e+1	**-9.91e-2**	-5.40e+0	**2.65e+1**	-3.42e+1	2.33e+1	**9.08e+2**	**-4.56e+0**
	Std	4.04e+4	4.69e+2	**6.37e-1**	4.33e+1	**1.33e+1**	1.28e+2	2.63e+1	**1.68e+2**	**5.59e+1**
FRFMOA	Mean	1.54e+4	-1.76e+4	-2.11e+1	-2.71e+4	-1.47e+6	-2.00e+6	1.18e+1	4.05e+2	-8.24e+8
	Std	2.72e+3	1.58e+2	1.67e-2	4.15e+2	1.46e+4	2.21e+4	7.84e-1	3.99e+0	1.83e+7
OMOA	Mean	**3.23e+5**	**-1.17e-1**	-2.22e-1	**9.19e-1**	-1.15e+1	**-1.00e-1**	**2.66e+1**	7.77e+2	-4.78e+4
	Std	**7.71e+4**	**7.20e-1**	1.10e+0	**6.42e-2**	2.62e+2	**3.54e-4**	**2.61e+0**	2.06e+2	1.62e+5

Table 5: The experimental results for FMOA, FRFMOA and OMOA for 18 multimodal benchmark functions. The best results are bolded.

		f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}
D=100										
FMOA	Mean	-1.77e+3	4.39e+1	-2.89e+4	**-1.45e+03**	-4.13e+01	-7.92e+10	**-1.48e+03**	-4.21e+01	-1.77e+11
	Std	9.02e+1	4.92e-2	1.74e+3	**1.05e+03**	2.64e-01	7.94e+09	**3.99e+01**	1.03e-01	7.24e+09
FRFMOA	Mean	-2.36e+3	4.39e+1	-6.76e+4	-7.61e+08	-4.23e+01	-1.85e+11	-1.94e+03	-4.25e+01	-4.71e+11
	Std	1.21e+2	3.24e-2	7.30e+3	2.62e+08	1.29e-01	3.19e+10	5.79e+01	1.24e-01	4.52e+10
OMOA	Mean	**-1.58e+3**	**4.39e+1**	**-2.78e+4**	-9.22e+03	**-4.07e+01**	**-6.93e+10**	-1.56e+03	**-4.15e+01**	**-1.70e+11**
	Std	**5.55e+1**	**2.61e-2**	**2.23e+3**	1.31e+04	**4.85e-01**	**1.18e+10**	4.27e+01	**1.05e-01**	**1.05e+10**
D=500		f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}
FMOA	Mean	-9.59e+3	4.37e+1	-2.31e+5	**-9.33e+04**	-1.25e+02	-4.06e+11	**-8.33e+03**	**-2.11e+02**	-9.30e+11
	Std	2.53e+2	4.27e-2	5.41e+3	**6.92e+04**	2.53e-01	1.42e+10	**1.06e+02**	**3.18e-01**	1.80e+10
FRFMOA	Mean	-1.33e+4	4.38e+1	-7.76e+5	-5.87e+11	-1.28e+02	-1.80e+12	-1.15e+04	-2.15e+02	-3.99e+12
	Std	3.25e+2	2.29e-2	2.38e+4	1.82e+11	1.53e-01	8.10e+10	2.11e+02	1.87e-01	1.35e+11
OMOA	Mean	**-9.36e+3**	**4.38e+1**	**-2.31e+5**	-2.74e+09	**-1.25e+02**	**-4.03e+11**	-8.52e+03	-2.11e+02	**-9.30e+11**
	Std	**2.31e+2**	**2.69e-2**	**6.44e+3**	1.70e+10	**2.19e-01**	**1.31e+10**	8.12e+01	2.27e-01	**1.78e+10**
D=1000		f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}
FMOA	Mean	-1.95e+4	4.37e+1	**-4.87e+5**	**-3.37e+10**	-2.29e+02	**-9.85e+11**	**-1.72e+04**	**-4.23e+02**	-1.95e+12
	Std	3.99e+2	3.88e-2	**8.71e+3**	**2.38e+11**	2.91e-01	**1.69e+10**	**1.71e+02**	**3.61e-01**	2.21e+10
FRFMOA	Mean	-2.70e+4	4.38e+1	-1.72e+6	-1.30e+13	-2.36e+02	-4.25e+12	-2.40e+04	-4.30e+02	-8.83e+12
	Std	4.46e+2	2.69e-2	5.22e+4	5.04e+12	1.60e-01	1.08e+11	1.86e+02	1.75e-01	1.84e+11
OMOA	Mean	**-1.92e+4**	**4.38e+1**	-4.87e+5	-3.34e+11	**-2.29e+02**	-9.89e+11	-1.75e+04	-4.23e+02	**-1.94e+12**
	Std	**3.47e+2**	**2.00e-2**	8.38e+3	1.55e+12	**2.87e-01**	1.91e+10	1.14e+02	1.48e-01	**2.85e+10**

Table 6: The Friedman Test for all the problems where $D = 100, 500$ and 1000.

	Unimodal			Multimodal			Both and multimodal		
Algorithm	FMOA	FRFMOA	OMOA	FMOA	FRFMOA	OMOA	FMOA	FRFMOA	OMOA
$D = 100$	18	9	27	42	18	48	69	27	66
$D = 500$	24	9	21	43	18	47	67	27	68
$D = 1000$	20	8	20	45	18	45	63	27	63

Accepted manuscript

ACCEPTED MANUSCRIPT

5.1. Test problems

A selected set of benchmark problems consisting of 27 global optimization problems that are usually used in the literature [10, 38, 39, 22] is used in this paper to test the proposed algorithm. These problems are listed in Table 7.

In general, with regard to their modality, these problems are categorized into two classes. The first class is the set of unimodal problems with only one global optimum ($f_1 - f_9$) and no local optima in the search space. The second class is the multi-modal problems with a large number of local optima ($f_{10} - f_{27}$) and one or more global optima. Obviously the second class are the harder problems to solve.

5.2. The benefit of the OBL approach

In this section, we study the performance of the proposed algorithm when OBL operator is applied. To do so, we use the qualified Run-Length Distributions (RLDs), which is a statistical tool for evaluating and comparing algorithms [48]. Note that in this section we only study the OBL operator, so the parameters of the algorithm are tuned using the systematic approach. The best parameters for the algorithm are represented in Table 1.

RLDs attempts to answer the question of "how to empirically measure the run-time behavior of an algorithm on a given problem?". RLDs provides a graphical view of the progress of the likelihood of finding a solution for a problem with a certain quality over a large number of runs.

RLDs has a number of elements. The cumulative probability distribution RLDs of an algorithm, is defined as,

$$P_d = P(RT <= l, SQ <= q'), \tag{16}$$

where q' is a predefined solution quality (fitness), P is the probability of finding a solution whose quality, SQ, is better than or equal to the predetermined solution quality q' and the function evaluation spent to find the solution (RT) is less than or equal to l function evaluations.

To empirically obtain the algorithm's RLDs, the following steps are taken.

1- The algorithm is run for a specific number of function evaluations l. It is terminated when either the maximum number of function evaluations l is reached ($RT \geq l$) or a solution is found that is better than or equal to the pre-specific quality q' ($SQ \geq q'$).

2- Step 1 is iterated until the total number of independent runs A is reached.

3- The success rate of the algorithm is measured through the following formula,

$$S_r = \frac{S}{R}, \tag{17}$$

where S is the number of independent runs that the algorithm has successfully reached q'.

In order to provide a better chance of finding a predetermined solution (q'), we set it in a way that both algorithms can find it within l function evaluations.

The total number of runs is set to 100; the number of function evaluations l is set to 10^6 for all the problems where the size of the population for both algorithms is equal to 50. While M_I (the maximum number of iterations) for MOA is constant and is equal to 20000. For MOA with OBL procedure it is not fixed and is affected by the number of times OBL operator is performed during the run, as each time OBL operator is performed, one solution should be evaluated. We call MOA with the OBL procedure as Opposition-based Magnetic Optimization Algorithm (OMOA) to distinguish it from the original version of MOA.

Figure 4 shows the RLDs plot of MOA and OMOA on f_2 as a representative of unimodal problems and f_{10} as a representative of unimodal problems. The number in the brackets in each plot represents the pre-determined solution quality q' for each problem.

As shown in figure 4, OMOA reaches the predetermined solution quality much faster than MOA. We performed a similar test on all unimodal ($f_1 - f_9$) and mulimodal ($f_{10} - f_{27}$) problems and achieved similar results, although the results were not reported in this paper.

Accepted manuscript

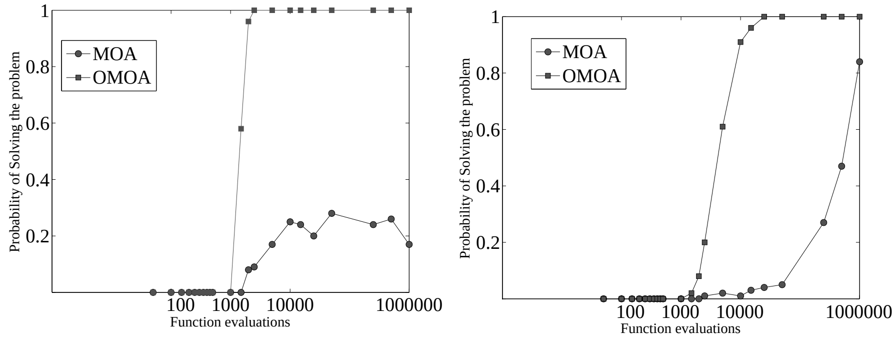
ACCEPTED MANUSCRIPT

Table 7: The benchmark problems used in this paper.

benchmark function	Search range	Reference
$f_1(x) = -\sum_{i=1}^{n} x_i^2$	[-100,100]	[43]
$f_2(x) = -\sum_{i=1}^{n} \|x_i\| - \prod_{i=1}^{n} \|x_i\|$	[-10,10]	[43]
$f_3(x) = -\max_i \{\|x_i\|, 1 \leq i \leq n\}$	[-100,100]	[43]
$f_4(x) = -\sum_{i=1}^{n-1} 100 \left(x_{i+1} - x_i^2\right)^2 - (1 - x_i)^2$	[-2,2]	[44]
$f_5(x) = -\sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x_i^2$	[-10,10]	[44]
$f_6(x) = -\sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} z_i^2,\ z = x \times M$	[-10,10]	[45]
$f_7(x) = f_{rot-elliptic}[z(P_1 : P_m)] * 10^6 + f_{elliptic}[z(P_{m+1} : P_n)]$	[-10,10]	[45]
$f_8(x) = f_{rot-schwefel}[z(P_1 : P_m)] * 10^6 + f_{schwefel}[z(P_{m+1} : P_n)]$	[-100,100]	[45]
$f_9(x) = -\sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	[-65.5,-65.5]	[43]
$f_{10}(x) = \sum_{i=1}^{n} \left(x_i \sin \sqrt{\|x_i\|}\right)$	[-500,500]	[43]
$f_{11}(x) = -\sum_{i=1}^{n} (x_i^2 - 10 \cos(2\pi x_i) + 10)$	[-5.12,5.12]	[43]
$f_{12}(x) = 20 \exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) + \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right) - 20 - e$	[-32,32]	[43]
$f_{13}(x) = \frac{-1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600,600]	[43]
$f_{14}(x) = -\frac{\pi}{n}\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2 \times [1 + 10\sin^2(\pi y_{i+1})]$ $+(y_n - 1)^2\} - \sum_{i=1}^{n} u(x_i, 10, 100, 1)$	[-50,50]	[43]
$f_{15}(x) = \frac{-1}{10}\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2 \left[1 + \sin^2(3\pi x_{i+1})\right]$ $+(x_n - 1)^2 \left[1 + \sin^2(2\pi x_n)\right]\} - \sum_{i=1}^{n} u(x_i, 5, 100, 1)$	[-50,50]	[43] [43]
$f_{16}(x) = \sum_{i=1}^{n} \left(\sin(x_i) \times \left(\sin\left(\frac{ix_i^2}{\pi}\right)\right)^{2n}\right)$	$[-\pi, \pi]$	[46]
$f_{17}(x) = \sum_{i=1}^{n} g_i h_i$ $g_i = [\sin(5\pi x_i + 0.5)]^2, h_i = \exp\left(-2.0\log(2.0)\frac{(x_i-0.1)^2}{0.64}\right)$	[-1,1]	[47]
$f_{18}(x) = -\sum_{i=1}^{n} ix_i^4 - \text{Gauss}(0, 1)$	[-10,10]	[46]
$f_{19}(x) = f_{rot-rastrigin}[z(P_1 : P_m)] * 10^6 + f_{rastrigin}[z(P_{m+1} : P_n)]$	[-5.12,5.12]	[45]
$f_{20}(x) = f_{rot-ackley}[z(P_1 : P_m)] * 10^6 + f_{ackley}[z(P_{m+1} : P_n)]$	[-32,32]	[45]
$f_{21}(x) = f_{rot-rosenbrock}[z(P_1 : P_m)] * 10^6 + f_{rosenbrock}[z(P_{m+1} : P_n)]$	[-2,2]	[45]
$f_{22}(x) = \sum_{i=1}^{\frac{D}{2m}} f_{rot-rastrigin}[z(P_{(k-1)*m+1} : P_{k*m})] + f_{rastrigin}[z(P_{\frac{D}{2}+1} : P_D)]$	[-2,2]	[45]
$f_{23}(x) = \sum_{i=1}^{\frac{D}{2m}} f_{rot-ackley}[z(P_{(k-1)*m+1} : P_{k*m})] + f_{ackley}[z(P_{\frac{D}{2}+1} : P_D)]$	[-2,2]	[45]
$f_{24}(x) = \sum_{i=1}^{\frac{D}{2m}} f_{rosenbrock}[z(P_{(k-1)*m+1} : P_{k*m})] + f_{sphere}[z(P_{\frac{D}{2}+1} : P_D)]$	[-2,2]	[45]
$f_{25}(x) = \sum_{i=1}^{\frac{D}{m}} f_{rot-rastrigin}[z(P_{(k-1)*m+1} : P_{k*m})]$	[-2,2]	[45]
$f_{26}(x) = \sum_{i=1}^{\frac{D}{m}} f_{rot-ackley}[z(P_{(k-1)*m+1} : P_{k*m})]$	[-2,2]	[45]
$f_{27}(x) = \sum_{i=1}^{\frac{D}{m}} f_{rosenbrock}[z(P_{(k-1)*m+1} : P_{k*m})]$	[-2,2]	[45]

Accepted manuscript

ACCEPTED MANUSCRIPT



(a) $f_1(-5500\%)$ (b) $f_{10}(9000\%)$

Figure 4: RLDs on f_1 and f_{10} benchmark functions for MOA and OMOA. Each RLD is obtained with 50 particles. The results are averaged over 100 independent runs and 10^6 function evaluations.

Another interesting characteristic seen in figure 4 is the 'slope' of the curves that represents useful information about the behavior of the algorithms [49]. A steeper RLDs indicates that the algorithm reaches the pre-specified quality much easier. OMOA has steep curves; conversely, the MOA algorithm has less steep curves indicating that the problem is challenging for the algorithm.

The stagnation behavior is another feature that can be inferred from RLDs. An algorithm shows the stagnation behavior whenever its performance growth significantly declines or stops advancing. As shown in figure 4, while MOA shows strong stagnation behavior, OMOA shows steady progress.

The comparisons suggest that applying the OBL significantly improves both the performance and convergence rate of the algorithm.

5.3. Boundary Handling

In order to detect and handle the infeasible solutions generated during the optimization process, there are different boundary constraint handling techniques. Among them random reinitialization (RR) strategy is the most unbiased technique [50, 51]. In this strategy, the variables that violate the boundaries (upper or lower boundary) are randomly reinitialized within the range of l_b and u_b where l_b, u_b are the lower and the upper boundaries of variables for a specific problem. Another strategy that has recently come to surface is Boundary Adjustment (BA) technique [52], in which the variables that exceed boundaries are projected on bounds. The last strategy studied in this paper is the Reflection Strategy (RS) that is proposed by Jani et al. in [53]. In this technique, the variables that violate the boundaries are fixed by reflecting back from the infeasible values.

In this section, in order to evaluate these strategies, we use the 27 benchmark problems where $n = 100$. In order to make a fair comparison, parameters and population size are set according to table 12,table 17 and table 10. For all the algorithms, the number of FE is set to 50000. Table 8 summarizes the results of this experiment for different boundary handling techniques.

As shown in table 8, the stochastic strategy has achieved the best results for all the algorithms except OMOA and BBO. Therefore, hereafter, we use the BA strategy for all the algorithms except OMOA and BBO and for the OMOA and BBO, we employ the RR strategy for the boundary handling.

5.4. Solution Representation

Usually, a real-encoding scheme is used by research papers [22, 37, 30, 29, 31, 41] for mathematical optimization problems, including the ones we have used here for the test functions. However, in this section in order to investigate the efficiency of the real-encoding scheme, we compare the scheme with integer-encoding and binary-encoding schemes. For the sake of comparison, all the 27 problems are used where $D = 100$. Like the previous section, the number of FE for all the algorithms is set 50000 and the population

Accepted manuscript

ACCEPTED MANUSCRIPT

Table 8: The best handling constraint technique for all the algorithms and for all the 27 benchmark functions where $D = 100$.

Problem	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
FMOA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
MOA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
OMOA	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR
GA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
PSO	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
DE	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
ES	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
FES	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
EP	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
FEP	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
3SOME	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR
PMS	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR
MASW	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR
MASSW	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
CCPSO2	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
JADE	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
BBO	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR
ODE	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
CMAES	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA

Problem	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}
FMOA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
MOA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
OMOA	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR
GA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
PSO	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
DE	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
ES	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
FES	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
EP	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
FEP	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
3SOME	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR
PMS	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR
MASW	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR
MASSW	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
CCPSO2	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
JADE	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
BBO	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR	RR
ODE	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA
CMAES	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA	BA

Accepted manuscript

ACCEPTED MANUSCRIPT

Table 9: The best handling constraint technique for all the algorithms and for all the 27 benchmark functions where $D = 100$. The Real stands for real-encoding and Integer for integer-encoding and Binary for binary-encoding schemes.

Problem	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
FMOA	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
MOA	Real	Real	Real	Real	Real	Integer	Real	Real	Real	Real	Real	Real	Real	Real
OMOA	Real	Real	Integer	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
GA	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Integer	Real	Real
PSO	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Integer	Real	Real	Real
DE	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
ES	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
FES	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
EP	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
FEP	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
3SOME	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
PMS	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
MASW	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
MASSW	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
CCPSO2	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
JADE	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
BBO	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
ODE	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
CMAES	Real	Real	Real	Real	Real	Real	Real	Integer	Real	Real	Real	Real	Real	Real

Problem	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}
FMOA	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
MOA	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
OMOA	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
GA	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
PSO	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
DE	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
ES	Real	Real	Real	Real	Integer	Real	Integer	Real	Real	Real	Real	Real	Real
FES	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
EP	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
FEP	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
3SOME	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
PMS	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
MASW	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
MASSW	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
CCPSO2	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
JADE	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
BBO	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real
ODE	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Integer	Real	Real
CMAES	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real	Real

size and the parameters are set according to table 10, table 12 and table 17. The boundary violations are also fixed and repaired according to the best boundary handling method for each algorithm in table 8. Table 9 represents the averaged fitness of the algorithms where different solution representations are used and all the 27 numerical problems with $100 - D$ are utilized.

As shown in table 9, the best encoding scheme is real as it helps the algorithms achieve the best results for most of the problems. The reason behind this is obvious. Since the benchmark problems used in this paper are encoded with real values, the best values are obtained when the solutions are formatted with real values. Hereafter we use the real-encoding scheme for all the experiments.

Accepted manuscript

5.5. Population Setting

In population-based algorithms, the size of population affects the performance as it has a significant effect on the diversity of solutions. In this section, we study the population size and it's effect on the performance of the algorithms. To do so, different population sizes N_p=1,5,10,25,50,100,200,500 are considered and for the test functions, all 27 problems with $D = 100$ are used. To make a fair comparison, the number of FE for each of the experiments is set to 50000; that is for example if the number of iterations is set to 2000, the size of the population is $N_p = 25$. The parameters of the algorithms are also set according to table 12 and table 17. For the boundary handling and solution representation, we apply the best boundary handling technique and the best solution representation found in table 8 and table 9, respectively. Table 10 lists the best population size for all the algorithms on all the benchmark problems.

ACCEPTED MANUSCRIPT

Table 10: The best population size for FMOA, MOA, GA, PSO, DE, ES, FES, EP, FEP, 3SOME, PMS, MASW, MASSW, CCPSO2, JADE, BBO and ODE on all benchmark functions where $D = 100$.

Algorithm	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
FMOA	50	25	50	50	100	100	200	50	100
MOA	500	200	200	200	500	200	500	500	5
GA	25	50	100	25	500	50	50	50	100
PSO	100	100	100	100	100	100	100	100	100
DE	5	25	5	5	100	50	5	5	100
ES	200	200	200	200	100	200	200	200	200
FES	100	100	100	100	100	100	200	100	100
EP	200	50	500	100	50	100	10	500	200
FEP	10	25	200	50	100	100	50	200	500
3SOME	100	100	200	5	25	50	200	50	500
PMS	50	200	500	200	50	50	100	200	200
MASW	100	200	200	500	500	200	500	200	500
MASSW	50	50	100	25	100	50	100	50	100
CCPSO2	25	5	5	5	200	10	25	100	100
JADE	100	100	200	50	100	100	200	100	200
BBO	100	100	100	100	100	100	200	100	50
ODE	5	5	5	5	5	100	200	200	5

Algorithm	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
FMOA	500	25	50	100	200	5	50	50	25
MOA	500	5	200	200	500	100	50	50	25
GA	50	25	25	50	25	50	100	50	25
PSO	100	100	100	100	100	100	100	100	100
DE	5	5	25	5	5	5	5	25	5
ES	200	200	200	200	200	100	200	100	100
FES	100	100	100	100	100	100	200	100	100
EP	500	500	500	200	100	10	50	50	100
FEP	500	100	200	100	100	25	500	100	100
3SOME	50	500	200	50	500	25	500	10	25
PMS	200	50	10	25	200	5	500	10	25
MASW	200	500	200	200	100	200	200	100	50
MASSW	25	100	10	100	50	10	25	50	50
CCPSO2	5	5	5	5	5	5	5	5	5
JADE	100	100	100	100	100	100	25	100	100
BBO	100	100	100	100	100	100	100	200	100
ODE	200	5	5	5	5	5	5	200	50

Algorithm	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}
FMOA	50	25	50	50	25	100	50	50	50
MOA	500	5	100	100	50	50	50	100	50
GA	500	500	50	50	25	50	100	50	100
PSO	100	100	100	100	100	100	100	100	100
DE	25	5	25	5	5	25	5	5	5
ES	100	100	200	200	200	200	200	200	100
FES	100	100	100	100	100	100	200	100	100
EP	100	50	200	500	50	50	200	10	50
FEP	50	100	200	100	100	50	200	25	25
3SOME	10	10	50	50	25	5	500	25	50
PMS	500	10	25	50	50	25	100	5	500
MASW	500	200	200	500	200	200	200	200	500
MASSW	50	100	100	50	25	50	100	50	50
CCPSO2	5	5	5	25	25	100	100	5	5
JADE	200	100	50	100	200	50	200	100	200
BBO	100	50	100	100	100	100	100	100	100
ODE	200	5	5	5	5	100	100	5	5

Accepted manuscript

ACCEPTED MANUSCRIPT

Table 11: A brief description of the parameters of the traditional algorithms.

Algorithms	Parameters	Description
GA	M R	Mutation rate Crossover rate
PSO	C W	Acceleration coefficient factor Inertia weight factor
DE	F O	The differential amplification factor Crossover rate
ES	L S	The number of breeds produced at each generation The regulator parameter
FES	L S	The number of breeds produced at each generation The regulator parameter
EP	T	The tournament size for recombination operator
FEP	T	The tournament size for recombination operator

As shown in table 10, the best population size for each algorithm is different and varies from 5 to 500. In Some algorithms like PSO, ES, FES and BO, the best population size for different problems is not significantly different, but in some other algorithms like ODE, 3SOME, PMS, the best population size significantly changes. Hereafter, we will use the best population size found here for each algorithm and for each problem.

Contrary to other algorithms that their population size is determined beforehand, in CMAES algorithm, the size of population is related to the problem size (D) and as D grows the population size logarithmically increases. It is defined based on the equation presented in [54].

5.6. Comparison with popular optimization algorithms

In this part, we compare the proposed algorithm with GA [29], PSO [30], DE [31], ES [32], FES [33], EP [34] and FEP [35]. We first set the parameters of each algorithm to make a fair comparison between them. Before setting the parameters, we provide a short description of the parameters of the algorithms, represented in Table 11.

To set the parameters, we utilize the systematic parameter setting [14] method described in section 4.2. As mentioned before, since the proposed algorithm uses the parameter adaptation technique, it has no parameter to set. Table 12 exhibits the best parameters for each well-known population-based algorithm for each benchmark function. The results are averaged over 50 runs.

After setting the parameters, we run each algorithm with the best parameters reported in table 12, the best population size found in table 10 and the best handling technique and the best solution representation reported in table 8 and table 9, respectively. The means and standard deviation of results obtained by the proposed algorithm and the other algorithms for 9 unimodal and 18 multi-modal problems are summarized in Table 13 and 14.

As shown in table 13, for all problem sizes ($D = 100, 500$ and 1000) FMOA offers the best performance on f_1-f_5 and f_9 and DE on f_6 and f_8. For f_7, when the size of the problem is equal to 100, DE is superior, but when the size of the problem increases it loses to FES.

As seen in table 14, for $D = 100$, FMOA is superior in terms of solution quality over the other algorithms on 7 out of 9 multi-modal functions, GA on f_{10} and f_{16}, FES on f_{19}, f_{20} and DE on f_{21}. Similarly, for $D = 500$ and 1000 the proposed algorithm outperforms the other algorithms on 7 multi-modal problems, GA on f_{10} and f_{16}, FES on f_{19}, f_{20} and f_{21}.

From table 13 and 14, it can be observed that FMOA outperforms the other population based algorithms on the both unimodal and multi-modal problems.

5.7. Comparison with State-of-the-art Optimization Algorithms

In this section, we compare the proposed algorithm with nine state-of-the-art algorithms, including MASW [36], MASSW [36], CCPSO2 [37], JADE [22], 3SOME [38], PMS [39], BBO [40], ODE [41] and

Accepted manuscript

ACCEPTED MANUSCRIPT

Table 12: Best parameters for GA, PSO, DE, ES, FES, EP and FEP on all benchmark functions.

Algorithm	Parameter	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
GA	R	1	1	1	1	1	0.8	0.4	0.4	1
	M	0.003	0.05	0.003	0.003	0.003	0.01	0.01	0.01	0.003
PSO	C	0.01	0.01	0.01	0.01	0.01	1	1	1	0.5
	W	1	1	1	1	0	1	0.5	0.5	0.5
DE	O	0.01	0.4	0.1	0.4	0.1	0.01	0.8	0.1	0.01
	F	0.8	0.1	0.1	0.1	0.4	0.8	0.1	0.4	1.2
ES	L	1	1	1	1	1	1	1	1	1
	S	1.1	1.1	1.1	1.1	1.1	1.5	1.1	1.1	1.1
FES	L	1	1	1	1	1	0.8	1	1	1
	S	1.1	1.1	1.1	1.1	1.1	2	1.1	1.1	1.1
EP	T	0.1	0.1	0.2	0.1	0.1	0.4	0.1	0.1	0.3
FEP	T	0.7	0.1	0.5	0.4	0.6	0.2	1	0.1	1
Algorithm	Parameter	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
GA	R		1	1	1	1	1	1	1	0.8
	M	0.003	0.003	0.01	0.003	0.003	0.003	0.05	0.01	0.01
PSO	C	0.01	0.01	1	0.01	0.01	0.01	1	1	0.01
	W	1	1	0.5	1	1	1	0.5	0.5	0.5
DE	O	0.01	0.1	0.1	0.1	0.4	0.4	0.01	0.8	0.4
	F	0.4	0.4	0.01	0.1	0.1	0.1	1.2	0.1	0.1
ES	L	1	1	1	1	1	1	1	1	1
	S	1.1	1.1	1.2	1.1	1.1	1.1	1.1	5	1.1
FES	L	1	1	1	1	1	1	1	1	1
	S	1.1	1.1	1.2	1.1	1.1	1.1	1.1	1.1	1.1
EP	T	0.2	0.2	0.9	0.4	0.2	0.3	0.6	0.1	0.3
FEP	T	0.2	0.5	0.1	0.9	0.1	0.4	0.4	0.5	0.6
Algorithm	Parameter	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}
GA	R	0.2	1	1	1	0.4	1	0.4	0.2	0.4
	M	0.003	0.05	0.003	0.005	0.5	0.5	0.5	0.5	0.5
PSO	C	1	1	0.01	0.001	1.5	1.5	1.5	1.5	0.001
	W	0.5	1	1	1	0.73	0.73	0.73	0.73	1
DE	O	0.01	0.4	0.4	0.8	0.2	0.4	0.2	0.2	0.4
	F	0.8	0.1	0.1	0.1	0.5	0.1	0.5	0.5	0.1
ES	L	1	0.6	1	1	0.8	0.6	1	1	0.8
	S	1.1	1.1	1.1	1.01	1.01	1.01	1.01	1.01	1.01
FES	L	1	1	1	1	1	1	1	1	1
	S	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
EP	T	0.1	0.7	0.1	1	0.5	0.1	0.5	0.3	0.5
FEP	T	0.4	1	0.7	0.4	0.5	0.8	0.2	1	0.4

Accepted manuscript

ACCEPTED MANUSCRIPT

Table 13: The experimental results for FMOA, GA, PSO, DE, ES, FES, EP and FEP on unimodal benchmark problems f_1-f_9. The best results are bolded.

		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
D=100										
FMOA	Mean	**-1.67e-1**	**-3.60e-1**	**-1.04e-1**	**-9.80e+1**	**-1.62e-9**	-6.24e-7	-5.13e-3	-1.48e+3	**-1.02e+6**
	Std	**8.00e-2**	**6.38e-2**	**2.06e-2**	**3.21e-3**	**9.59e-10**	2.16e-6	7.89e-4	3.44e+2	**5.81e+6**
GA	Mean	-6.69e+2	-1.58e+1	-4.33e+1	-3.99e+2	-6.08e-6	-2.02e-4	-6.27e-3	-1.69e+3	-1.32e+8
	Std	1.21e+2	1.39e+0	4.79e+0	5.58e+1	1.19e-6	4.69e-5	6.34e-4	2.29e+2	1.93e+7
PSO	Mean	-1.55e+4	-1.30e+2	-3.42e+1	-8.70e+2	-1.56e-4	-5.10e-5	-3.98e-3	-1.10e+3	-4.12e+8
	Std	2.76e+3	1.42e+1	3.36e+0	1.68e+2	2.75e-5	1.07e-5	2.67e-4	1.38e+2	6.06e+7
DE	Mean	-4.57e+3	-1.52e+2	-5.57e+1	-5.09e+2	-4.41e-5	**-5.47e-9**	**-1.27e-3**	**-3.13e+2**	-7.23e+7
	Std	5.61e+2	7.17e+0	1.81e+0	5.24e+1	6.31e-6	**1.74e-9**	**2.21e-4**	**2.90e+1**	7.49e+6
ES	Mean	-6.66e+3	-3.89e+1	-5.88e+1	-7.03e+2	-7.30e-5	-6.33e-5	-2.03e-3	-8.07e+2	-1.77e+8
	Std	1.28e+3	4.67e+0	2.92e+0	9.83e+1	1.23e-5	2.06e-5	3.86e-4	1.04e+2	4.37e+7
FES	Mean	-5.36e+3	-3.05e+1	-6.16e+1	-6.69e+2	-5.66e-5	-6.62e-5	-1.67e-3	-7.50e+2	-1.83e+8
	Std	8.83e+2	3.95e+0	3.06e+0	7.52e+1	1.00e-5	2.22e-5	3.65e-4	1.72e+2	4.31e+7
EP	Mean	-2.04e+5	-3.31e+038	-8.90e+1	-1.52e+4	-2.05e-3	-2.97e-4	-6.97e-3	-2.01e+3	-9.00e+8
	Std	1.13e+4	1.76e+039	2.01e+0	1.28e+3	1.08e-4	1.19e-4	1.12e-3	2.96e+2	2.49e+8
FEP	Mean	-1.62e+5	-7.05e+029	-8.48e+1	-1.49e+4	-1.63e-3	-1.09e-4	-4.80e-3	-1.22e+3	-3.69e+8
	Std	1.00e+4	2.05e+030	1.84e+0	1.10e+3	9.99e-5	1.94e-5	4.38e-4	1.79e+2	3.82e+7
D=500										
		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
FMOA	Mean	**-2.19e+0**	**-3.92e-1**	**-2.53e-2**	**-6.31e+2**	**-1.50e-6**	-2.21e-7	-2.06e-2	-1.58e+3	**-2.51e+8**
	Std	**1.52e+1**	**6.94e-2**	**4.25e-3**	**3.54e+2**	**1.06e-5**	6.41e-7	1.46e-3	4.13e+2	**9.69e+8**
GA	Mean	-1.84e+5	-5.50e+2	-8.31e+1	-1.54e+4	-1.85e-3	-4.71e-6	-3.19e-2	-1.73e+3	-9.77e+10
	Std	1.09e+4	2.02e+1	1.09e+0	9.41e+2	9.05e-5	1.66e-6	1.08e-3	3.00e+2	7.53e+9
PSO	Mean	-7.26e+4	-6.40e+2	-4.01e+1	-3.98e+3	-7.20e-4	-1.09e-6	-2.22e-2	-1.09e+3	-2.58e+11
	Std	8.17e+3	4.96e+1	3.21e+0	4.86e+2	9.54e-5	3.22e-7	5.57e-4	1.21e+2	2.89e+10
DE	Mean	-4.38e+5	-6.45e+125	-8.71e+1	-3.34e+4	-4.36e-3	**-1.31e-10**	-1.69e-2	**-3.11e+2**	-7.73e+10
	Std	1.05e+4	2.13e+126	6.48e-1	1.36e+3	1.33e-4	**4.97e-11**	6.13e-4	**3.05e+1**	6.33e+9
ES	Mean	-7.12e+5	-2.26e+117	-9.76e+1	-4.01e+4	-7.20e-3	-2.12e-6	-1.62e-2	-9.38e+2	-4.18e+11
	Std	5.32e+4	1.60e+118	2.94e-1	3.58e+3	4.93e-4	9.36e-7	9.19e-4	1.78e+2	2.34e+11
FES	Mean	-2.05e+5	-7.06e+2	-8.62e+1	-1.42e+4	-2.06e-3	-1.90e-6	**-1.11e-2**	-7.29e+2	-1.26e+11
	Std	1.37e+4	3.61e+1	1.20e+0	1.43e+3	1.43e-4	5.43e-7	**1.09e-3**	1.25e+2	3.08e+10
EP	Mean	-1.28e+6	-1.48e+234	-9.76e+1	-1.16e+5	-1.27e-2	-9.68e-6	-3.13e-2	-1.61e+3	-7.28e+11
	Std	2.44e+4	Inf	3.18e-1	3.28e+3	3.36e-4	1.03e-5	1.03e-3	3.21e+2	2.11e+11
FEP	Mean	-1.18e+6	-3.62e+220	-9.69e+1	-1.35e+5	-1.18e-2	-2.31e-6	-2.76e-2	-1.20e+3	-2.33e+11
	Std	2.17e+4	Inf	2.94e-1	4.23e+3	2.08e-4	5.62e-7	8.01e-4	1.74e+2	2.18e+10
D=1000										
		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
FMOA	Mean	**-1.81e+3**	**-**	**-1.36e-2**	**-1.03e+3**	**-3.93e-5**	-5.16e-7	-4.44e-2	-2.64e+3	**-1.85e+10**
	Std	**1.02e+4**	**-**	**2.63e-3**	**2.69e+2**	**1.39e-4**	1.60e-6	2.79e-3	6.23e+2	**7.89e+10**
GA	Mean	-7.85e+5	-	-9.12e+1	-6.70e+4	-7.76e-3	-9.69e-7	-6.48e-2	-1.93e+3	-1.68e+12
	Std	2.28e+4	-	4.67e-1	2.64e+3	2.13e-4	3.46e-7	1.67e-3	4.10e+2	1.22e+11
PSO	Mean	-1.32e+5	-	-4.21e+1	-7.20e+3	-1.30e-3	-1.98e-7	-4.59e-2	-1.13e+3	-3.99e+12
	Std	1.64e+4	-	3.09e+0	7.14e+2	1.26e-4	6.23e-8	9.13e-4	1.65e+2	5.27e+11
DE	Mean	-1.24e+6	-	-9.18e+1	-1.15e+5	-1.24e-2	**-2.41e-11**	-3.88e-2	**-3.22e+2**	-1.32e+12
	Std	2.40e+4	-	3.92e-1	2.74e+3	1.79e-4	**8.83e-12**	9.07e-4	**2.65e+1**	1.19e+11
ES	Mean	-1.65e+6	-	-9.88e+1	-9.89e+4	-1.64e-2	-4.35e-7	-3.59e-2	-8.72e+2	-1.25e+13
	Std	9.47e+4	-	1.47e-1	5.50e+3	1.00e-3	1.76e-7	1.46e-3	1.50e+2	1.83e+13
FES	Mean	-6.60e+5	-	-9.16e+1	-4.88e+4	-6.70e-3	-3.40e-7	**-2.52e-2**	-6.89e+2	-1.85e+12
	Std	3.32e+4	-	9.03e-1	2.81e+3	3.56e-4	1.18e-7	**1.45e-3**	1.16e+2	5.47e+11
EP	Mean	-2.67e+6	-	-9.87e+1	-2.51e+5	-2.67e-2	-2.48e-6	-6.22e-2	-1.69e+3	-1.25e+13
	Std	4.48e+4	-	1.88e-1	5.07e+3	4.41e-4	2.98e-6	1.42e-3	2.74e+2	3.48e+12
FEP	Mean	-2.54e+6	-	-9.84e+1	-3.03e+5	-2.54e-2	-4.57e-7	-5.74e-2	-1.21e+3	-3.64e+12
	Std	3.59e+4	-	1.91e-1	6.32e+3	2.44e-4	1.42e-7	1.31e-3	1.91e+2	4.79e+11

Accepted manuscript

ACCEPTED MANUSCRIPT

Table 14: The experimental results for FMOA, GA, PSO, DE, ES, FES, EP and FEP for multimodal benchmark functions f_{10}-f_{18}

. The best results are bolded.

		f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
						D=100				
FMOA	Mean	6.17e+3	**-8.74e-2**	**-6.58e-2**	**1.15e+0**	**1.61e+2**	**-1.05e-1**	1.57e+1	**9.88e+1**	**3.05e+0**
	Std	2.42e+3	**4.93e-2**	**1.34e-2**	**7.98e-2**	**5.10e+1**	**3.12e-3**	2.47e+0	**5.27e+0**	**3.02e-1**
GA	Mean	**4.05e+4**	-7.74e+1	-4.55e+0	-4.57e+0	1.52e+2	-2.51e+1	**7.01e+1**	9.69e+1	-9.17e+1
	Std	**2.58e+2**	6.71e+0	2.65e-1	1.06e+0	6.01e+0	4.89e+0	**1.90e+0**	2.71e-1	4.16e+1
PSO	Mean	1.09e+4	-8.65e+2	-1.29e+1	-1.37e+2	-3.42e+3	-1.80e+4	1.21e+1	6.46e+1	-3.06e+4
	Std	1.49e+3	3.27e+1	6.81e-1	2.54e+1	1.70e+3	3.54e+3	2.10e+0	2.39e+0	1.09e+4
DE	Mean	1.85e+4	-8.07e+2	-1.39e+1	-3.69e+1	-2.31e+1	-5.06e+3	1.95e+1	6.75e+1	-6.24e+3
	Std	5.49e+2	2.01e+1	2.62e-1	5.83e+0	1.03e+2	1.28e+3	6.76e-1	7.01e-1	1.62e+3
ES	Mean	2.97e+4	-1.79e+2	-9.55e+0	-5.79e+1	-1.39e+2	-2.91e+3	5.17e+1	9.13e+1	-2.51e+4
	Std	8.79e+2	1.95e+1	5.70e-1	1.23e+1	3.29e+2	1.07e+3	2.66e+0	1.00e+0	6.04e+3
FES	Mean	3.33e+4	-1.68e+2	-8.91e+0	-4.68e+1	4.90e+1	-1.76e+3	5.52e+1	9.22e+1	-1.83e+4
	Std	9.31e+2	1.39e+1	5.48e-1	9.36e+0	8.68e+1	8.79e+2	2.94e+0	8.75e-1	6.09e+3
EP	Mean	9.77e+3	-1.41e+3	-2.05e+1	-1.81e+3	-1.01e+5	-1.47e+5	8.21e+0	5.13e+1	-4.11e+6
	Std	8.72e+2	4.02e+1	8.08e-2	1.17e+2	6.07e+3	7.51e+3	6.49e-1	1.25e+0	4.94e+5
FEP	Mean	8.71e+3	-1.30e+3	-2.02e+1	-1.45e+3	-8.10e+4	-1.23e+5	6.98e+0	5.25e+1	-2.79e+6
	Std	6.17e+2	2.86e+1	9.48e-2	7.11e+1	4.51e+3	5.65e+3	5.79e-1	1.10e+0	2.87e+5
						D=500				
FMOA	Mean	1.41e+4	**-1.84e-2**	**-2.52e-1**	**-1.16e+1**	**3.35e+1**	**-1.91e+2**	1.70e+1	**4.47e+2**	**3.32e+0**
	Std	3.27e+3	**8.74e-3**	**1.19e+0**	**4.95e+1**	**1.92e+1**	**1.34e+3**	1.81e+1	**8.81e+1**	**3.09e-1**
GA	Mean	**1.50e+5**	-2.33e+3	-1.58e+1	-1.63e+3	-5.50e+4	-1.32e+5	**1.46e+2**	4.15e+2	-8.63e+6
	Std	**1.96e+3**	5.87e+1	1.68e-1	8.52e+1	4.13e+3	6.81e+3	**3.61e+0**	2.42e+0	6.83e+5
PSO	Mean	2.66e+4	-4.90e+3	-1.29e+1	-6.58e+2	-1.63e+4	-8.57e+4	1.91e+1	2.69e+2	-6.58e+5
	Std	3.45e+3	9.36e+1	4.60e-1	1.11e+2	6.47e+3	1.35e+4	2.33e+0	5.54e+0	1.50e+5
DE	Mean	4.56e+4	-6.34e+3	-1.99e+1	-3.92e+3	-1.93e+5	-3.53e+5	2.53e+1	2.61e+2	-2.46e+7
	Std	1.11e+3	5.29e+1	3.34e-2	1.06e+2	6.63e+3	9.04e+3	1.10e+0	1.90e+0	1.40e+6
ES	Mean	7.04e+4	-7.14e+3	-1.97e+1	-6.43e+3	-3.66e+5	-5.56e+5	2.42e+1	2.31e+2	-6.94e+7
	Std	7.76e+3	3.52e+2	1.67e-1	5.79e+2	3.55e+4	5.42e+4	2.35e+0	5.99e+0	8.37e+6
FES	Mean	1.09e+5	-2.71e+3	-1.62e+1	-1.86e+3	-7.40e+4	-1.73e+5	9.64e+1	3.89e+2	-8.91e+6
	Std	2.99e+3	1.08e+2	2.57e-1	1.40e+2	8.00e+3	1.08e+4	5.13e+0	3.89e+0	1.12e+6
EP	Mean	2.87e+4	-8.05e+3	-2.09e+1	-1.15e+4	-6.29e+5	-8.73e+5	1.28e+1	2.23e+2	-1.60e+8
	Std	1.76e+3	1.10e+2	2.71e-2	2.88e+2	1.35e+4	1.67e+4	7.88e-1	2.36e+0	6.30e+6
FEP	Mean	1.90e+4	-7.80e+3	-2.08e+1	-1.06e+4	-5.81e+5	-8.19e+5	9.80e+0	2.23e+2	-1.41e+8
	Std	1.54e+3	7.91e+1	2.82e-2	1.91e+2	9.80e+3	1.29e+4	8.19e-1	1.80e+0	3.90e+6
						D=1000				
		f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
FMOA	Mean	3.26e+4	**-6.63e+1**	**-9.91e-2**	**-5.40e+0**	**2.65e+1**	**-3.42e+1**	2.33e+1	**9.08e+2**	**-4.56e+0**
	Std	4.04e+4	**4.69e+2**	**6.37e-1**	**4.33e+1**	**1.33e+1**	**1.28e+2**	2.63e+1	**1.68e+2**	**5.59e+1**
GA	Mean	**2.24e+5**	-7.69e+3	-1.83e+1	-6.99e+3	-3.40e+5	-6.16e+5	**1.69e+2**	7.24e+2	-1.06e+8
	Std	**2.91e+3**	1.19e+2	8.57e-2	1.86e+2	1.22e+4	1.48e+4	**4.57e+0**	3.79e+0	4.67e+6
PSO	Mean	3.94e+4	-1.00e+4	-1.25e+1	-1.15e+3	-2.38e+4	-1.46e+5	2.32e+1	5.09e+2	-2.28e+6
	Std	5.74e+3	1.36e+2	3.70e-1	1.25e+2	7.82e+3	2.03e+4	2.73e+0	8.06e+0	4.34e+5
DE	Mean	6.51e+4	-1.39e+4	-2.03e+1	-1.11e+4	-5.78e+5	-9.42e+5	2.87e+1	4.86e+2	-2.03e+8
	Std	1.68e+3	5.43e+1	2.14e-2	2.10e+2	1.20e+4	1.20e+4	1.00e+0	2.24e+0	8.34e+6
ES	Mean	1.31e+5	-1.53e+4	-2.03e+1	-1.48e+4	-8.16e+5	-1.24e+6	3.13e+1	4.44e+2	-3.51e+8
	Std	1.04e+4	4.99e+2	1.12e-1	9.42e+2	4.56e+4	5.42e+4	3.41e+0	8.32e+0	3.49e+7
FES	Mean	1.74e+5	-7.40e+3	-1.79e+1	-6.03e+3	-2.97e+5	-5.67e+5	1.14e+2	7.11e+2	-7.38e+7
	Std	4.79e+3	1.72e+2	1.21e-1	2.81e+2	2.05e+4	2.46e+4	5.64e+0	7.10e+0	6.69e+6
EP	Mean	4.91e+4	-1.66e+4	-2.10e+1	-2.40e+4	-1.31e+6	-1.81e+6	1.60e+1	4.29e+2	-6.94e+8
	Std	2.57e+3	1.42e+2	1.78e-2	5.03e+2	2.17e+4	2.35e+4	9.32e-1	4.45e+0	1.89e+7
FEP	Mean	2.73e+4	-1.62e+4	-2.09e+1	-2.28e+4	-1.25e+6	-1.74e+6	1.12e+1	4.30e+2	-6.46e+8
	Std	1.94e+3	1.13e+2	1.69e-2	3.12e+2	1.41e+4	1.75e+4	7.62e-1	2.66e+0	1.13e+7

Accepted manuscript

ACCEPTED MANUSCRIPT

Table 15: The experimental results for FMOA, GA, PSO, DE, ES, FES, EP and FEP for f_{19}-f_{27}. The best results are bolded.

D=100										
Algorithm		f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}
FMOA	Mean	-1.77e+3	4.39e+1	-2.89e+4	**-1.45e+03**	-4.13e+01	-7.92e+10	-1.48e+03	-4.21e+01	-1.77e+11
	Std	9.02e+1	4.92e-2	1.74e+3	**1.05e+03**	2.64e-01	7.94e+09	3.99e+01	1.03e-01	7.24e+09
GA	Mean	-2.23e+3	4.39e+1	-8.14e+4	-1.31e+08	-4.20e+01	-1.29e+11	-1.79e+03	-4.24e+01	-4.67e+11
	Std	9.93e+1	3.31e-2	9.96e+3	1.93e+07	2.00e-01	1.74e+10	4.66e+01	1.19e-01	3.42e+10
PSO	Mean	-1.74e+3	4.40e+1	-2.83e+4	-1.18e+08	-4.04e+01	-4.34e+10	-1.53e+03	-4.16e+01	-1.75e+11
	Std	6.00e+1	6.97e-2	2.67e+3	2.57e+07	4.36e-01	9.45e+09	5.24e+01	3.05e-01	1.97e+10
DE	Mean	-1.32e+3	4.41e+1	**-1.29e+3**	-7.41e+07	-2.49e+01	-5.63e+06	-1.30e+03	-3.85e+01	-1.26e+09
	Std	4.33e+1	2.80e-2	**2.15e+2**	6.57e+06	1.23e+00	2.11e+06	3.67e+01	8.36e-01	4.74e+08
ES	Mean	-7.39e+2	4.43e+1	-3.13e+3	-3.87e+07	**-3.66e-02**	**-7.60e+01**	**-1.38e+02**	**-6.85e+00**	**-6.97e+03**
	Std	7.35e+1	6.19e-2	6.05e+2	8.13e+06	**2.07e-03**	**1.22e+01**	**2.15e+01**	**2.61e+00**	**7.26e+03**
FES	Mean	**-6.36e+2**	**4.43e+1**	-2.33e+3	-1.84e+08	-1.64e+01	-1.62e+08	-8.70e+02	-2.61e+01	-2.42e+09
	Std	**5.98e+1**	**5.33e-2**	5.28e+2	4.05e+07	2.00e+00	7.21e+07	6.04e+01	1.53e+00	7.40e+08
EP	Mean	-2.33e+3	4.39e+1	-7.93e+4	-3.28e+08	-4.03e+01	-4.61e+10	-1.52e+03	-4.17e+01	-2.30e+11
	Std	1.55e+2	3.71e-2	9.93e+3	3.07e+07	6.72e-01	1.27e+10	5.94e+01	4.17e-01	3.81e+10
FEP	Mean	-1.91e+3	4.39e+1	-4.51e+4	-3.66e+08	-4.08e+01	-5.51e+10	-1.57e+03	-4.19e+01	-2.66e+11
	Std	8.04e+1	3.72e-2	6.03e+3	4.76e+07	4.33e-01	1.30e+10	6.93e+01	2.90e-01	3.41e+10
D=500										
Algorithm		f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}
FMOA	Mean	-9.59e+3	4.37e+1	-2.31e+5	**-9.33e+04**	-1.25e+02	-4.06e+11	-8.33e+03	-2.11e+02	-9.30e+11
	Std	2.53e+2	4.27e-2	5.41e+3	**6.92e+04**	2.53e-01	1.42e+10	1.06e+02	3.18e-01	1.80e+10
GA	Mean	-1.27e+4	4.38e+1	-7.35e+5	-9.95e+10	-1.28e+02	-1.44e+12	-1.10e+04	-2.14e+02	-3.66e+12
	Std	2.37e+2	2.26e-2	2.88e+4	6.31e+09	1.94e-01	6.90e+10	1.27e+02	1.54e-01	9.97e+10
PSO	Mean	-1.03e+4	4.38e+1	-2.46e+5	-7.61e+10	-1.27e+02	-9.51e+11	-1.00e+04	-2.14e+02	-1.02e+12
	Std	1.53e+2	3.26e-2	1.26e+4	1.73e+10	3.04e-01	6.85e+10	1.61e+02	4.01e-01	3.96e+10
DE	Mean	-9.38e+3	4.39e+1	-1.66e+5	-7.75e+10	-1.22e+02	-1.00e+11	-8.86e+03	-2.12e+02	-7.66e+11
	Std	1.16e+2	1.88e-2	7.80e+3	6.16e+09	6.47e-01	1.08e+10	1.23e+02	6.93e-01	3.78e+10
ES	Mean	-7.83e+3	4.39e+1	-2.19e+5	-3.75e+10	**-4.71e+01**	**-4.11e+08**	**-1.99e+03**	**-1.29e+02**	**-5.60e+09**
	Std	2.96e+2	5.35e-2	2.21e+4	5.39e+09	**5.09e+00**	**1.13e+08**	**1.64e+02**	**8.55e+00**	**9.51e+08**
FES	Mean	**-5.10e+3**	**4.40e+1**	**-5.62e+4**	-1.25e+11	-9.83e+01	-3.83e+10	-5.77e+03	-1.87e+02	-2.12e+11
	Std	**2.43e+2**	**4.74e-2**	**5.52e+3**	2.90e+10	1.94e+00	6.79e+09	2.87e+02	2.69e+00	2.60e+10
EP	Mean	-1.24e+4	4.38e+1	-6.11e+5	-2.06e+11	-1.27e+02	-9.24e+11	-9.90e+03	-2.13e+02	-2.67e+12
	Std	2.63e+2	2.53e-2	2.44e+4	2.27e+10	5.02e-01	7.85e+10	1.95e+02	4.28e-01	1.39e+11
FEP	Mean	-1.17e+4	4.38e+1	-5.64e+5	-2.32e+11	-1.27e+02	-1.05e+12	-1.02e+04	-2.14e+02	-2.89e+12
	Std	1.75e+2	2.08e-2	2.15e+4	2.54e+10	2.66e-01	6.68e+10	1.30e+02	2.65e-01	1.37e+11
D=1000										
Algorithm		f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}
FMOA	Mean	-1.95e+4	4.37e+1	-4.87e+5	**-3.37e+10**	-2.29e+02	-9.85e+11	-1.72e+04	-4.23e+02	-1.95e+12
	Std	3.99e+2	3.88e-2	8.71e+3	**2.38e+11**	2.91e-01	1.69e+10	1.71e+02	3.61e-01	2.21e+10
GA	Mean	-2.61e+4	4.38e+1	-1.62e+6	-1.64e+12	-2.35e+02	-3.71e+12	-2.31e+04	-4.30e+02	-8.16e+12
	Std	3.31e+2	2.63e-2	3.93e+4	8.87e+10	2.09e-01	8.55e+10	1.86e+02	1.32e-01	1.66e+11
PSO	Mean	-2.14e+4	4.38e+1	-5.16e+5	-1.23e+12	-2.33e+02	-2.73e+12	-2.13e+04	-4.29e+02	-2.13e+12
	Std	2.17e+2	3.88e-2	1.57e+4	3.33e+11	4.53e-01	1.25e+11	2.34e+02	3.49e-01	5.49e+10
DE	Mean	-2.05e+4	4.39e+1	-6.01e+5	-1.29e+12	-2.29e+02	-7.69e+11	-1.93e+04	-4.28e+02	-2.98e+12
	Std	1.56e+2	1.73e-2	2.01e+4	1.15e+11	6.06e-01	4.17e+10	2.15e+02	5.33e-01	8.17e+10
ES	Mean	-1.71e+4	4.39e+1	-5.33e+5	-5.83e+11	**-1.53e+02**	**-2.29e+10**	**-6.23e+03**	**-3.22e+02**	**-1.37e+11**
	Std	4.98e+2	4.50e-2	3.71e+4	7.20e+10	**5.76e+00**	**3.50e+09**	**3.17e+02**	**7.63e+00**	**7.16e+10**
FES	Mean	**-1.22e+4**	**4.39e+1**	**-2.05e+5**	-1.86e+12	-2.02e+02	-2.14e+11	-1.27e+04	-3.96e+02	-8.84e+11
	Std	**4.14e+2**	**4.57e-2**	**1.59e+4**	4.58e+11	2.87e+00	2.95e+10	4.93e+02	2.97e+00	7.17e+10
EP	Mean	-2.52e+4	4.38e+1	-1.34e+6	-3.27e+12	-2.33e+02	-2.69e+12	-2.11e+04	-4.29e+02	-6.20e+12
	Std	3.36e+2	2.41e-2	4.26e+4	3.84e+11	4.35e-01	1.50e+11	3.37e+02	3.97e-01	2.71e+11
FEP	Mean	-2.43e+4	4.38e+1	-1.32e+6	-3.62e+12	-2.34e+02	-2.95e+12	-2.16e+04	-4.29e+02	-6.78e+12
	Std	2.59e+2	2.58e-2	3.90e+4	4.38e+11	3.42e-01	1.38e+11	2.71e+02	2.81e-01	1.66e+11

Accepted Manuscript

ACCEPTED MANUSCRIPT

CMAES [42]. Before the comparison, we first set the parameters of the algorithms represented in table 16 using the same parameter setting method employed for the population based algorithms.

We first provide a short description about each algorithms, describing the general features and major components of each algorithm.

ODE algorithm [41] is a conventional DE, which uses the OBL procedure in order to both accelerate the search process and help the algorithm escape from local optima. JADE [22] is another version of DE that employs a parameter adaptation strategy for dynamically setting its parameters and an external archive for storing the best solutions.

MASW [36] and MASSW [36] are two different versions of genetic memetic algorithm that use a typical version of GA as the global search and the Solis and Wet's algorithm as the local search and chaining mechanism for updating its local search. The only difference between them is that MASW utilizes single group while MASSW uses subgrouping mechanism for applying the local search on the solutions. They have gained remarkable results in CEC 2010 [45] by achieving the first place in the competition.

3SOME [38] is a single-solution memetic algorithm that has three memes for exploring the search space, one for long range, one for middle range and one for small range. It was proposed based on the idea of Ockham's razor that says the simpler an algorithm is, the more powerful it is to solve problems. PMS [39] is an extended version of 3SOME that utilizes three memetic operators for finding better solutions. The difference between 3SOME and PMS is the mechanisms used for implementing the memes.

BBO [40] is a population-based algorithm inspired by the biogeography research studies. Like the GA, it uses mutation operator to maintain its population diversity. The difference here is that GA has a reproduction operator to make an offspring, while BBO uses a similar method to DE and PSO. This means that through interacting with its neighboring particles, each particle attempts to find the best solution.

CCPSO2 [37] is a new variant of PSO algorithms that uses cooperative co-evolving strategy to deal with large-scale optimization problems. The co-evolving strategy is a new kind of divide-and-conquer strategy that divides a large-scale problem into a set of smaller sub-problems. It also employs a combination of Cauchy and Gaussian distribution for the movement operator. In order to provide further improvement, it uses dynamically changing-group-size strategy for interaction patterns of particles in the population.

CMAES is an extended version of evolution strategy algorithm, which has been extensively utilized by many research papers for unconstrained or bounded constraint optimization problems. Similar to quasi-Newton methods, the CMA-ES is considered as a second order approach, which estimates a positive definite matrix within an iterative procedure. Because of this, it can be feasible on non-separable and/or badly conditioned problems. Unlike quasi-Newton methods, it does not utilize or approximate gradients; thus, it is feasible on non-smooth and even non-continuous problems, multimodal and/or noisy problems [55] and for global optimization [42]. One important feature of this algorithm is that it does not need to employ parameter configuration, as its parameters are set during the optimization process.

Table 16: A brief description of the main parameters of the state-of-the-art algorithms used in this paper.

Algorithms	Parameters	Description
3SOME	I	inheritance factor
	D	The side width of the hypercube constructed
PMS	I	inheritance factor
	E	The size of the search region in the short-distance stage
MASW	B	mutation rate
	U	$\frac{local}{global}$ rate
MASSW	B	mutation rate
	U	$\frac{local}{global}$ rate
CCPSO2	P	probability of choosing cauchy-based operator
	Q	A set of several possible group sizes
JADE	H	rate of adaptation of parameters
	G	mutation greediness factor
BBO	S	The step size
	N	mutation probability
ODE	F	differential amplification factor
	O	mutation rate
CMAES	G	wise standard deviation coordination

The best parameters for each algorithm are reported in table 17.

Accepted manuscript

ACCEPTED MANUSCRIPT

Table 17: The best parameters for MASW, MASSW, CCPSO2, JADE, 3SOME, PMS, BBO, ODE andCMAES on all the benchmark functions.

Algorithm	Parameter	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
3SOME	I	0.3	0.6	0.9	0.9	0.6	0.9	0.6	0.1	0.3
	D	0.1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
PMS	I	0.05	0.3	0.05	0.05	0.3	0.01	0.05	0.01	0.05
	E	0.01	0.01	0.01	0.01	0.01	0.01	0.1	0.01	0.01
MASW	B	0.05	0.05	0.05	0.01	0.05	0.05	0.01	0.05	0.05
	U	0.9	0.9	0.3	0.5	0.3	0.5	0.5	0.5	0.5
MASSW	B	0.01	0.01	0	0.1	0.01	0.01	0.01	0.01	0.01
	U	0.5	0.1	0.1	0.3	0.3	0.1	0.5	0.5	0.3
CCPSO2	P	0.01	0	0.1	0.01	0	0.01	0.1	1	0.01
	Q	4	3	4	2	4	2	4	4	4
JADE	H	10^{-4}	0	10^{-4}	10^{-4}	0	0.01	0	0.01	0.1
	G	0.1	0.3	0.05	0.05	0.05	0.5	0.05	0.1	0.01
BBO	S	5	2	5	5	50	5	50	5	50
	N	0.01	0.01	0.01	0.05	0.05	0.1	0.05	0.01	0.01
ODE	O	0.01	0.4	0.1	0.4	0.1	0.01	0.8	0.1	0.01
	F	0.8	0.1	0.1	0.1	0.4	0.8	0.1	0.4	1.2
CMAES	G	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3
Algorithm	Parameter	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
3SOME	I	0.6	0.3	0.3	0.6	0.6	0.9	0.3	0.9	0.01
	D	0.01	0.4	0.01	0.01	0.01	0.01	0.4	0.1	0.01
PMS	I	0.01	0.05	0.3	0.01	0.01	0.05	0.01	0.05	0.1
	E	0.01	0.01	0.01	0.01	0.01	0.1	0.7	0.01	0.01
MASW	B	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.01	0.05
	U	0.5	0.9	0.1	0.3	0.5	0.5	0.1	0.9	0.3
MASSW	B	0.1	0.01	0.5	0	0	0	0.5	0.1	0.01
	U	0.3	0.5	0.5	0.3	0.3	0.1	0.3	0.7	0.3
CCPSO2	P	0.01	0	0	0.1	0	0	0	0.5	1
	Q	4	4	4	4	4	4	4	3	4
JADE	0.1	10^{-4}	10^{-4}	0	10^{-4}	0	0	0.01	0	0.1
	G	0.3	0.3	0.5	0.3	0.1	0.1	0.1	0.5	0.05
BBO	S	50	5	5	5	50	5	0.5	2	5
	N	0.05	0.05	0.01	0.05	0.01	0.05	0.1	0.1	0.05
ODE	O	0.01	0.1	0.1	0.1	0.4	0.4	0.01	0.8	0.4
	F	0.4	0.4	0.01	0.1	0.1	0.1	1.2	0.1	0.1
CMAES	G	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3
Algorithm	Parameter	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}
3SOME	I	0.6	0.6	0.6	0.6	0.6	0.9	0.6	0.3	0.9
	D	0.01	0.01	0.01	0.01	0.4	0.001	0.4	0.001	0.001
PMS	I	0.01	0.01	0.01	0.001	0.05	0.01	0.05	0.001	0.01
	E	0.01	0.1	0.01	0.3	0.1	0.1	0.1	0.1	0.3
MASW	B	0.05	0.05	0.05	0.01	0.01	0.05	0.001	0.001	0.01
	U	0.5	0.5	0.3	0.9	0.1	0.5	0.1	0.1	0.9
MASSW	B	0.01	0.01	0.01	0.05	0.0	0.0	0.0	0.01	0.1
	U	0.5	0.5	0.3	0.7	0.1	0.5	0.7	0.7	0.9
CCPSO2	P	0.1	0	0.01	1	0.5	1	1	1	1
	Q	4	4	4	2	4	4	1	2	4
JADE	H	0.01	0	0.01	0.05	0.05	0.05	0.1	0.05	0.05
	G	0.1	0.1	0.1	0.1	0	0	0.1	0.001	0.1
BBO	S	0.5	5	2	200	200	200	50	5	2
	N	0.01	0.05	0.05	0.005	0.01	0.001	0.01	0.01	0.001
ODE	O	0.01	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
	F	0.8	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
CMAES	G	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3

Accepted manuscript

ACCEPTED MANUSCRIPT

To make a fair comparison, the number of function evaluations, l, assigned to each algorithm is set to 50000, and the population size for each algorithm (except CMAES that has a dynamic population size setting) is set according to table 10. Note that for ODE and the proposed algorithm in which the OBL is involved, M_I is determined during the search process and, for MASW and MASSW that have a local search, it is determined by U the $\frac{local}{global}$ rate.

The experimental results for nine unimodal and eighteen multi-modal benchmark functions over several dimensions D=(100, 500 ,1000) are summarized and shown in table 18, 19 and 20.

As shown in table 18, for $D = 100$, CMAES performs the best on three unimodal problems and FMOA is the best on no problem. However, as the size of the problem increases, FMOA outperforms the other algorithms in some cases. More specifically, for $D = 100$, CMAES outperforms other algorithms in f_1, f_2 and f_3 and MASW performs the best on f_5 and f_7, JADE on f_4 and f_8, MASSW on f_9, 3SOME on f_6. For $D = 500$ FMOA is superior over the other algorithms on f_2, f_3, f_6 and f_9, MASSW on f_4, f_5 and f_7, CMAES on f_1 and JADE on f_8. For $D = 1000$, FMOA outperforms the other algorithms on f_3, f_4, MASSW on f_1 and f_7, JADE on f_6 and f_8, MASW on f_5 and CMAES on f_9.

As shown in table 19, for $D = 100$, FMOA performs the best on f_{11}, f_{12} and f_{18}, CCPSO2 on f_{14} and f_{17}, CMAES on f_{10} and f_{16}, MASW on f_{13} and f_{21}, JADE on f_{15}, PMS on f_{20} and BBO on f_{10}. As the problem size grows, FMOA shows steady performance. For $D = 500$, FMOA outperforms the other algorithms on f_{11}, f_{15}, f_{17} and $f_1 8$, CCPSO2 on f_{10} and f_{20}, JADE on f_{14} and f_{20}, CMAES on f_{10} and f_{12} MASW on f_{13}.

Table 13, 14, 18, 19 and 20, show that the performance of the proposed algorithm is improved as the problem size increases and it performs the best for some unimodal and multi-modal problems. The proposed algorithm does not offer the best results compared to some state-of-the-art algorithms (JADE, MASW and MASSW). Note that the advantage of our proposed algorithm is that it removes the parameter setting stage of the algorithms thus avoiding the time-consuming parameter configuration process.

In order to verify the significance of the experimental results we conduct the Wilcoxon Signed-Ranks Test [56, 57] between the proposed algorithm and other competitive algorithms. Table 21 represents the results two-tailed Wilcoxon Signed-Ranks Test where R^+ shows the sum of ranks for all the problems in which the proposed algorithm outperforms the second algorithm and R^- shows the sum of ranks for the opposite.

According to table 21, the proposed algorithm gained the sixth place after MASW, MASSW, CCPSO2, JADE and 3SOME where D is up to 100. However when the problem size grows, it gained better place. For $D = 500$ and $D = 1000$ the proposed algorithm gained the third position after MASSW and MASW.

In addition to Wilcoxon signed-ranks test, we also employ Friedman two-way analysis of variance by ranks [58] to show the significance of the results. Table 22 shows the results of Friedman test where the proposed algorithm is compared with both traditional and state-of-the art algorithms for unimodal and multimodal problems where $D = 100$, 500 and 1000.

As seen in Table 22, the proposed algorithm has achieved the sixth place for unimodal problems when the problem size is equal to 100. As the problem size grows, the algorithm gains better position (third rank for $D = 500$ and forth place for $D = 1000$). For the multimodal problems, it holds the sixth place when $D = 100$ and for $D = 500$ and $D = 1000$ it gets sixth and forth places, respectively. Like unimodal and multi-modal problems, in all the tests, the proposed algorithm gains the sixth place when $D = 100$ and for $D = 500$ and $D = 1000$, it holds the fifth place.

6. Conclusion

In this paper, we propose a new version of magnetic optimization algorithm called FMOA that is characterized by the parameter adaptation strategy and the OBL procedure as well as MOA. The proposed parameter adaptation strategy enables the algorithm to automatically set its control parameters to appropriate values during the evolutionary search. It is natural to use the adaptation technique along with an OBL technique to improve the convergence rate while keeping the robustness of the algorithm at high level.

Accepted manuscript

ACCEPTED MANUSCRIPT

Table 18: The experimental results for FMOA, MASW, MASSW, CCPSO2, JADE, 3SOME, PMS, BBO, ODE and CMAES for 9 unimodal benchmark functions. The results are obtained over 50 independent runs. The best results are typed in bold.

		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
D=100										
FMOA	Mean	-1.67e-1	-3.60e-1	-1.04e-1	-9.80e+1	-1.62e-9	-6.24e-7	-5.13e-3	-1.48e+3	-1.02e+6
	Std	8.00e-2	6.38e-2	2.06e-2	3.21e-3	9.59e-10	2.16e-6	7.89e-4	3.44e+2	5.81e+6
MASW	Mean	-1.32e-38	-1.29e+2	-5.34e+1	-9.63e+1	**-4.26e-41**	-8.34e-7	**-2.29e-5**	-7.34e+2	-1.29e+6
	Std	1.83e-38	1.44e+1	3.55e+0	7.76e+0	**2.58e-40**	4.48e-7	**1.65e-5**	8.02e+1	4.84e+5
MASSW	Mean	-8.67e-4	-1.73e+1	-3.58e+1	-9.64e+1	-1.47e-14	-6.52e-6	-5.93e-5	-6.33e+2	**-5.56e+5**
	Std	5.69e-3	5.88e+0	2.57e+0	1.52e+0	6.45e-14	2.38e-6	2.69e-5	6.33e+1	**3.50e+5**
CCPSO2	Mean	-4.58e+1	-3.27e+0	-5.15e+1	-1.29e+2	-4.18e-7	-7.72e-5	-6.34e-4	-1.11e+3	-8.70e+7
	Std	1.98e+1	1.36e+0	1.62e+1	1.81e+1	2.28e-7	4.44e-5	2.95e-4	2.09e+2	2.84e+7
JADE	Mean	-1.73e-8	-2.46e-3	-1.78e+1	**-9.17e+1**	-1.37e-16	-1.96e-6	-2.95e-5	**-4.03e+2**	-3.74e+6
	Std	3.01e-8	5.38e-3	2.08e+0	**1.71e+0**	1.77e-16	1.03e-6	8.48e-6	**6.99e+1**	8.62e+5
3SOME	Mean	-2.73e-1	-8.66e+1	-2.24e+1	-1.03e+2	-2.46e-9	**-3.96e-7**	-2.32e-5	-9.87e+2	-1.65e+6
	Std	8.11e-2	9.06e+1	7.37e+0	1.74e+1	6.44e-10	**1.95e-7**	9.08e-6	2.00e+2	3.61e+5
PMS	Mean	-5.84e+3	-1.49e+044	-7.98e+1	-1.34e+2	-9.64e-18	-5.63e-5	-2.63e-4	-1.63e+3	-3.84e+7
	Std	4.13e+4	9.96e+044	7.17e+0	4.34e+1	2.97e-17	8.66e-5	8.91e-4	6.41e+2	1.33e+8
BBO	Mean	-3.55e+4	-1.07e+2	-8.44e+1	-3.13e+3	-3.64e-4	-8.28e-5	-2.14e-3	-1.08e+3	-2.85e+8
	Std	7.18e+3	1.55e+1	3.49e+0	6.80e+2	7.86e-5	2.49e-5	4.65e-4	2.18e+2	5.43e+7
ODE	Mean	-1.05e+4	-1.21e+2	-6.50e+1	-1.02e+3	-1.02e-4	-5.37e-5	-4.70e-3	-1.31e+3	-2.43e+8
	Std	2.95e+3	2.42e+1	2.86e+0	2.01e+2	3.02e-5	1.49e-5	3.12e-4	1.42e+2	2.62e+7
CMAES	Mean	**-1.04e-41**	**1.33e+3**	**3.87e+40**	-9.91e+1	-1.50e+3	2.62e-5	2.42e-5	-1.62e+3	-3.92e+3
	Std	**6.09e-42**	**138.07**	**7.20e+40**	21.22	305.33	2.49e-5	3.02e-4	2.52e+2	3.79e+3
D=500										
		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
FMOA	Mean	-2.19e+0	**-3.92e-1**	**-2.53e-2**	-6.31e+2	-1.50e-6	**-2.21e-7**	-2.06e-2	-1.58e+3	**-2.51e+8**
	Std	1.52e+1	**6.94e-2**	**4.25e-3**	3.54e+2	1.06e-5	**6.41e-7**	1.46e-3	4.13e+2	**9.69e+8**
MASW	Mean	**-4.90e-3**	-3.37e+044	-7.07e+1	-1.20e+3	-6.45e-9	-1.51e-6	-1.10e-4	-6.88e+2	-1.01e+010
	Std	**1.87e-2**	2.37e+045	2.97e+0	4.84e+2	4.55e-8	4.89e-7	1.65e-4	8.68e+1	1.73e+9
MASSW	Mean	-2.75e-1	-1.39e+2	-4.73e+1	**-5.12e+2**	**-3.80e-9**	-5.87e-7	**-5.01e-5**	-6.59e+2	-2.40e+9
	Std	8.22e-1	1.85e+1	2.30e+0	**3.49e+1**	**1.51e-8**	1.50e-7	**2.33e-5**	6.35e+1	7.21e+8
CCPSO2	Mean	-1.02e+5	-4.10e+2	-8.47e+1	-1.09e+4	-1.05e-3	-4.69e-6	-7.75e-3	-1.19e+3	-6.52e+10
	Std	1.95e+4	8.46e+1	5.71e+0	9.04e+2	1.98e-4	1.97e-6	4.84e-4	2.50e+2	1.35e+10
JADE	Mean	-7.96e+3	-1.53e+2	-3.66e+1	-1.32e+3	-9.67e-5	-5.51e-7	-1.75e-3	**-4.03e+2**	-8.38e+9
	Std	1.51e+3	1.49e+1	1.47e+0	1.33e+2	1.67e-5	2.23e-7	2.25e-4	**1.24e+2**	1.21e+10
3SOME	Mean	-3.10e+5	-5.25e+96	-7.45e+1	-3.72e+3	-3.11e-3	-5.43e-7	-6.36e-5	-8.09e+2	-1.28e+010
	Std	2.58e+4	3.71e+97	2.65e+0	3.27e+2	2.58e-4	1.75e-7	1.03e-5	2.09e+2	8.84e+8
PMS	Mean	-7.35e+5	-2.88e+288	-9.52e+1	-8.56e+4	-9.04e-3	-2.79e-3	-5.45e-2	-4.06e+3	-1.07e+012
	Std	8.06e+5	Inf	4.19e+0	1.05e+5	8.11e-3	6.66e-3	7.64e-2	3.62e+3	1.37e+12
BBO	Mean	-6.32e+5	-2.46e+84	-9.70e+1	-6.46e+4	-6.44e-3	-2.97e-6	-2.05e-2	-1.27e+3	-1.50e+011
	Std	4.72e+4	1.74e+085	7.75e-1	8.47e+3	4.86e-4	9.53e-7	1.71e-3	2.43e+2	2.73e+10
ODE	Mean	-4.85e+5	-2.97e+41	-9.88e+1	-5.44e+4	-5.39e-3	-1.59e-6	-2.52e-2	-1.24e+3	-1.77e+11
	Std	1.71e+4	1.83e+42	3.00e-1	2.29e+3	2.72e-4	5.58e-7	7.37e-4	1.42e+2	2.11e+10
CMAES	Mean	**-3.91e-05**	-491.49	-20.42	-5.46e+2	-3.33e+6	-1.23e-6	-2.41e-2	-1.61e+3	-1.60e+9
	Std	**6.96e-6**	174.39	8.80	64.84	3.30e+4	2.36e-7	5.16e-4	4.44e+2	3.41e+10
D=1000										
		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
FMOA	Mean	-1.81e+3	-	**-1.36e-2**	**-1.03e+3**	-3.93e-5	-5.16e-7	-4.44e-2	-2.64e+3	-1.85e+10
	Std	1.02e+4	-	**2.63e-3**	**2.69e+2**	1.39e-4	1.60e-6	2.79e-3	6.23e+2	7.89e+10
MASW	Mean	-2.38e+2	-	-7.53e+1	-6.09e+3	**-9.02e-7**	-4.49e-7	-2.09e-3	-6.35e+2	-2.40e+11
	Std	1.04e+3	-	2.80e+0	3.21e+3	**1.12e-7**	2.14e-7	2.26e-3	7.49e+1	6.09e+10
MASSW	Mean	**-1.97e+2**	-	-5.87e+1	-1.05e+3	-1.56e-6	-1.49e-7	**-2.25e-4**	-5.88e+2	-4.23e+10
	Std	**1.56e+2**	-	2.42e+0	4.77e+1	1.20e-6	4.70e-8	**1.05e-4**	8.10e+1	9.90e+9
CCPSO2	Mean	-5.32e+5	-	-9.27e+1	-5.14e+4	-5.30e-3	-1.05e-6	-2.45e-2	-1.03e+3	-1.13e+12
	Std	6.87e+4	-	4.15e+0	3.51e+3	7.54e-4	5.14e-7	7.55e-4	2.08e+2	2.68e+11
JADE	Mean	-7.28e+4	-	-4.28e+1	-5.49e+3	-8.97e-4	**-1.20e-7**	-9.86e-3	**-3.98e+2**	-1.01e+11
	Std	6.38e+3	-	1.46e+0	4.08e+2	8.85e-5	**4.07e-8**	5.14e-4	**7.02e+1**	1.93e+10
3SOME	Mean	-1.13e+6	-	-8.69e+1	-2.43e+4	-1.12e-2	-2.47e-7	-3.44e-4	-6.85e+2	-2.71e+11
	Std	4.94e+4	-	1.26e+0	2.10e+3	5.19e-4	9.99e-8	2.70e-5	1.48e+2	1.96e+10
PMS	Mean	-1.49e+6	-	-9.81e+1	-2.08e+5	-1.62e-2	-1.24e-2	-9.35e-2	-6.41e+3	-1.54e+13
	Std	1.63e+6	-	2.16e+0	2.17e+5	1.64e-2	2.88e-2	9.27e-2	4.70e+3	1.94e+13
BBO	Mean	-1.66e+6	-	-9.85e+1	-1.79e+5	-1.72e-2	-5.39e-7	-4.58e-2	-1.08e+3	-2.18e+12
	Std	7.82e+4	-	3.02e-1	1.14e+4	1.11e-3	1.85e-7	2.59e-3	2.25e+2	4.40e+11
ODE	Mean	-1.46e+6	-	-9.95e+1	-1.65e+5	-1.50e-2	-4.28e-7	-5.15e-2	-1.51e+3	-2.92e+12
	Std	3.29e+4	-	1.43e-1	8.04e+3	1.40e-3	2.58e-7	9.52e-4	2.01e+2	3.65e+11
CMAES	Mean	-16.02	-	-173.90	-1.15e+3	-2.66e+6	-2.45e-7	-4.25e-2	-2.42e+3	**-3.63e+8**
	Std	1.19	-	11.08	143.78	1.73e+05	1.48e-7	4.71e-4	1.43e+2	**1.29e+8**

Accepted manuscript

ACCEPTED MANUSCRIPT

Table 19: The experimental results for FMOA, MASW, MASSW, CCPSO2, JADE, 3SOME, PMS, BBO and ODE for benchmark functions $f_{10} - f_{18}$. The results are obtained over 50 independent runs. The best results are typed in bold.

		f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
D=100										
FMOA	Mean	6.17e+3	**-8.74e-2**	**-6.58e-2**	1.15e+0	1.61e+2	-1.05e-1	1.57e+1	9.88e+1	**3.05e+0**
	Std	2.42e+3	**4.93e-2**	**1.34e-2**	7.98e-2	5.10e+1	3.12e-3	2.47e+0	5.27e+0	**3.02e-1**
MASW	Mean	2.40e+4	-4.36e+2	-1.27e+1	**2.00e+0**	9.04e+1	-8.37e+1	2.11e+1	8.34e+1	3.45e-1
	Std	8.61e+2	4.18e+1	6.97e-1	**2.14e-3**	5.55e+0	3.86e+1	2.25e+0	1.32e+0	1.04e-1
MASSW	Mean	2.42e+4	-1.13e+2	-6.91e+0	1.99e+0	9.35e+1	-1.94e+1	1.14e+1	9.42e+1	2.05e+0
	Std	1.31e+3	2.58e+1	5.31e-1	1.63e-2	6.59e+0	9.45e+0	1.90e+0	1.19e+0	4.80e-1
CCPSO2	Mean	4.16e+4	-2.30e+1	-1.99e+0	5.70e-1	**2.00e+2**	-1.30e+0	5.82e+1	**9.88e+1**	-5.25e+0
	Std	1.29e+2	6.73e+0	7.35e-1	1.52e-1	**6.27e+0**	6.15e-1	1.99e+0	**2.69e-1**	3.31e+0
JADE	Mean	1.44e+4	-5.87e+2	-1.50e+0	1.99e+0	1.38e+2	**-2.06e-2**	1.39e+1	6.94e+1	1.87e+0
	Std	2.20e+3	5.87e+1	3.38e-1	1.06e-2	9.66e+0	**4.66e-2**	9.08e-1	4.07e+0	8.02e-1
3SOME	Mean	2.42e+4	-3.46e+2	-1.36e+1	1.87e+0	1.26e+2	-6.98e+1	1.63e+1	8.88e+1	-6.52e-1
	Std	1.31e+3	6.73e+1	1.60e+0	3.38e-2	8.02e+0	3.54e+1	2.35e+0	1.87e+0	1.74e+0
PMS	Mean	2.43e+4	-4.73e+2	-1.85e+1	-4.42e+1	-2.32e+3	-3.35e+3	3.83e+1	8.44e+1	-1.07e+6
	Std	1.29e+3	1.72e+2	1.69e+0	3.26e+2	1.74e+4	2.37e+4	9.96e+0	4.99e+0	2.69e+6
BBO	Mean	4.44e+4	-4.47e+2	-1.58e+1	-3.18e+2	-1.44e+4	-2.79e+4	3.99e+1	8.43e+1	-4.42e+5
	Std	1.14e+3	4.59e+1	7.50e-1	6.35e+1	3.83e+3	5.71e+3	1.98e+0	1.72e+0	1.45e+5
ODE	Mean	1.49e+4	-7.00e+2	-1.21e+1	-1.54e+2	-2.25e+3	-1.34e+4	9.50e+0	5.95e+1	-3.56e+4
	Std	1.31e+3	1.32e+2	9.84e-1	2.30e+1	1.25e+3	3.09e+3	8.21e-1	1.44e+0	1.07e+4
CMAES	Mean	**7.02e+4**	-9.28e+2	-4.18e+1	-5.48e+53	8.54e+1	-3.22e-1	**7.71e+3**	6.55e+1	-3.25e+4
	Std	**1.92e+4**	3.83e+0	3.21e-2	1.03e+54	5.51e+0	6.12e-2	**7.11e+1**	1.29e+0	2.11e+4
D=500										
		f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
FMOA	Mean	1.41e+4	**-1.84e-2**	-2.52e-1	-1.16e+1	3.35e+1	**-1.91e+2**	1.70e+1	**4.47e+2**	**3.32e+0**
	Std	3.27e+3	**8.74e-3**	**1.19e+0**	4.95e+1	1.92e+1	1.34e+3	1.81e+1	**8.81e+1**	**3.09e-1**
MASW	Mean	1.09e+5	-2.70e+3	-1.72e+1	**1.96e+0**	5.72e+1	-1.98e+5	2.90e+1	4.05e+2	-6.38e-1
	Std	3.11e+3	1.33e+2	3.17e-1	**1.22e-1**	4.43e+0	2.29e+4	2.24e+0	4.98e+0	6.43e+0
MASSW	Mean	1.11e+5	-7.27e+2	-1.39e+1	1.83e+0	7.63e+1	-4.48e+3	1.99e+1	4.28e+2	-1.22e+3
	Std	8.26e+3	3.38e+2	4.37e-1	2.21e-1	4.68e+0	3.25e+3	2.95e+0	1.28e+1	6.57e+2
CCPSO2	Mean	1.27e+5	-2.43e+3	-1.38e+1	-9.97e+2	-1.40e+4	-7.60e+4	7.58e+1	4.03e+2	-3.20e+6
	Std	2.86e+3	1.95e+2	6.33e-1	1.84e+2	9.30e+3	2.09e+4	1.93e+0	5.56e+0	2.58e+6
JADE	Mean	3.28e+4	-4.06e+3	-8.79e+0	-7.07e+1	**8.70e+1**	-3.23e+3	1.71e+1	2.87e+2	-3.86e+4
	Std	5.89e+3	1.53e+2	5.16e-1	1.06e+1	**2.91e+0**	1.28e+3	8.39e-1	1.74e+1	1.31e+4
3SOME	Mean	1.19e+5	-3.68e+3	-1.81e+1	-2.70e+3	-1.73e+5	-8.77e+2	3.92e+1	3.77e+2	-6.19e+2
	Std	2.82e+3	1.32e+2	2.03e-1	2.58e+2	1.72e+4	4.21e+2	4.92e+0	4.20e+0	7.49e+1
PMS	Mean	6.38e+4	-6.33e+3	-2.03e+1	-6.30e+3	-3.39e+5	-5.67e+5	**8.87e+1**	3.26e+2	-1.08e+8
	Std	5.90e+4	3.33e+3	1.05e+0	7.19e+3	3.86e+5	5.29e+5	**7.03e+1**	1.18e+2	1.19e+8
BBO	Mean	9.60e+4	-5.22e+3	-1.97e+1	-5.83e+3	-3.12e+5	-4.84e+5	5.10e+1	3.17e+2	-5.83e+7
	Std	4.49e+3	2.59e+2	1.82e-1	4.18e+2	2.56e+4	3.08e+4	4.05e+0	6.93e+0	9.63e+6
ODE	Mean	6.70e+4	-5.75e+3	-1.91e+1	-4.14e+3	-2.13e+5	-3.67e+5	1.74e+1	2.68e+2	-2.74e+7
	Std	7.11e+3	1.68e+2	1.48e-1	1.32e+2	7.92e+3	8.54e+3	1.12e+0	3.60e+0	1.87e+6
CMAES	Mean	**4.44e+5**	-4.70e+3	-2.033e+1	-2.96e+10	6.91e+1	-7.67e+2	7.58e+4	1.25e+2	-2.64e+7
	Std	**3.24e+4**	9.09e+0	1.95e+1	1.84e+10	2.69e+0	1.63e+1	3.67e+3	6.14e+0	1.35e+6
D=1000										
		f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
FMOA	Mean	3.26e+4	**-6.63e+1**	**-9.91e-2**	-5.40e+0	**2.65e+1**	**-3.42e+1**	2.33e+1	**9.08e+2**	**-4.56e+0**
	Std	4.04e+4	**4.69e+2**	**6.37e-1**	4.33e+1	**1.33e+1**	**1.28e+2**	2.63e+1	**1.68e+2**	**5.59e+1**
MASW	Mean	2.06e+5	-5.96e+3	-1.78e+1	**1.95e-1**	-3.74e+3	-5.59e+5	3.01e+1	7.91e+2	-6.07e+2
	Std	1.07e+4	4.54e+2	2.63e-1	**1.17e-1**	7.06e+3	4.73e+4	2.20e+0	1.63e+1	1.47e+2
MASSW	Mean	2.17e+5	-2.11e+3	-1.42e+1	-6.51e-1	-3.05e+1	-4.26e+4	1.90e+1	8.58e+2	-1.83e+5
	Std	1.78e+4	8.59e+2	5.45e-1	1.58e+0	2.90e+2	9.69e+3	4.43e+0	2.98e+1	4.94e+4
CCPSO2	Mean	1.84e+5	-7.74e+3	-1.72e+1	-5.14e+3	-2.24e+5	-4.72e+5	8.39e+1	7.00e+2	-5.90e+7
	Std	3.99e+3	4.25e+2	4.63e-1	7.15e+2	4.60e+4	8.13e+4	2.26e+0	1.07e+1	1.89e+7
JADE	Mean	4.54e+4	-6.67e+3	-1.15e+1	-5.90e+2	-3.83e+3	-7.02e+4	1.95e+1	5.52e+2	-6.24e+6
	Std	7.24e+3	1.30e+3	2.36e-1	5.13e+1	2.05e+3	7.57e+3	9.99e-1	1.19e+1	1.11e+6
3SOME	Mean	2.35e+5	-8.39e+3	-1.94e+1	-1.00e+4	-6.26e+5	-4.79e+3	4.74e+1	7.19e+2	-1.78e+4
	Std	3.89e+3	1.55e+2	9.29e-2	4.92e+2	2.49e+4	1.33e+3	2.41e+0	6.06e+0	1.33e+3
PMS	Mean	1.39e+5	-1.13e+4	-2.01e+1	-1.63e+4	-9.69e+5	-1.23e+6	1.53e+2	5.88e+2	-4.25e+8
	Std	1.18e+5	6.78e+3	1.08e+0	1.46e+4	7.67e+5	1.06e+6	1.15e+2	2.31e+2	4.85e+8
BBO	Mean	1.36e+5	-1.22e+4	-2.02e+1	-1.52e+4	-8.08e+5	-1.22e+6	5.76e+1	5.76e+2	-3.40e+8
	Std	8.16e+3	3.61e+2	1.11e-1	7.63e+2	4.11e+4	6.27e+4	4.28e+0	1.19e+1	3.39e+7
ODE	Mean	1.26e+5	-1.21e+4	-1.97e+1	-1.17e+4	-6.30e+5	-9.96e+5	2.26e+1	5.24e+2	-2.18e+8
	Std	1.13e+4	2.35e+2	1.73e-1	3.16e+2	1.36e+4	1.37e+4	1.81e+0	6.34e+0	8.36e+6
CMAES	Mean	**7.89e+5**	-9.46e+3	-13.44	-1.05e+6	1.18e+1	-1.39e+3	**2.40e+5**	1.97e+2	-5.36e+6
	Std	**9.90e+4**	1.60e+1	1.58e+1	8.34e+4	2.69e+0	1.79e+2	**6.94e+3**	9.4e+0	4.35e+4

Accepted manuscript

ACCEPTED MANUSCRIPT

Table 20: The experimental results for FMOA, MASW, MASSW, CCPSO2, JADE, 3SOME, PMS, BBO and ODE for f_{19}-f_{27}. The best results are bolded.

					D=100					
Algorithm		f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}
FMOA	Mean	-1.77e+3	4.39e+1	-2.89e+4	**-1.45e+03**	-4.13e+01	-7.92e+10	-1.48e+03	-4.21e+01	-1.77e+11
	Std	9.02e+1	4.92e-2	1.74e+3	**1.05e+03**	2.64e-01	7.94e+09	3.99e+01	1.03e-01	7.24e+09
MASW	Mean	-1.23e+3	4.42e+1	**-9.63e+1**	-4.75e+05	-3.78e+01	4.90e+01	-5.89e+02	-3.86e+01	-2.83e+03
	Std	1.36e+2	5.38e-2	**1.65e+0**	1.78e+05	6.55e-01	1.85e-13	4.97e+01	4.12e-01	3.14e+03
MASSW	Mean	-7.95e+2	4.40e+1	-1.05e+2	-1.35e+05	-3.76e+01	4.90e+01	**-5.55e+02**	-3.84e+01	**-2.30e+03**
	Std	9.10e+1	3.88e-2	2.15e+1	7.90e+04	5.33e-01	3.85e-04	**4.31e+01**	3.52e-01	**2.84e+03**
CCPSO2	Mean	**-3.64e+2**	4.45e+1	-4.85e+2	-8.17e+07	-2.06e+01	-5.21e+06	-8.98e+02	-3.24e+01	-1.16e+07
	Std	**5.64e+1**	2.41e-2	6.72e+1	3.34e+07	1.14e+01	2.38e+06	1.47e+02	5.02e+00	8.89e+06
JADE	Mean	-9.11e+2	4.40e+1	-1.09e+2	-2.87e+07	**-3.17e-09**	**4.90e+01**	-9.31e+02	**-1.35e+01**	-6.96e+07
	Std	4.54e+1	3.09e-2	2.47e+1	1.09e+08	**2.74e-09**	**2.15e-14**	2.04e+02	**3.85e+00**	8.58e+07
3SOME	Mean	-1.36e+3	4.41e+1	-1.05e+2	-1.62e+06	-3.08e+01	-1.94e+03	-1.33e+03	-4.01e+01	-1.14e+04
	Std	3.62e+2	1.17e-1	1.50e+1	6.02e+05	6.53e+00	3.97e+02	1.74e+02	1.42e-01	6.83e+03
PMS	Mean	-1.36e+3	**4.45e+1**	-1.48e+2	-9.23e+06	-4.06e+01	4.90e+01	-1.39e+03	-3.99e+01	-4.31e+03
	Std	4.56e+2	**1.28e-1**	5.71e+1	2.26e+07	2.62e-01	1.92e-03	2.70e+02	4.60e-01	4.00e+03
BBO	Mean	-9.32e+2	4.44e+1	-7.85e+3	-2.85e+08	-3.32e+01	-8.67e+09	-9.31e+02	-3.39e+01	-3.26e+10
	Std	7.63e+1	4.08e-2	3.16e+3	5.77e+07	2.66e+00	5.56e+09	7.20e+01	3.02e+00	1.32e+10
ODE	Mean	-1.92e+3	4.40e+1	-3.36e+4	-2.43e+08	-4.17e+01	-5.53e+10	-1.68e+03	-4.23e+01	-1.83e+11
	Std	6.95e+1	2.68e-2	3.95e+3	3.12e+07	2.41e-01	1.28e+10	3.98e+01	1.21e-01	2.05e+10
CMAES	Mean	-2.56e+3	3.50e+1	-2.45e+4	-3.35e+08	-3.56e+01	-3.45e+10	-3.58e+3	-3.55e+01	-2.56e+3
	Std	3.45e+1	3.58e-2	4.42e+3	3.14e+07	2.52e-01	4.35e+10	4.38e+01	2.31e-01	1.45e+3
					D=500					
Algorithm		f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}
FMOA	Mean	-9.59e+3	4.37e+1	-2.31e+5	**-9.33e+04**	-1.25e+02	-4.06e+11	-8.33e+03	-2.11e+02	-9.30e+11
	Std	2.53e+2	4.27e-2	5.41e+3	**6.92e+04**	2.53e-01	1.42e+10	1.06e+02	3.18e-01	1.80e+10
MASW	Mean	-6.72e+3	4.39e+1	-1.22e+3	-4.36e+09	-1.17e+02	**2.45e+02**	-4.55e+03	-2.00e+02	-6.53e+5
	Std	3.07e+2	2.99e-2	6.44e+2	8.27e+08	7.78e-01	**2.29e-02**	3.17e+02	1.40e+00	5.31e+5
MASSW	Mean	-4.79e+3	4.40e+1	**-8.70e+2**	-2.58e+09	-1.18e+02	1.47e+02	-3.87e+03	-1.96e+02	-2.39e+06
	Std	4.62e+2	4.18e-2	**1.66e+2**	5.78e+08	8.02e-01	6.23e+02	2.67e+02	9.14e-01	1.64e+06
CCPSO2	Mean	-5.86e+3	**4.42e+1**	-4.81e+4	-5.61e+10	-1.25e+02	-1.40e+10	-6.36e+03	-2.05e+02	-1.65e+11
	Std	1.72e+2	**2.60e-2**	4.33e+3	1.60e+10	3.24e+00	2.41e+09	3.40e+02	1.22e+00	1.35e+10
JADE	Mean	**-3.99e+3**	4.38e+1	-8.83e+3	-1.91e+10	**-7.60e+01**	-5.26e+07	-6.89e+03	-1.76e+02	-2.21e+11
	Std	**7.59e+2**	2.85e-2	1.11e+3	4.72e+10	**6.79e+00**	3.80e+07	2.36e+02	5.68e+00	2.49e+10
3SOME	Mean	-8.85e+3	4.41e+1	-1.11e+3	-1.27e+10	-1.22e+02	-5.31e+09	-7.86e+03	-2.01e+02	-1.54e+11
	Std	4.19e+2	4.96e-2	1.47e+2	9.89e+08	3.18e+00	2.82e+09	3.98e+02	4.02e-01	2.78e+10
PMS	Mean	-1.07e+4	4.40e+1	-6.34e+5	-1.19e+12	-1.25e+02	-1.12e+12	-9.77e+03	-2.07e+02	-2.64e+12
	Std	6.67e+3	5.21e-1	7.15e+5	2.28e+12	4.07e+00	1.19e+12	2.61e+03	8.21e+00	2.69e+12
BBO	Mean	-8.72e+3	4.41e+1	-2.31e+5	-1.48e+11	-1.21e+02	-2.87e+11	-7.61e+03	-2.09e+02	-9.50e+11
	Std	4.52e+2	3.35e-2	3.33e+4	2.16e+10	1.78e+00	7.38e+10	3.38e+02	3.83e+00	1.50e+11
ODE	Mean	-1.08e+4	4.39e+1	-4.35e+5	-1.74e+11	-1.27e+02	-6.72e+11	-9.84e+03	-2.13e+02	-1.87e+12
	Std	1.69e+2	2.40e-2	1.84e+4	1.82e+10	2.10e-01	4.57e+10	1.38e+02	2.09e-01	8.99e+10
CMAES	Mean	-2.15e+4	4.39e+1	-4.35e+5	-1.74e+11	-1.27e+02	-6.72e+11	**-9.84e+2**	**-1.13e+2**	**-1.47e+5**
	Std	1.69e+2	2.40e-2	1.84e+4	1.82e+10	2.10e-01	4.57e+10	**1.38e+2**	**2.09e-1**	**8.99e+4**
					D=1000					
Algorithm		f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}	f_{26}	f_{27}
FMOA	Mean	-1.95e+4	4.37e+1	-4.87e+5	**-3.37e+10**	-2.29e+02	-9.85e+11	-1.72e+04	-4.23e+02	-1.95e+12
	Std	3.99e+2	3.88e-2	8.71e+3	**2.38e+11**	2.91e-01	1.69e+10	1.71e+02	3.61e-01	2.21e+10
MASW	Mean	-1.44e+4	4.38e+1	-6.95e+3	-8.42e+10	-2.17e+02	**-1.51e+03**	-1.09e+04	-4.08e+02	-9.45e+06
	Std	6.79e+2	2.89e-2	3.19e+3	1.74e+10	1.64e+00	**8.98e+03**	6.88e+02	2.81e+00	1.09e+07
MASSW	Mean	-1.00e+4	4.39e+1	**-4.17e+3**	-4.60e+10	-2.17e+02	-5.47e+05	-8.49e+03	-3.96e+02	-1.87e+08
	Std	8.14e+2	5.21e-2	**4.67e+2**	1.25e+10	1.74e+00	1.84e+06	3.26e+02	2.06e+00	4.00e+08
CCPSO2	Mean	-1.51e+4	**4.41e+1**	-2.74e+5	-8.92e+11	-2.31e+02	-1.97e+11	-1.45e+04	-4.19e+02	-1.20e+12
	Std	2.21e+2	**2.57e-2**	1.26e+4	2.81e+11	2.50e+00	1.98e+10	7.94e+02	7.47e-01	5.53e+10
JADE	Mean	**-9.93e+3**	4.38e+1	-7.33e+4	-4.30e+11	**-1.93e+02**	-1.60e+10	-1.53e+04	-3.94e+02	-1.03e+12
	Std	**4.60e+2**	2.20e-2	6.22e+3	1.13e+12	**5.11e+00**	3.84e+09	3.48e+02	5.70e+00	7.70e+10
3SOME	Mean	-1.83e+4	4.40e+1	-6.02e+3	-2.77e+11	-2.30e+02	-1.29e+11	-1.63e+04	-4.06e+02	-1.04e+12
	Std	5.55e+2	4.72e-2	5.00e+2	2.34e+10	2.55e+00	2.50e+10	4.00e+02	4.91e-01	1.17e+11
PMS	Mean	-1.93e+4	4.40e+1	-1.10e+6	-4.63e+13	-2.30e+02	-2.77e+12	-2.10e+04	-4.16e+02	-5.75e+12
	Std	1.05e+4	5.03e-1	1.18e+6	8.62e+13	7.87e+00	2.59e+12	5.21e+03	1.61e+01	5.16e+12
BBO	Mean	-1.96e+4	4.40e+1	-6.73e+5	-2.18e+12	-2.27e+02	-9.99e+11	-1.72e+04	-4.25e+02	-2.79e+12
	Std	6.72e+2	2.98e-2	7.74e+4	4.26e+11	1.57e+00	2.00e+11	5.61e+02	2.79e+00	3.25e+11
ODE	Mean	-2.22e+4	4.38e+1	-9.87e+5	-2.85e+12	-2.33e+02	-1.92e+12	-2.06e+04	-4.28e+02	-4.54e+12
	Std	2.38e+2	2.46e-2	2.46e+4	3.94e+11	3.28e-01	8.30e+10	2.19e+02	2.14e-01	1.37e+11
CMAES	Mean	-3.07e+8	-1.98e+7	-4.67e+5	-6.53e+10	-2.17e+2	-2.44e+11	**-6.36e+3**	**-2.16e+2**	**-4.09e+5**
	Std	3.39e+7	4.10e+4	-1.47e+5	3.10e+11	2.78e+00	8.75e+10	**2.45e+2**	**8.83e-1**	**3.10e+5**

Accepted manuscript

ACCEPTED MANUSCRIPT

Table 21: The Wilcoxon Signed Rank Test for the proposed algorithm.

	$D = 100$			$D = 500$			$D = 1000$		
Algorithm	R^+	R^-	$p-value$	R^+	R^-	$p-value$	R^+	R^-	$p-value$
GA	338	40	3.43e-04	344	34	1.96e-04	310	41	6.35e-04
PSO	247	131	0.163	339	39	3.13e-04	318	33	3.10e-04
DE	206	172	0.683	249	129	0.14	276	75	0.01
ES	198	182	0.82	232	146	0.30	220	131	0.25
FES	197	181	0.84	214	164	0.54	203	148	0.48
EP	326	52	9.96e-04	358	20	4.89e-05	322	29	1.98e-4
FEP	245	133	0.17	334	44	4.94e-04	335	16	5.10e-05
MASW	164	214	0.54	172	206	0.68	160	191	0.69
MASSW	149	229	0.33	151	227	0.36	159	192	0.67
CCPSO2	157	221	0.44	209	169	0.63	212	139	0.35
JADE	143	235	0.26	193	185	0.92	196	155	0.60
3SOME	178	200	0.78	221	157	0.44	203	148	0.48
PMS	230	148	0.32	336	42	4.12e-04	306	45	9.18e-4
BBO	213	165	0.56	258	120	0.09	297.5	53.5	1.9e-4
ODE	310	68	3.0e-3	342	36	2.36e-04	319	32	2.67e-4
CMAES	189	189	0.787	259	119	0.09	180	171	0.90

Table 22: The Friedman Test for the benchmark problems where $D = 100, 500$ and 1000.

Unimodal Problems																	
Algorithm	FMOA	GA	PSO	DE	ES	FES	EP	FEP	MASW	MASSW	CCPSO2	JADE	3SOME	PMS	BBO	ODE	CMAES
$D = 100$	108	67	59	91	70	74	11	29	121	122	90	132	113	72	46	52	120
$D = 500$	124	67	94	87	55	87	20	37	118	138	93	128	101	21	53	55	99
$D = 1000$	97	72	88	90	61	95	26	36	121	138	89	120	109	35	53	54	93
Multimodal Problems																	
Algorithm	FMOA	GA	PSO	DE	ES	FES	EP	FEP	MASW	MASSW	CCPSO2	JADE	3SOME	PMS	BBO	ODE	CMAES
$D = 100$	170	142	100	151	207	198	50	141	209	228	240	209	190	160	145	78	136
$D = 500$	191	139	135	141	166	209	51	56	237	256	227	212	214	103	150	105	162
$D = 1000$	206	110	141	123	158	199	45	42	227	238	192	209	198	111	124	92	186
All the Problems																	
Algorithm	FMOA	GA	PSO	DE	ES	FES	EP	FEP	MASW	MASSW	CCPSO2	JADE	3SOME	PMS	BBO	ODE	CMAES
$D = 100$	278	209	159	242	277	272	61	170	330	350	330	341	303	232	191	130	256
$D = 500$	315	206	229	228	221	296	71	93	355	394	320	340	315	124	203	160	261
$D = 1000$	303	182	229	213	219	294	71	78	348	376	281	329	307	146	177	146	279

Accepted manuscript

ACCEPTED MANUSCRIPT

The main contribution of this paper is the new parameter setting algorithm that dynamically configures the parameters. The algorithm tries to learn the properties of the fitness landscape and move the parameters toward desired optimal value. As a result, the parameter tuning process of the optimization algorithms is omitted. Despite this advantage, the proposed algorithm may not be as good as the some state-of-the-art algorithms. Therefore, in the future work, we will concentrate on designing some powerful components to improve the algorithm's performance.

We analysed the parameter adaptation strategy and compared it with the JADE's scheme. We have shown that the proposed adaptation strategy promisingly find the best parameter value during the optimization process and regardless of the initial value for each parameter, the algorithm finds optimal value for each parameter. We also challenged the parameter adaptation strategy against two parameter setting techniques on several unimodal and multi-modal problems, and the results showed that the proposed adaptation strategy outperforms the other two techniques in most problems, especially when the problem size is greater than 500.

Furthermore, we studied the performance of the OBL part of the algorithm by making an comparison based on RLDs between the original version of MOA and the MOA with the OBL mechanism, and the results showed that the OBL helps the algorithm improve its convergence rate.

Finally, we compared the proposed algorithm with nine state-of-the-art optimization algorithms and seven popular population-based algorithm on 27 benchmark functions. The results indicated that FMOA outperforms all the traditional population-based algorithms on most problems and its results are comparative to those found by other state-of-the-art algorithms, particularly when the problem size is large (greater than 500).

In future, we are planning to apply the proposed algorithm on some real-world problems. For example, for solving a hypercube problem [59], which is a NP-hard problem, the proposed algorithm could be a very good choice. This is because the problem is prohibitively time-consuming and removing the expensive parameter configuration process will help us focus on solving the problems without excessively spending computational effort on the parameter configuration process. Using the proposed method for machine learning applications like training neural networks, optimizing the parameters of learning algorithms, etc. is another line of research.

References

[1] H. Tizhoosh, Opposition-based learning: A new scheme for machine intelligence, in: Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on, Vol. 1, 2005, pp. 695 –701.

[2] S. Rahnamayan, G. G. Wang, M. Ventresca, An intuitive distance-based explanation of opposition-based sampling, Applied Soft Computing 12 (9) (2012) 2828 – 2839.

[3] N. Dong, C.-H. Wu, W.-H. Ip, Z.-Q. Chen, C.-Y. Chan, K.-L. Yung, An opposition-based chaotic ga/pso hybrid algorithm and its application in circle detection, Computers & Mathematics with Applications 64 (6) (2012) 1886 – 1902.

[4] W. feng Gao, S. yang Liu, L. ling Huang, Particle swarm optimization with chaotic opposition-based population initialization and stochastic search technique, Communications in Nonlinear Science and Numerical Simulation 17 (11) (2012) 4316 – 4327.

[5] S. Rahnamayan, H. Tizhoosh, M. Salama, Opposition-based differential evolution, Evolutionary Computation, IEEE Transactions on 12 (1) (2008) 64–79.

[6] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, M. Ventresca, Enhancing particle swarm optimization using generalized opposition-based learning, Information Sciences 181 (20) (2011) 4699 – 4714.

[7] H. Wang, S. Rahnamayan, Z. Wu, Parallel differential evolution with self-adapting control parameters and generalized opposition-based learning for solving high-dimensional optimization problems, Journal of Parallel and Distributed Computing 73 (1) (2013) 62 – 73, metaheuristics on GPUs.

[8] S. Rahnamayan, H. R. Tizhoosh, M. M. Salama, Opposition versus randomness in soft computing techniques, Applied Soft Computing 8 (2) (2008) 906 – 918.

[9] M. Ahandani, H. Alavi-Rad, Opposition-based learning in the shuffled differential evolution algorithm, Soft Computing 16 (2012) 1303–1337.

[10] M. Tayarani-N, M. Akbarzadeh-T, Magnetic optimization algorithms a new synthesis, in: Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on, 2008, pp. 2659 –2664.

[11] M.-H. Tayarani-N., M.-R. Akbarzadeh-T., Magnetic-inspired optimization algorithms: Operators and structures, Swarm and Evolutionary Computation 19 (0) (2014) 82 – 101. doi:http://dx.doi.org/10.1016/j.swevo.2014.06.004.
URL http://www.sciencedirect.com/science/article/pii/S2210650214000509

ACCEPTED MANUSCRIPT

[12] M. Ismail, M. Zakaria, A. Abidin, J. Juliani, J. Lit, Magnetic optimization algorithm approach for travelling salesman problem, in: World Academy of Science, Engineering and Technology, 2012.
[13] S. Mirjalili, A. Sadiq, Magnetic optimization algorithm for training multi layer perceptron, in: Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on, 2011, pp. 42 –46.
[14] K.-H. Han, J.-H. Kim, Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, Evolutionary Computation, IEEE Transactions on 6 (6) (2002) 580–593.
[15] M. Torshizi, M. Tayarani-N., Functional sized population magnetic optimization algorithm, in: F. Peper, H. Umeo, N. Matsui, T. Isokawa (Eds.), Natural Computing, Vol. 2 of Proceedings in Information and Communications Technology, Springer Japan, 2010, pp. 316–324.
[16] M. Tayarani, M. Akbarzadeh.T, A cellular structure and diversity preserving operator in quantum evolutionary algorithms, in: Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on, 2008, pp. 2665–2670.
[17] K. Burnham, D. Anderson, Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach, Springer, 2002.
[18] M. Birattari, T. Stützle, L. Paquete, K. Varrentrapp, A racing algorithm for configuring metaheuristics, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '02, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002, pp. 11–18.
[19] M. Birattari, Z. Yuan, P. Balaprakash, T. Sttzle, F-race and iterated f-race: An overview, in: T. Bartz-Beielstein, M. Chiarandini, L. Paquete, M. Preuss (Eds.), Experimental Methods for the Analysis of Optimization Algorithms, Springer Berlin Heidelberg, 2010, pp. 311–336.
[20] M. Lpez-Ibez, T. Sttzle, Automatic configuration of multi-objective aco algorithms, in: M. Dorigo, M. Birattari, G. Caro, R. Doursat, A. Engelbrecht, D. Floreano, L. Gambardella, R. Gro, E. ahin, H. Sayama, T. Sttzle (Eds.), Swarm Intelligence, Vol. 6234 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2010, pp. 95–106.
[21] M. Montes de Oca, D. Aydn, T. Sttzle, An incremental particle swarm for large-scale continuous optimization problems: an example of tuning-in-the-loop (re)design of optimization algorithms, Soft Computing 15 (11) (2011) 2233–2255.
[22] J. Zhang, A. Sanderson, Jade: Adaptive differential evolution with optional external archive, Evolutionary Computation, IEEE Transactions on 13 (5) (2009) 945–958.
[23] P. J. Angeline, Adaptive and self-adaptive evolutionary computations, in: in Computational Intelligence: A Dynamic Systems Perspective, 1995, p. 152163.
[24] A. E. Eiben, R. Hinterding, Z. Michalewicz, Parameter control in evolutionary algorithms, IEEE Transactions on Evolutionary Computation 3 (2) (1999) 124141.
[25] H. A. Abbass, The self-adaptive pareto differential evolution algorithm, in: Evolutionary Computation, Vol. 1, 2002, p. 831836.
[26] A. K. Qin, P. Suganthan, Self-adaptive differential evolution algorithm for numerical optimization, in: Evolutionary Computation, 2005. The 2005 IEEE Congress on, Vol. 2, 2005, pp. 1785–1791.
[27] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, Evolutionary Computation, IEEE Transactions on 10 (6) (2006) 646–657.
[28] N. ShahidI, H. Esmaeilzadeh, M. Abdollahi, E. Ebrahimi, C. Lucas, Self-adaptive memetic algorithm: an adaptive conjugate gradient approach, in: Cybernetics and Intelligent Systems, 2004 IEEE Conference on, Vol. 1, 2004, pp. 6–11.
[29] J. H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, MIT Press, Cambridge, MA, USA, 1992.
[30] M. AlRashidi, M. El-Hawary, A survey of particle swarm optimization applications in electric power systems, Evolutionary Computation, IEEE Transactions on 13 (4) (2009) 913 –918.
[31] R. Storn, K. Price, Differential evolution a simple and efficient heuristic for global optimization over continuous spaces, J. of Global Optimization 11 (4) (1997) 341–359.
[32] C. Barnard, R. Sibly, Producers and scroungers: A general model and its application to captive flocks of house sparrows, Animal Behaviour 29 (2) (1981) 543 – 550.
[33] X. Yao, Y. Liu, Fast evolution strategies, in: Proceedings of the 6th International Conference on Evolutionary Programming VI, EP '97, Springer-Verlag, London, UK, UK, 1997, pp. 151–162.
[34] L. J. Fogel, Intelligence through simulated evolution: forty years of evolutionary programming, John Wiley & Sons, Inc., New York, NY, USA, 1999.
[35] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, Evolutionary Computation, IEEE Transactions on 3 (2) (Jul) 82–102.
[36] D. Molina, M. Lozano, A. Snchez, F. Herrera, Memetic algorithms based on local search chains for large scale continuous optimisation problems: Ma-ssw-chains, Soft Computing 15 (2011) 2201–2220.
[37] X. Li, X. Yao, Cooperatively coevolving particle swarms for large scale optimization, Evolutionary Computation, IEEE Transactions on 16 (2) (2012) 210–224.
[38] G. Iacca, F. Neri, E. Mininno, Y.-S. Ong, M.-H. Lim, Ockhams razor in memetic computing: Three stage optimal memetic exploration, Information Sciences 188 (0) (2012) 17 – 43.
[39] F. Caraffini, F. Neri, G. Iacca, A. Mol, Parallel memetic structures, Information Sciences 227 (0) (2013) 60 – 82.
[40] D. Simon, Biogeography-based optimization, Evolutionary Computation, IEEE Transactions on 12 (6) (2008) 702–713.
[41] S. Rahnamayan, H. Tizhoosh, M. Salama, Opposition-based differential evolution, Evolutionary Computation, IEEE Transactions on 12 (1) (2008) 64–79.
[42] N. Hansen, S. D. Müller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with

Accepted manuscript

ACCEPTED MANUSCRIPT

covariance matrix adaptation (cma-es), Evol. Comput. 11 (1) (2003) 1–18.

[43] W. Zhong, J. Liu, M. Xue, L. Jiao, A multiagent genetic algorithm for global numerical optimization, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 34 (2) (2004) 1128–1141.

[44] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, S. Tiwari, Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization, Tech. rep., Nanyang Technological University, Singapore (2005).

[45] X. Li, K. Tang, M. N. Omidvar, Z. Yang, K. Qin, Benchmark functions for the cec2013 special session and competition on large-scale global optimization (2013).

[46] A. R. Khorsand, M.-R. Akbarzadeh-T, Quantum gate optimization in a meta-level genetic quantum algorithm, in: Systems, Man and Cybernetics, 2005 IEEE International Conference on, Vol. 4, 2005, pp. 3055–3062 Vol. 4.

[47] V. Koumousis, C. Katsaras, A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance, Evolutionary Computation, IEEE Transactions on 10 (1) (2006) 19–28.

[48] H. Hoos, T. Sttzle, Stochastic Local Search: Foundations & Applications, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[49] M. A. M. de Oca, T. Stützle, M. Birattari, M. Dorigo, A comparison of particle swarm optimization algorithms based on run-length distributions, in: Proceedings of the 5th international conference on Ant Colony Optimization and Swarm Intelligence, ANTS'06, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 1–12.

[50] K. Price, R. M. Storn, J. A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[51] A. Qin, V. Huang, P. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, Evolutionary Computation, IEEE Transactions on 13 (2) (2009) 398–417.

[52] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems, Evolutionary Computation, IEEE Transactions on 10 (6) (2006) 646–657.

[53] J. Ronkkonen, S. Kukkonen, K. Price, Real-parameter optimization with differential evolution, in: Evolutionary Computation, 2005. The 2005 IEEE Congress on, Vol. 1, 2005, pp. 506–513 Vol.1.

[54] N. Hansen, The cma evolution strategy: A comparing review, in: J. Lozano, P. Larraaga, I. Inza, E. Bengoetxea (Eds.), Towards a New Evolutionary Computation, Vol. 192 of Studies in Fuzziness and Soft Computing, Springer Berlin Heidelberg, 2006, pp. 75–102.

[55] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evol. Comput. 9 (2) (2001) 159–195.

[56] F. Wilcoxon, Individual comparisons by rankings methods, Biometrics 1 (80-83).

[57] D. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, CRC Press, Boca Raton, 1997.

[58] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the cec'2005 special session on real parameter optimization, Journal of Heuristics 15 (6) (2009) 617–644.

[59] V. Joseph, Y. Hung, Orthogonal-maximin latin hypercube designs, Statistica Sinica 18 (2008) 171–186.

Accepted manuscript