Improving the local search capability of Effective Butterfly Optimizer using Covariance Matrix Adapted Retreat phase

Abhishek Kumar
Department of Electrical Engineering
Indian Institute of Technology (B.H.U)
Varanasi, India-221005
Email: abhishek.kumar.eee13@iitbhu.ac.in

Rakesh Kumar Misra
Department of Electrical Engineering
Indian Institute of Technology (B.H.U)
Varanasi, India-221005
Email: rkmishra.eee@iitbhu.ac.in

Devender Singh
Department of Electrical Engineering
Indian Institute of Technology (B.H.U)
Varanasi, India-221005
Email: dsingh.eee@iitbhu.ac.in

***Abstract*—Effective Butterfly Optimizer(EBO) is a self-adaptive Butterfly Optimizer which incorporates a crossover operator in Perching and Patrolling to increase the diversity of the population. This paper proposes a new retreat phase called Covariance Matrix Adapted Retreat Phase (CMAR), which uses covariance matrix to generate a new solution and thus improves the local search capability of EBO. This version of EBO is called EBOwithCMAR. We evaluated the performance of EBOwithCMAR on CEC-2017 benchmark problems and compared with the results of winners of a special session of CEC-2016 for bound-constrained problems. The experimental results show that EBOwithCMAR is competitive with the compared algorithms.**

I. Introduction

Numerous algorithms have been introduced which have got their foundation knowledge from nature, human-being, culture, society, politics, etc. The word "meta-heuristics" is used for such techniques that coalesce rules and uncertainty while mimicking specific natural phenomena. This research ground has attracted different categories of research from several areas of science. The significance is by far in utilizing the active meta-heuristics for solving real-world optimization problems in numerous areas such as engineering, industry, business, etc.

Among meta-heuristics, most of the algorithms, motivated by swarming behavior of organisms and Darwin's theory of evolution, are called Swarm Intelligence (SIs) and Evolution based Algorithms(EAs) respectively. Some popular SIs are: Ant Colony Optimization (ACO) [1], Particle Swarm Optimization (PSO) [2], etc. Dorigo proposed ACO in 1991, and it is encouraged by the skill of real ants to locate the shortest path between the nest and food source [1]. On the other hand, PSO, developed by the Kennedy and Eberhart in 1995 [2], is motivated by the flocking behavior of birds and fishes. Behaviors of butterflies have also inspired some researchers to develop an optimization technique. Butterfly Particle Swarm Optimization (BF-PSO) [3], Monarch Butterfly Optimization(MBO) [4] and Butterfly Optimizer [5] are three newly developed optimization techniques inspired by the intelligent behaviors of a butterfly.

Success History Based Adaption (SHBA) is a new method for parameter adaption based on a historic remembrance of successful parameter settings in Differential Evolution(DE) [6], [7], [8]. In DE, SHBA uses, memory matrices, M_{CR} and M_F which supplies a set of crossover probability (CR), and scaling factor (F) values for generation of the new parameter by directly sampling the parameter space near this pair [6], [7], [8].

The parameter "population size" plays an important role in guiding the rate of convergence. Small population sizes give faster convergence, but also increases the possibility of getting stuck in a local optimum. On the other hand, large population sizes can support broader exploration of the search space, but the rate of convergence becomes slower. The suitable population size depends on number of factors the properties of problems and parameters [8]. In recent years population resizing methods on the basis of simple, deterministic rules are proposed in [9], [10], [11], [12], [13]. These methods have been established as highly effective ways to improve the algorithms performance. Most of the deterministic methods either monotonically increase or decrease the population size. In LSHADE algorithm [8] proposed a linear population size reduction (LPSR) is proposed which is a simple deterministic population resizing method which continuously shrinks the population size in accordance with a linear function [8].

Hybridization, in framework of optimization technique, mostly denotes a procedure of merging the best features of two or more algorithm together to form a new algorithm that is presumed to outperform its ancestors [14]. Hybridization of algorithms is receiving popularity due to their abilities to handle the several real world problems [15], [16].

In this paper, a simple yet powerful hybrid algorithm EBOwithCMAR is proposed that combines the characters of global optimizer and a local optimizer. In EBOwithCMAR, the search capability of EBO is improved by using the SHBA and LPSR to automatically adapt the various parameters during optimization process. This paper also proposes a Covariance Matrix Adapted Retreat (CMAR) phase to improve the local search capability of algorithm.

The proposed algorithm is benchmarked on CEC-2017's competition on bound constrained problems, with 10, 30, 50,

978-1-5090-4601-0/17/$31.00 ©2017 IEEE

and 100 dimensions [17]. The experimental results validate its capability for providing good quality solutions which are better than results obtained using other state-of-the-art algorithms.

The remaining part of this paper is organized as follows: a brief description of EBO and CMAR is provided in section-II; the proposed algorithm is discussed in Section-III; finally the experimental results and conclusions are provided in Section IV and V respectively.

II. Effective Butterfly Optimizer(EBO) and its Operators

In this section, EBO and CMAR, discussed in this study are briefly described.

A. EBO

EBO is a dual population-based bio-inspired optimization technique, which is inspired by the mate-locating behaviors of male butterflies [5]. EBO starts with random initail populations ($X_1 = \{\bar{x}1_1, \bar{x}1_2, ...\bar{x}1_{PS_1}\}, X_2 = \{\bar{x}2_1, \bar{x}2_2, ...\bar{x}2_{PS_2}\}$), where PS_1 and PS_2 are the population size of the primary population and secondary population respectively. Elements of vectors, $\bar{x}2_z$ and $\bar{x}2_z$ should be within the search domain. Then, a new population, $V = \{\bar{v}_1, \bar{v}_2, ...\bar{v}_{PS_1}\}$ is generated after "criss-cross" or "towards-best" modification. Consequently, every $\bar{x}1_z$ is crossed with its corresponding $\bar{v}_z$ to generate a sample vector $\bar{s}_z$. A pair-wise comparison between $\bar{x}1_z$ and $\bar{s}_z$, is made for all such pairs based on the fitness values. A primary population set consist of the samples which are better in the pairwise comparison, whereas, the remaining vectors are put in the secondary population set. Below is a brief explanation of each step and operator mentioned above.

1) Initialization: A D-dimensional vector is used to represent each individual in both the populations. Initially, each variable of the vectors is generated randomly within its boundaries as shown below.

$$\bar{x}1^0_{z,j} = x_{LB,j} + rand_j(0,1) * (x_{UB,j} - x_{LB,j}) \forall j = 1, 2, ...D$$

$$\bar{x}2^0_{z,j} = x_{LB,j} + rand_j(0,1) * (x_{UB,j} - x_{LB,j}) \forall j = 1, 2, ...D$$

where $rand_j(0,1)$ is a uniform random number within 0 and 1, $x_{LB,j}$ and $x_{UB,j}$ are the lower and upper bound of j^{th} decision variable of problem respectively.

2) Modification: In this step, new vectors are generated that are perturbations of the current ones towards crisscross-neighbor or best-neighbor.There are two types of modifications used in EBO; during perching, crisscross modification is used, whereas during patrolling, towards-best modification is used. In a criss-cross modification, $\bar{v}_z$ is generated by adding a scaled difference between two random neighbors to a criss-cross neighbor (equation 1). In towards-best modification, $\bar{v}_z$ is generated by adding the scaled difference between criss-cross neighbor and a random neighbor to the best neighbor (equation 2).

$$\bar{v}_z = \bar{x}1_{cc_z} + F * (\bar{x}1_{r1_z} - (X_1 \cup X_2)_{r2_z}) \tag{1}$$

$$\bar{v}_z = \bar{x}1_{best_z} + F * (\bar{x}1_{cc_z} - (X_1 \cup X_2)_{r2_z}) \tag{2}$$

where ($\bar{x}1_{cc_z}, \bar{x}1_{r1_z}, and (X_1 \cup X_2)_{r2_z}$) are distinct individual vectors. $\bar{x}1_{best_z}$ is a best-neighbor of z^{th} vector. F is a positive real number that controls the rate at which the population evolves. $X_1 \cup X_2$ is the union of the both the populations. $\bar{x}1_{cc_z}$ is called criss-cross neighbor of z^{th} vector which is calculated by equation 3.

$$\{cc_1, cc_2, ...cc_{PS_1}\} = randperm(1, PS_1) \tag{3}$$

where $randperm(1, PS_1)$ is a random permutation vector having elements between 1 and PS_1. The selection of modification operator for particular individual is determined according to probabilities, $prob_{perch}$ and $prob_{pat}$.

3) Crossover: Two well-known crossover schemes, binomial and exponential, exist in the literature for real-coded optimization techniques [18]. The binomial crossover, is operated on every $j \epsilon [1, D]$ separately with a tuned crossover probability, CR [18]. That is, for each j , a uniform random number is produced. If value of random number is greater than CR, the value of $\bar{s}_{z,j}$ will be taken corresponding value of $\bar{x}_{z,j}$, otherwise it will be equal to $\bar{v}_{z,j}$.

$$\bar{s}_{z,j} = \begin{cases} \bar{v}_{z,j}, & \text{if } (rand_j(0,1) \leq CR \text{ or } j = j_{rand}) \\ \bar{x}_{z,j}, & \text{otherwise.} \end{cases} \tag{4}$$

Where j_{rand} is a randomly chosen integer index which guarantees that $\bar{s}_{z,j}$ acquires at least one component from $\bar{v}_{z,j}$.

In binomial crossover, the value of CR is same for all the j. In EBO, a modified version of Binomial crossover is used where CR is not similar for all the j. Crossover probability for particular j, cr_j is calculated by equations (5) and (6).

$$n_j = rem(D + j - j_{rand}, \frac{D}{2}) \tag{5}$$

where $rem(a, b)$ is a remainder function which gives the remainder of division of a and b.

$$cr_j = CR * e^{-\frac{T}{D} * n_j} \tag{6}$$

where T is a real number within range of $[0, 0.5]$. If $T = 0$, then this crossover method is reduced to binomial crossover.

4) Selection: In EBO, one-to-one individual selection at iteration t is used, where the objective function values $f_{\bar{s}_z}$ is compared against $f_{\bar{x}_z}$, and the better one is taken as member of new primary population ($X1$) in the next iteration $t+1$. The worse one $\bar{x}_z$, is selected for the X2 and it replaces the randomly selected vector in X2 in the next iteration $t+1$.

B. Covariance Matrix Adapted Retreat

In Retreat phase, new solutions are sampled in search space. To generate new sample solution it is essential to calculate the mean and covariance matrix of the random distribution to be used. The newly sampled individuals are then weighted by their objective function value. Based on these values, a subset of individuals are selected from sampled individuals, which form a new population that is used to calculate the mean and covariance matrix in the next iteration. A brief explanation of each step for generating the sample solution.

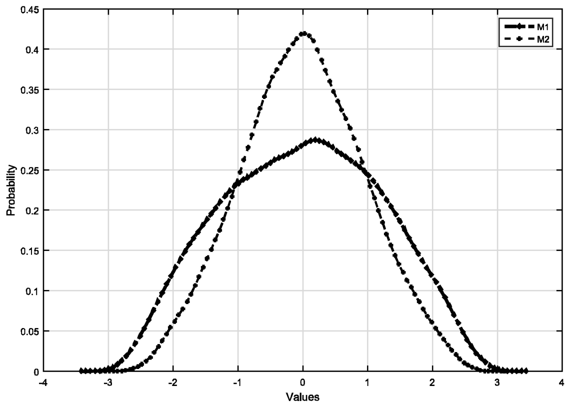


Fig. 1. Probability density plot of Multivariate distributions $M1(0, I))$ and $M2(0, I)$.

1) Multi-variate Random Distribution: Two new multivariate distribution $M1(m, C)$ and $M2(m, C)$ are used in CMAR to sample the new solution with a probability of 0.5. $M1(0, I)$ and $M2(0, I)$ have a unimodal bell-shaped density shown in the figure 1, where the modal value tallies to the mean of the distribution, m. The distributions $M1(m, C)$ and $M2(m, C)$ are individually controlled by its mean m and a symmetric and positive definite covariance matrix C. The distribution M1 and M2 can be represented in following ways.

$$\begin{aligned} M1(m, C) &\sim m + M1(0, C) \sim C^{\frac{1}{1}} M1(0, I) \\ &\sim BDB^T M1(0, I) \sim BDM1(0, I) \end{aligned}$$

similarly,

$$\begin{aligned} M2(m, C) &\sim m + M2(0, C) \sim C^{\frac{1}{1}} M2(0, I) \\ &\sim BDB^T M2(0, I) \sim BDM2(0, I) \end{aligned}$$

where " $\sim$ " denotes similar in distribution. $M1(0, I)$ produces a cubical(isotropic) distribution and $M2(0, I)$ produces a spherical (isotropic) distribution as shown in figure 2.

2) Sampling:: The new solution vectors are produced in CMAR by methods of sampling which use the following equation.

$$\bar{x}3_z^{t+1} \sim \begin{cases} m^t + \sigma^t M1(0, C^t), & \text{if } rand(0,1) \leq 0.5 \\ m^t + \sigma^t M2(0, C^t), & \text{otherwise.} \end{cases} \tag{7}$$

Where σ is a step size that controls the size of distribution

3) Mean Calculation: The new mean m of the population is the weighted average of half of the best solutions from the current samples $(\{\bar{x}3_1, \bar{x}3_2, ...\bar{x}3_{PS_3}\})$ as shown in equation (8)

$$m^t = \sum_{i=1}^{\frac{PS_3}{2}} w_i^t \bar{x}3_i^t \tag{8}$$

here, $\sum_{i=1}^{\frac{PS_3}{2}} w_i = 1$, $w_1 \geq w_2 \geq ....w_{\frac{PS_3}{2}} \geq 0$ weight w_i is calculated by using equation (9)

$$w_z^t = \frac{f(\bar{x}3_z^t)}{\sum_{i=1}^{\frac{PS_3}{2}} f(\bar{x}3_i^t)} \tag{9}$$

4) Covariance Matrix and global step size adaption: Covariance matrix and global step size adaptation are adopted from the CMA-ES [19]. The concept of self-adaptation of the covariance matrix of a multivariate normal distribution is developed in CMA-ES [19]. We directly use the concept of adaptation of covariance matrix and step size control from where rank μ update and cumulation are used to update the covariance matrix. Due to page restriction, explanation of self-adaptation of the covariance matrix and the step size is not discussed here.(For more explanation see [20])

III. Effective Butterfly Optimizer with Covariance Matrix Adapted Retreat Phase(EBOwithCMAR)

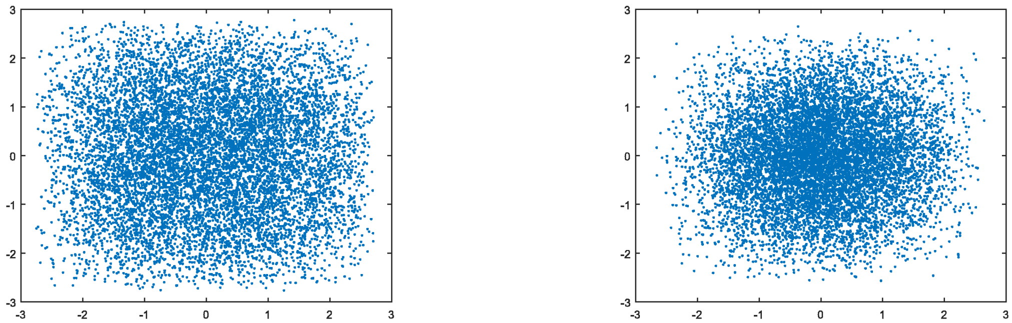
In this section, the proposed algorithm, EBOwithCMAR, and its components are briefly explained.

A. Framework of the EBOwithCMAR

The basic framework of EBOwithCMAR is inspired from the basic framework of UMOEAs-II, proposed in [21]. The basic procedure of the EBOwithCMAR is explained in Algorithm 1. An initial population of solutions of size PS is randomly generated within the bound of search space. Then, PS is split into three sub-populations of sizes, PS_1, PS_2, and PS_3. The solutions in the first and second population are used in EBO, while solutions in the last sub-population are used in CMAR. The probabilities $prob_1$ and $prob_2$ decides which specific algorithm out of EBO and CMAR is to applied at a particular iteration. Probabilities $prob_1$ and $prob_2$ are dynamically fluctuating over the optimization process [21]. In this algorithm prefixed number of iterations are called here as cycle, CS. At the beginning of every cycle, both $prob_1$, and $prob_2$ are set to 1. This amounts to running both the algorithms in parallel for half the cycle, after which the new values of $prob_1$ and $prob_2$ are calculated. The value of probabilities $prob_1$ and $prob_2$ are based on following criteria [21].

1) The superiority of the best solution found by EBO and CMAR, and
2) The diversity rate of PS_1 and PS_3.

Once the cycle is over, a data sharing system is through, and both the $prob_1$ and $prob_2$ are retuned to 1 [21]. Consequently, both algorithms will enter into next cycle, and the identical procedure is repeated. To improve the exploitation potential of EBOwithCMAR, the sequential quadratic programming (SQP) is employed at the later phases when 75% of optimization process is completed with a dynamic probability, $prob_{ls}$. In case that solution of local search is better than the best solution found so far, then $prob_{ls}$ is reset to default value, otherwise it reduced to very small value. Until a stopping criterion is not met, the algorithm continues [21].



(a) Case M1 (b) Case M2

Fig. 2. Distribution of samples in $M1(0, I)$ and $M2(0, I)$.

The different component of modified EBO is presented in the following sections.

B. Modified EBO

In the previous section, a brief introduction of the procedure of EBO is discussed. In EBOwithCMAR, equations (1) and (2) are modified to increase the diversity of the population, and the self-adaptive mechanism is employed for different parameters of EBO, PS_1, F, CR, and T. As declared above, EBO initializes with PS_1 and PS_2 populations which have taken solutions randomly from the entire population X. In this paper, two new modification equations are used in EBO for patrolling and perching respectively. Improved phases of EBO which is not discussed earlier are described in following sub-sections.

1) Improved Perching and Patrolling: In perching, a new criss-cross modification with modified binomial crossover was used as shown in equation (10). Similarly, in patrolling a new towards-best modification with modified binomial crossover was employed as given in equation (11).

$$s_{z,j} = \begin{cases} x1_{z,j} + F_z(x1_{cc_z,j} - x1_{z,j} + x1_{r1_z,j} - (X_1 \cup X_2)_{r2_z,j}), \\ \text{if } (rand_j(0,1) \le cr_{z,j} \text{ or } j = j_{rand}) \\ x1_{z,j}, \\ \text{otherwise.} \end{cases} \tag{10}$$

$$s_{z,j} = \begin{cases} x1_{z,j} + F_z(x1_{best_z,j} - x1_{z,j} + x1_{cc_z,j} - (X_1 \cup X_2)_{r2_z,j}), \\ \text{if } (rand_j(0,1) \le cr_{z,j} \text{ or } j = j_{rand}) \\ x1_{z,j}, \\ \text{otherwise.} \end{cases} \tag{11}$$

where $(cc_z, r1_z, \text{and } r2_z)$ are distinct integers.

2) Selection of $best_z$: A new way to select $best_z$ individual for z^{th} individual is proposed in this paper. When the size of the dynamic population of X_1, PS_1 is greater than $2D$, then for the z^{th} individual, the best individual, $best_z$, is the best individual among the randomly chosen D individuals. Otherwise if the population size is less then $2D$, $best_z$ is randomly chosen from the best 10% individual.

3) Calculation of $prob_{perch}$ and $prob_{pat}$: Initially, the probabilities, $prob_{perch}$ and $prob_{pat}$ are set to 0.5. To update these probabilities, the improvement rate in objective function values is considered. So, at the end of each iteration, the improvement rate are calculated by using equation (12)[21].

$$I_i^{t+1} = \frac{\sum_{z=1,i}^{PS_1} max(0, f_z^{t+1} - f_z^t)}{\sum_{z=1,i}^{PS_1} f_z^t}$$

$$\forall \bar{x}1_z \text{updated by } \begin{cases} perching, & \text{if } i = 1 \\ patrolling, & \text{if } i = 2. \end{cases} \tag{12}$$

Then, $prob_{perch}$ and $prob_{pat}$ are calculated/updated as [21]

$$prob_{perch} = max(0.1, min(0.9, \frac{I_1}{I_1 + I_2})) \tag{13}$$

$$prob_{pat} = 1 - prob_{perch} \tag{14}$$

4) Linear Reduction of Size of PS_1 and PS_2: A linear reduction of PS_1 and PS_2 are made at the end of each iteration by eliminating the worst individual and random individual respectively. The new size of populations PS_1 and PS_2 is calculated by using equations (15) and (16) [8].

$$PS_1^{t+1} = round\left(\left(\frac{PS_{1,min} - PS_{1,max}}{FE_{max}}\right) * cFE\right) + PS_{1,max} \tag{15}$$

$$PS_2^{t+1} = round\left(\left(\frac{PS_{2,min} - PS_{2,max}}{FE_{max}}\right) * cFE\right) + PS_{2,max} \tag{16}$$

where $(PS_{1,max}, PS_{2,max})$ and $(PS_{1,min}, PS_{2,min})$ are the maximum and minimum values of PS_1 and PS_2 respectively, and FE_{max} and cFE are the maximum allowed function evaluation and current function evaluation respectively.

5) Adaptation of F, freq, CR and T: In this algorithm, an ensemble of parameter adaption techniques are used to auto-tune the parameter F. For parameters $freq$, CR and T, SHBA is used to sample new parameter setting during the process [6]. First of all we discuss the steps of the proposed SHBA technique which provide a efficient way to generate new parameter setting.

- A historical memory of size H for all the parameter is initialized equal to their default values. Default value of parameters F, CR, freq, T are 0.7, 0.5, 0.5, and 0.1 respectively.
- New parameter setting, CR_z,F_z,$freq_z$, and T_z, associates with individual $\bar{x}1_z$ is calculated by equations (17), (18), (19),and (20) respectively.

$$CR_z = M2(M_{CR,r}, 0.1) \tag{17}$$

$$F_z = randci(M_{F,r}, 0.1) \tag{18}$$

$$freq_z = randci(M_{CR,r}, 0.1) \tag{19}$$

$$T_z = M2(M_{T,r}, 0.05) \tag{20}$$

Where r is a integer from $[1, H]$, $randci$ gives the values randomly from cauchy distributions.

- In S_i, the values of parameter are recorded which is used by successful individual to update their function value [6]. After that the content of memory are updated by using equation (21]).

$$M_{i,d} = mean_{WA}(S_i) if S_i \neq null, \forall i \epsilon \{CR, F, freq, T\} \tag{21}$$

Where $1 < d < H$ is the index of the memory to be updated. It is initialized to 1, and then increased by 1 whenever an index of memory is updated and if it is greater than H, it is reset to 1. $mean_{WA}(S_i)$ is calculated by using equation (22)[6].

$$mean_{WA}(S_i) = \frac{\sum_{\gamma=1}^{|S_i|} w_\gamma S_{i,\gamma}^2}{\sum_{\gamma=1}^{|S_i|} w_\gamma S_{i,\gamma}} \forall i \epsilon \{CR, F, freq, T\} \tag{22}$$

where,

$$w_\gamma = \frac{\triangle f_\gamma}{\sum_{\gamma=1}^{|S_i|} \triangle f_\gamma} \tag{23}$$

and $\triangle f_\gamma = |f_{\gamma,old} - f_{\gamma,new}|$.

At each iteration $t \epsilon [0, t_{max}/2]$, a two tangential approaches is used to generate the new value of F, decreasing non-adaptive tangential approach and increasing adaptive tangential approach. One of these two approach is chosen randomly with equal probability to adapt F for each individual. In decreasing non-adaptive tangential approach, the new F_z is calculated by using equation (24)

$$F_z^t = \frac{1}{2}(tan(\pi(t+1))\frac{t_{max} - t}{t_{max}} + 1) \tag{24}$$

where t_{max} is maximum allowed iteration. On the other hand, in case of increasing adaptive tangential approach, the new F_z is calculate as shown below:

$$F_z^t = \frac{1}{2}(tan(2\pi freq_z * t)\frac{t}{t_{max}} + 1) \tag{25}$$

here $freq_z$ is adapted by using SHBA.

C. CMAR

EBOwithCMAR uses CMAR as described in previous section. It starts with a random initial population of size PS3 ($X_3 = \{\bar{x}3_1, \bar{x}3_2, ...\bar{x}3_{PS_3}\}$), where every individual is uniformly initialized within the bound. Initial mean is calculate by the arithmetic mean of X_3 [21]

D. Update of Probabilities $prob_1$ and $prob_2$

As proposed in Section-III-A, to update $prob_1$ and $prob_2$, the two factors considered in this paper are quality of solution and diversity of the population [21].

The normalized quality values, $\hat{Q}_i$ at the end of half of cycle, $\frac{CS}{2}$ are calculated by the equation(26) [21].

$$\hat{Q}_i = \frac{f_{\frac{CS}{2},i}^{best}}{f_{\frac{CS}{2},1}^{best} + f_{\frac{CS}{2},2}^{best}} \tag{26}$$

where $f_{\frac{CS}{2},i}^{best}$ is the best objective function value at the end of half of cycle, $\frac{CS}{2}$ by i^{th} algorithm.

Concurrently, the normalized diversity, $\hat{div}_i$ is calculated using equation (27) [21].

$$\hat{div}_i = \frac{div_i}{div_1 + div_2} \tag{27}$$

where div_i is the diversity rate of population with respect to best solution at the end of half of cycle, $\frac{CS}{2}$ by i^{th} algorithm. Then a progress index, PI_i is calculated by using equation (28).

$$PI_i = (1 - \hat{Q}_i) + \hat{div}_i, \forall i = 1, 2 \tag{28}$$

Finally, the probability, $prob_i$ is calculated as shown in equation (29) [21].

$$prob_i = max(0.1, min(0.9, \frac{PI_i}{PI_1 + PI_2})), \forall i = 1, 2 \tag{29}$$

If the sum of PI is equal to zero, $prob_1$ and $prob_2$ are set to 1.

E. Data Sharing

At the end of every cycle, CS, algorithm having a greater value of probability is considered to be best algorithm of that cycle. If EBO is considered as the best, then population X_3 is replaced by the random solution of population X_1. Parameters of CMAR is also reinitialized at default value except the step size, σ, where σ is calculated as $\sigma = \sigma_{initial} * (1 - cFE/FE_{max})$ [21].

On the other hand, if CMAR is reflected as the best, the worst individual in X_1 is replaced by the best individual in X_3. After data sharing, new cycle is again started and process repeated [21].

Algorithm 1 EBOwithCMAR

Define $PS \leftarrow PS_1+PS_2+PS_3$,$Cy \leftarrow 0$, $prob_1 = prob_2 \leftarrow 1$ and all other parameters required
for $i = 1$ to PS **do**
  $X_i \leftarrow$ uniformly distributed D random numbers
end for
Randomly assign PS_1, PS_2, and PS_3 individuals from X to $X_i, \forall i = 1, 2, 3$
while termination condition is not satisfied **do**
  $cy \leftarrow cy + 1$
  if $cy == \frac{CS}{2}$ **then**
    Calculate $prob_1$ and $prob_2$
  end if
  if $cy == CS$ **then**
    Share Data
    $prob_1 = 1$, and $prob_2 = 1$
    $cy \leftarrow 0$
  end if
  if $rand(0, 1) \leq prob_1$ **then**
    Apply EBO
    $cFE \leftarrow cFE + PS_1$
  end if
  if $rand(0, 1) \leq prob_2$ **then**
    Apply CMAR
    $cFE \leftarrow cFE + PS_1$
  end if
  if $rand(0, 1) \leq prob_{ls}\&cFE \geq 0.75 * FE_{max}$ **then**
    Apply SEQ
    $cFE \leftarrow cFE + FE_{seq}$
    if best solution is improved **then**
      $prob_{ls} \leftarrow 0.1$
      update X_1 and X_2
    else
      $prob_{ls} \leftarrow 0.0001$
    end if
  end if
  $t \leftarrow t + 1$
end while
return X

IV. Experimental Results

We have assessed the performance of EBOwithCMAR on the CEC2017 competition on real-parameter single objective optimization [17] and compared the performance of EBOwith-CMAR with the performance of joint winners of CEC2016.

A. Benchmark Suite

The CEC2017 benchmark suite covers 30 test functions with a distinct set of features. For all of the problems, the functions are evaluated on $10D$, $30D$, $50D$ and $100D$ within the search space $[-100, 100]^D$, where D is the dimensionality of the problem. In short, Functions $F1 - F3$ are unimodal, $F4 - F10$ are simple multimodal functions, $F11 - F20$ are hybrid functions, and finally, $F21 - F30$ are composite functions which merge multiple test functions into a multifaceted landscape. More details of the test suite are presented in [17].

B. Parameter Settings

For EBO, $PS_{1,max} = 18D$, $PS_{1,min} = 4$,$PS_{2,max} = 46.8D$, $PS_{2,min} = 10$ $H = 6$[8]. For CMAR, $PS_3 = 4 + (3log(D))$[20], and $\sigma = 0.3$. $CS = 100$ and 200 for the $10D$ and $30D$, respectively, and 300 for $50D$ and $100D$ problems. For local search, $prob_{ls} = 0.1$ and $cfe_{ls} = 0.25 * FE_{max}$ function evaluations [21].

TABLE I
ALGORITHM COMPLEXITY

	T_0	T_1	$\hat{T}_2$	$(\hat{T}_2$-$T_1)/T_0$
D = 10	0.0413	0.8218	7.5794	163.6223
D = 30		1.1507	6.591	131.7264
D = 50		1.8792	8.7886	167.2978
D = 100		5.6887	18.4969	310.1259

TABLE II
STATISTICAL RESULTS OF EBOWITHCMAR ON THE $10D$ BENCHMARK FUNCTION SUITE, AVERAGED OVER 51 INDEPENDENT RUNS

Func	Best	Worst	Median	Mean	Std
1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
6	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
7	1.04E+01	1.10E+01	1.05E+01	1.06E+01	1.75E-01
8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
10	1.25E-01	2.17E+02	1.36E+01	3.72E+01	5.39E+01
11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
12	0.00E+00	2.37E+02	1.18E+02	9.02E+01	7.44E+01
13	0.00E+00	7.95E+00	3.13E-02	2.17E+00	2.53E+00
14	0.00E+00	9.95E-01	0.00E+00	6.05E-02	2.36E-01
15	3.81E-05	5.00E-01	3.07E-02	1.09E-01	1.74E-01
16	2.62E-02	9.35E-01	4.43E-01	4.17E-01	1.98E-01
17	1.00E-02	1.01E+00	4.94E-02	1.47E-01	2.03E-01
18	3.92E-04	2.00E+01	4.09E-01	7.00E-01	2.77E+00
19	0.00E+00	1.22E-01	1.79E-02	1.50E-02	1.88E-02
20	0.00E+00	3.12E-01	0.00E+00	1.47E-01	1.57E-01
21	1.00E+02	2.02E+02	1.00E+02	1.14E+02	3.52E+01
22	2.17E+01	1.00E+02	1.00E+02	9.85E+01	1.10E+01
23	3.00E+02	3.03E+02	3.00E+02	3.00E+02	7.07E-01
24	1.00E+02	3.30E+02	1.00E+02	1.66E+02	9.97E+01
25	3.98E+02	4.43E+02	3.98E+02	4.12E+02	2.12E+01
26	2.00E+02	3.00E+02	3.00E+02	2.65E+02	4.74E+01
27	3.90E+02	3.95E+02	3.90E+02	3.92E+02	2.40E+00
28	0.00E+00	5.84E+02	3.00E+02	3.07E+02	7.18E+01
29	2.27E+02	2.45E+02	2.30E+02	2.31E+02	3.77E+00
30	3.95E+02	4.43E+02	3.95E+02	4.07E+02	1.78E+01

C. Algorithm Complexity

This section expresses the algorithm complexity of our EBOwithCMAR code as described in [17].The algorithm was implemented using MATLAB 2016a and was run on PC with an INTEL CPU (3.40GHZ) and 10GB RAM. Table I presents the algorithm complexity on D = 10, 30, 50, and 100. As outlined in [17], T_0 is the time assessed by running the following test problem:

$$x = 0.65;$$
$$for\ i = 1 : 1000000$$
$$x = x + x; x = x/2; x = x * x;$$
$$x = sqrt(x); x = log(x);$$
$$x = exp(x); x = x/(x + 2);$$
$$end$$

T_1 is the time to execute for benchmark function $f18$ for $200,000$ objective function evaluations while T_2 is the time

TABLE III
STATISTICAL RESULTS OF EBOWITHCMAR ON THE $30D$ BENCHMARK FUNCTION SUITE, AVERAGED OVER 51 INDEPENDENT RUNS

Func	Best	Worst	Median	Mean	Std
1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4	0.00E+00	6.41E+01	5.86E+01	5.65E+01	1.11E+01
5	0.00E+00	7.96E+00	2.98E+00	2.78E+00	1.74E+00
6	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
7	3.17E+01	3.60E+01	3.34E+01	3.35E+01	8.37E-01
8	0.00E+00	5.97E+00	1.99E+00	2.02E+00	1.32E+00
9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
10	9.84E+02	1.94E+03	1.38E+03	1.41E+03	2.15E+02
11	0.00E+00	6.40E+01	1.99E+00	4.49E+00	8.77E+00
12	1.50E+01	1.10E+03	4.35E+02	4.63E+02	2.63E+02
13	1.38E+00	2.66E+01	1.66E+01	1.49E+01	6.25E+00
14	2.52E+00	2.80E+01	2.21E+01	2.19E+01	3.84E+00
15	2.79E-01	1.02E+01	3.39E+00	3.69E+00	2.15E+00
16	3.25E+00	2.57E+02	2.11E+01	4.26E+01	5.69E+01
17	9.95E+00	4.32E+01	3.10E+01	2.98E+01	7.50E+00
18	2.04E+01	2.55E+01	2.20E+01	2.21E+01	1.09E+00
19	3.95E+00	1.33E+01	8.14E+00	8.04E+00	2.28E+00
20	2.34E+01	6.56E+01	3.49E+01	3.57E+01	7.50E+00
21	1.00E+02	2.07E+02	2.03E+02	1.99E+02	2.02E+01
22	1.00E+02	1.00E+02	1.00E+02	1.00E+02	0.00E+00
23	3.45E+02	3.60E+02	3.51E+02	3.51E+02	3.51E+00
24	1.00E+02	4.28E+02	4.24E+02	4.18E+02	4.55E+01
25	3.83E+02	3.87E+02	3.87E+02	3.87E+02	7.56E-01
26	2.00E+02	9.28E+02	3.00E+02	5.37E+02	3.06E+02
27	4.93E+02	5.11E+02	5.03E+02	5.02E+02	4.03E+00
28	3.00E+02	4.14E+02	3.00E+02	3.08E+02	2.88E+01
29	4.07E+02	4.60E+02	4.34E+02	4.33E+02	1.13E+01
30	1.94E+03	2.10E+03	1.97E+03	1.99E+03	4.21E+01

TABLE IV
STATISTICAL RESULTS OF EBOWITHCMAR ON THE $50D$ BENCHMARK FUNCTION SUITE, AVERAGED OVER 51 INDEPENDENT RUNS

Func	Best	Worst	Median	Mean	Std
1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4	0.00E+00	1.42E+02	2.85E+01	4.29E+01	3.32E+01
5	2.98E+00	1.45E+01	6.96E+00	7.58E+00	2.42E+00
6	0.00E+00	4.31E-07	4.79E-08	8.54E-08	1.14E-07
7	5.53E+01	6.25E+01	5.79E+01	5.79E+01	1.53E+00
8	3.98E+00	1.39E+01	7.96E+00	7.91E+00	2.47E+00
9	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
10	2.14E+03	3.94E+03	3.16E+03	3.11E+03	4.01E+02
11	1.93E+01	3.31E+01	2.62E+01	2.64E+01	3.36E+00
12	1.23E+03	6.90E+03	1.78E+03	1.94E+03	8.34E+02
13	7.96E+00	1.22E+02	4.49E+01	4.14E+01	2.48E+01
14	2.50E+01	4.09E+01	3.09E+01	3.12E+01	3.52E+00
15	9.59E+00	4.07E+01	2.93E+01	2.94E+01	5.20E+00
16	1.25E+02	5.91E+02	3.44E+02	3.46E+02	1.46E+02
17	6.28E+01	4.65E+02	2.61E+02	2.75E+02	8.63E+01
18	2.25E+01	4.94E+01	3.05E+01	3.20E+01	5.99E+00
19	1.71E+01	3.11E+01	2.44E+01	2.45E+01	3.94E+00
20	5.76E+01	3.50E+02	1.21E+02	1.47E+02	7.44E+01
21	2.02E+02	2.22E+02	2.10E+02	2.11E+02	4.06E+00
22	1.00E+02	3.97E+03	1.00E+02	3.65E+02	9.24E+02
23	4.15E+02	4.64E+02	4.34E+02	4.34E+02	8.16E+00
24	4.99E+02	5.16E+02	5.06E+02	5.06E+02	3.85E+00
25	4.80E+02	5.63E+02	4.80E+02	4.89E+02	2.47E+01
26	3.00E+02	1.31E+03	9.06E+02	7.06E+02	4.06E+02
27	5.03E+02	5.40E+02	5.23E+02	5.22E+02	7.75E+00
28	4.59E+02	5.08E+02	4.59E+02	4.67E+02	1.79E+01
29	3.09E+02	4.06E+02	3.45E+02	3.47E+02	1.97E+01
30	5.79E+05	6.89E+05	6.14E+05	6.18E+05	3.62E+04

to execute EBOwithCMAR with $200,000$ objective function evaluations of $f18$ in D dimensions. T_2 is calculated for 5 independent runs, and $\hat{T}_2$ denotes the mean of T_2. Finally, the algorithm complexity is expressed by the linear growth of $(\hat{T}_2 - T_1)/T_0$.

TABLE V
STATISTICAL RESULTS OF EBOWITHCMAR ON THE $100D$ BENCHMARK FUNCTION SUITE, AVERAGED OVER 51 INDEPENDENT RUNS

Func	Best	Worst	Median	Mean	Std
1	0.00E+00	5.15E-08	0.00E+00	1.33E-09	7.52E-09
2	0.00E+00	3.78E+11	95166	1.27E+10	5.99E+10
3	7.61E-08	5.20E-06	1.86E-07	2.99E-07	7.04E-07
4	0.00E+00	2.18E+02	1.97E+02	1.93E+02	3.09E+01
5	1.79E+01	4.28E+01	2.79E+01	2.87E+01	5.28E+00
6	5.13E-06	3.41E-05	1.47E-05	1.63E-05	7.13E-06
7	1.16E+02	1.38E+02	1.21E+02	1.22E+02	4.47E+00
8	1.79E+01	4.68E+01	2.79E+01	2.97E+01	7.48E+00
9	0.00E+00	8.95E-02	0.00E+00	1.76E-03	1.25E-02
10	6.77E+03	1.34E+04	9.50E+03	9.91E+03	1.91E+03
11	3.38E+01	1.42E+02	6.15E+01	6.56E+01	2.00E+01
12	2.45E+03	6.06E+03	4.13E+03	4.19E+03	7.89E+02
13	1.19E+02	5.50E+02	2.14E+02	2.45E+02	8.84E+01
14	8.17E+01	2.06E+02	1.36E+02	1.38E+02	2.96E+01
15	8.60E+01	2.70E+02	1.63E+02	1.65E+02	3.87E+01
16	7.86E+02	2.53E+03	1.36E+03	1.41E+03	3.76E+02
17	5.35E+02	1.69E+03	1.18E+03	1.21E+03	2.57E+02
18	1.08E+02	4.05E+02	2.31E+02	2.37E+02	5.94E+01
19	8.16E+01	1.72E+02	1.12E+02	1.15E+02	1.88E+01
20	7.80E+02	2.09E+03	1.35E+03	1.36E+03	3.09E+02
21	2.44E+02	3.15E+02	2.58E+02	2.60E+02	1.06E+01
22	1.00E+02	1.44E+04	1.02E+04	1.02E+04	2.70E+03
23	5.50E+02	6.23E+02	5.76E+02	5.77E+02	1.31E+01
24	8.99E+02	9.74E+02	9.16E+02	9.19E+02	1.32E+01
25	5.77E+02	7.74E+02	7.05E+02	7.16E+02	3.71E+01
26	3.00E+02	3.53E+03	3.19E+03	2.77E+03	1.08E+03
27	5.48E+02	6.19E+02	5.86E+02	5.88E+02	1.53E+01
28	3.00E+02	5.77E+02	5.19E+02	5.10E+02	6.01E+01
29	8.61E+02	1.84E+03	1.28E+03	1.28E+03	2.42E+02
30	2.18E+03	2.79E+03	2.36E+03	2.40E+03	1.51E+02

D. Statistical results

We did our evaluation following the recommendations of the CEC2017 benchmark competition [17]. When the mismatch between the best solution obtained and the optimal solution was less or equal to 10^{-8}, the error was considered as 0. For all of the benchmark functions the number of dimensions D $=$ 10, 30, 50, and 100 and maximum number of objective function evaluation was Dx$10,000$ (i.e., $100,000$, $300,000$, $500,000$, and $1,000,000$ respectively) [17]. The number of independent runs per problem was 51, and the average performance of these runs was calculated. The statistical outcomes of EBOwithCMAR on $D = 10$, 30, 50, and 100 are demonstrated in Tables II-V. Each table shows the best, worst, median, and the mean over the 51 runs of the error value obtained.

E. Comparison of EBOwithCMAR with the joint winners of CEC2016

In this subsection, the performance of the EBOwithCMAR algorithm is compared with the performance of joint winner of CEC2016 competition on the real-parameter single objective optimization listed below:

1) LSHADE-EpSin[22]
2) UMOEA-II[21]

TABLE VI
COMPARISON SUMMARY OF EBOWITHCMAR AGAINST UMOEAS-II AND LSHADE-EPSIN

EBOwithCMAR	10D			30D			50D		
Vs	Better	Similar	Worse	Better	Similar	Worse	Better	Similar	Worse
UMOEAs-II	8	17	5	12	14	4	18	4	8
LSHADE-EpSin	20	9	1	14	10	6	18	5	7

A summery of comparison of the performance of all algorithms is presented in Table VI. Generally speaking, EBOwithCMAR can provide better mean results than other algorithms in the majority of test problems. Consedering the mean error values of all algorithms on all dimensions of CEC-2017 problems, EBOwithCMAR is consistently better than LSHADE-EpSin and UMOEAs-II. Additionally, the Wilcoxon test was conducted to find if there is a statistically significant difference between EBOwithCMAR and the two other algorithms compared in this paper. Based on the results shown in VI, EBOwithCMAR showed superior performance to that of LSHADE-EpSin and UMOEAS-II based on average fitness values achieved for all dimensions

V. Conclusion

This paper introduced a real parameter optimization technique, EBOwithCMAR, which is a modified version of the EBO algorithm. This paper proposed a self-adaptive framework in order to adapt the parameter during the optimization process and CMAR algorithm to improve the local search capability. Self-adaptation uses a new adaptive tangential approaches in order to set the values of the scaling factor automatically. The EBOwithCMAR was benchmarked on the problem suite used in the special session and competitions on CEC2017. When the performance of proposed algorithm is compared with the other state-of-arts algorithm taken from the literature, we found that the EBOwithCMAR provides better performance.

References

[1] M. Dorigo and M. Birattari, "Ant colony optimization," in *Encyclopedia of Machine Learning*. Springer, 2010, pp. 36–39.

[2] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*. Springer, 2010, pp. 760–766.

[3] A. Bohre, G. Agnihotri, and M. Dubey, "Hybrid butterfly based particle swarm optimization for optimization problems," in *Networks Soft Computing (ICNSC), 2014 First International Conference on*, Aug 2014, pp. 172–177.

[4] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Computing and Applications*, pp. 1–20, 2015.

[5] A. Kumar, R. K. Misra, and D. Singh, "Butterfly optimizer," in *2015 IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI)*, Dec 2015, pp. 1–6.

[6] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2013, pp. 71–78.

[7] ——, "Evaluating the performance of shade on cec 2013 benchmark problems," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2013, pp. 1952–1959.

[8] R. Tanabe and A. S. Fukunaga, "Improving the search performance of shade using linear population size reduction," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*. IEEE, 2014, pp. 1658–1665.

[9] A. Auger and N. Hansen, "A restart cma evolution strategy with increasing population size," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 2. IEEE, 2005, pp. 1769–1776.

[10] C. García-Martínez, M. Lozano, F. Herrera, D. Molina, and A. M. Sánchez, "Global and local real-coded genetic algorithms based on parent-centric crossover operators," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1088–1113, 2008.

[11] M. A. M. De Oca, T. Stutzle, K. Van den Enden, and M. Dorigo, "Incremental social learning in particle swarms," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 2, pp. 368–384, 2011.

[12] J. Brest and M. S. Maučec, "Population size reduction for the differential evolution algorithm," *Applied Intelligence*, vol. 29, no. 3, pp. 228–247, 2008.

[13] J. L. J. Laredo, C. Fernandes, J. J. Merelo, and C. Gagné, "Improving genetic algorithms performance via deterministic population shrinkage," in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*. ACM, 2009, pp. 819–826.

[14] S. Ghosh, S. Das, S. Roy, S. M. Islam, and P. N. Suganthan, "A differential covariance matrix adaptation evolutionary algorithm for real parameter optimization," *Information Sciences*, vol. 182, no. 1, pp. 199–219, 2012.

[15] W. E. Hart, N. Krasnogor, and J. E. Smith, *Recent advances in memetic algorithms*. Springer Science & Business Media, 2004, vol. 166.

[16] P. Moscato *et al.*, "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms," *Caltech concurrent computation program, C3P Report*, vol. 826, p. 1989, 1989.

[17] A. M. L. J. Awad, NH, B. Qu, and P. Suganthan, "Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective bound constrained real-parameter numerical optimization," *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, 2017.

[18] D. Zaharie, "A comparative analysis of crossover variants in differential evolution," *Proceedings of IMCSIT*, vol. 2007, pp. 171–181, 2007.

[19] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es)," *Evolutionary computation*, vol. 11, no. 1, pp. 1–18, 2003.

[20] N. Hansen, "The cma evolution strategy: A tutorial," *arXiv preprint arXiv:1604.00772*, 2016.

[21] S. Elsayed, N. Hamza, and R. Sarker, "Testing united multi-operator evolutionary algorithms-ii on single objective optimization problems," in *Evolutionary Computation (CEC), 2016 IEEE Congress on*. IEEE, 2016, pp. 2966–2973.

[22] N. H. Awad, M. Z. Ali, P. N. Suganthan, and R. G. Reynolds, "An ensemble sinusoidal parameter adaptation incorporated with l-shade for solving cec2014 benchmark problems," in *Evolutionary Computation (CEC), 2016 IEEE Congress on*. IEEE, 2016, pp. 2958–2965.

[23] A. Kumar and S. C. Gupta, "A new Initial Centroid finding Method based on Dissimilarity Tree for K-means Algorithm," *ArXiv e-prints*, Jun. 2015.