processes

MDPI

Article

Global Evolution Commended by Localized Search for Unconstrained Single Objective Optimization

Rashida Adeeb Khanum [1,†], Muhammad Asif Jan [2,*,†], Nasser Tairan [3,†], Wali Khan Mashwani [2,†], Muhammad Sulaiman [4,†], Hidayat Ullah Khan [5,†] and Habib Shah [3,†]

1 Jinnah College for Women, University of Peshawar, Peshawar 25000, Pakistan; rakhan@uop.edu.pk
2 Institute of Numerical Sciences, Kohat University of Science & Technology, Kohat 26000, Pakistan; mashwanigr8@gmail.com
3 College of Computer Science, King Khalid University, Abha 61321, Saudi Arabia; nmtairan@kku.edu.sa (N.T.); habibshah.uthm@gmail.com (H.S.)
4 Department of Mathematics, Abdul Wali Khan University Mardan, Mardan 23200, Pakistan; sulaiman513@gmail.com
5 Department of Economics, Abbottabad University of Science & Technology, Abbottabad 22010, Pakistan; masmaleo@yahoo.com
* Correspondence: majan@kust.edu.pk or majan.math@gmail.com; Tel.: +92-313-998-6123
† These authors contributed equally to this work.

Received: 27 April 2019; Accepted: 27 May 2019; Published: 11 June 2019



Abstract: Differential Evolution (DE) is one of the prevailing search techniques in the present era to solve global optimization problems. However, it shows weakness in performing a localized search, since it is based on mutation strategies that take large steps while searching a local area. Thus, DE is not a good option for solving local optimization problems. On the other hand, there are traditional local search (LS) methods, such as Steepest Decent and Davidon–Fletcher–Powell (DFP) that are good at local searching, but poor in searching global regions. Hence, motivated by the short comings of existing search techniques, we propose a hybrid algorithm of a DE version, reflected adaptive differential evolution with two external archives (RJADE/TA) with DFP to benefit from both search techniques and to alleviate their search disadvantages. In the novel hybrid design, the initial population is explored by global optimizer, RJADE/TA, and then a few comparatively best solutions are shifted to the archive and refined there by DFP. Thus, both kinds of searches, global and local, are incorporated alternatively. Furthermore, a population minimization approach is also proposed. At each call of DFP, the population is decreased. The algorithm starts with a maximum population and ends up with a minimum. The proposed technique was tested on a test suite of 28 complex functions selected from literature to evaluate its merit. The results achieved demonstrate that DE complemented with LS can further enhance the performance of RJADE/TA.

Keywords: optimization; evolutionary computation; population minimization; hybridization; local search; global search; adaptive differential evolution; external archives; metaheuristics

1. Introduction

Nonlinear unconstrained optimization is an active research area, since many real-life challenges/problems can be modeled as a continuous nonlinear optimization problem [1]. To deal with this kind of optimization problems, various nature-inspired population based search mechanisms have been developed in the past [2]. A few of those are Differential Evolution (DE) [3,4], Evolution Strategies (ES) [2,5], Partical Swarm Optimization (PSO) [6–9], Ant Colony Optimization (ACO) [10–13], Bacterial Foraging Optimization (BFO) [14,15], Genetic Algorithm (GA) [16–18], Genetic Programming

(GP) [2,19–21], Cuckoo Search (CS) [22,23], Estimation of Distribution Algorithm (EDA) [24–28] and Grey Wolf Optimization (GWO) [29,30].

DE does not need specific information about the complicated problem at hand [31]. That is why DE is implemented to solve a wide variety of optimization problems in the past two decades [30,32–34]. DE has merits over PSO, GA, ES and ACO, as it depends upon few control parameters. Its implementation is very easy and user friendly, too [2]. Due to these advantages, we selected DE to perform global search in the suggested hybrid design. In addition, because of its easy nature, DE is implemented widely [35–42] on practical optimization problems [35–42]. However, its convergence to known optima is not guaranteed [2,31,43]. Stagnation of DE is another weakness identified in various studies [31].

Traditional search approaches, such as Nelder–Mead algorithm, Steepest Descent and DFP [44] may be hybridized with DE to improve its search capability. Implementing LS into a global search for enhancing the solution quality is called Memetic Algorithms (MAs) [31,45]. Some of the recent MAs can be found in [1,31]. Very recently, Broyden–Fletcher–Goldfarb–Shanan LS was merged with an adaptive DE version, JADE [46], which produced the MA, Hybridization of Adaptive Differential Evolution with an Expensive Local Search Method [47]. In the majority of the established designs, LS is implemented to the overall best solutions, while in our design it is applied to the migrated elements of the archive. In addition, the population is adaptively decreased.

In this work, we propose a hybrid algorithm that combines DFP [44,48,49] with a recently developed algorithm, RJADE/TA [50], to enhance RJADE/TA's performance in local regions. The main idea is to operate DFP on the elements that are shifted to archive and record the information from both solutions, the previously brought forward and the new potential solutions to discourage the chance of losing the globally best solution. For this purpose, firstly, DFP is implemented to the archived information. Secondly, a decreasing population mechanism is suggested. The new algorithm is denoted by RJADE/TA-ADP-LS.

The structure of this work is as follows. Section 2 presents primary DE, DFP, and RJADE/TA methods. Section 3 describes the literature review. In Section 4, the suggested hybrid algorithm is outlined. Section 5 is devoted to the validation of results achieved by RJADE/TA-ADP-LS. At the end, the conclusions are summarized in Section 6.

2. Primary DE, DFP, and RJADE/TA

We reviewed in detail traditional DE and JADE in our previous works [47,50]. Here, we briefly review primary DE, DFP and RJADE/TA for ready reference.

2.1. Primary DE

DE [3,4] starts with a random population in the given search region. After initialization, a mutation strategy, where three different individuals from population are randomly selected and the scaled difference of the two individuals to the third one, target vector is added to produce a mutant vector. Following mutation, the mutant and the target vectors are combined through a crossover operator to produce a trial vector. At last, the target and trial vectors are compared based on a fitness function to select the better one for the next generation (see Lines 7–20 of Algorithm 1).

Algorithm 1 Outlines of RJADE/TA Procedure.

To form the primary population P_p produce $N^{[pop]}$ vectors uniformly and randomly, $\mathbf{w}^{\{y\}}_{[j,s_1]}, \mathbf{w}^{\{y\}}_{[j,s_2]}, \ldots, \mathbf{w}^{[y]}_{[j,s_{N^{[pop]}}]}$;
$M^{[first]} = M^{[sec]} = \emptyset$;
Initialize $\lambda CR = \lambda F = 0.5$; $p = 5\%$; $c = 0.1$;
Set $S_{CR} = S_F = \emptyset$;
Evaluate P_p;
while $FEs < MaxFEs$ **do**
    $F_j = rand(\lambda F, 0.1)$;
    Randomly sample $\mathbf{w}^{[p,y]}_{(best)}$ in $100p\%$ pop;
    Choose $\mathbf{w}^{\{y\}}_{[i,s_1]} \neq \mathbf{w}^{\{y\}}_{[i,s]}$ in P_p;
    Choose $\tilde{\mathbf{w}}^{\{y\}}_{[i,s_2]} \neq \mathbf{w}^{\{y\}}_{[i,s_2]}$ in $P_p \cup M^{[first]}$ do random selection;
    Produce the mutant vector $\mathbf{w}^{\{y\}}_{[i,mut]}$ as $\mathbf{w}^{\{y\}}_{[i,mut]} = \mathbf{w}^{\{y\}}_{[i,s]} + F_j(\mathbf{w}^{\{p,y\}}_{(best)} - \mathbf{w}^{\{y\}}_{[i,s]}) + F_j(\mathbf{w}^{\{y\}}_{[i,s_1]} - \tilde{\mathbf{w}}^{\{y\}}_{[i,s_2]})$;
    Produce the trial vector $\mathbf{q}^{\{y\}}_{[i,j]}$ as follows.
    for $i = 1$ to n **do**
        if $i < i_{rand}$ or $rand(0,1) < CR_j$ **then**
            $q^{\{y\}}_{[i,j]} = w^{\{y\}}_{[i,mut_j]}$;
        else
            $q^{\{y\}}_{[i,j]} = w^{\{y\}}_{[i,s_j]}$;
        end if
    end for
    Best selection $\{\mathbf{w}^{\{y\}}_{[i,s]}, \mathbf{q}^{\{y\}}_{[i,s]}\}$;
    if $\mathbf{q}^{\{y\}}_{[i,s]}$ is the best **then**
        $\mathbf{w}^{\{y\}}_{[i,s]} \rightarrow M^{[first]}$, $CR_j \rightarrow S_{CR}$, $F_j \rightarrow S_F$;
    end if
    If size of $M^{[first]} > N^{[pop]}$, delete extra solutions from $M^{[first]}$ randomly;
    Update $M^{[sec]}$ as follows.
    if $y = \kappa$ **then**
        $\mathbf{w}^{\{y\}}_{[j,best]} \rightarrow M^{[sec]}$;
        $P_p - \mathbf{w}^{\{y\}}_{[j,best]}$;
        Centroid calculation $\rightarrow \mathbf{w}^{\{y\}}_{[j,c]} = \frac{1}{N^{[pop]}-1} \sum_{i=2}^{N^{[pop]}} \mathbf{w}^{\{y\}}_{[j,c]}$;
        Reflection mechanism $\rightarrow \mathbf{w}^{\{y\}}_{[j,r]} = \mathbf{w}^{\{y\}}_{[j,c]} + (\mathbf{w}^{\{y\}}_{[j,c]} - \mathbf{w}^{\{y\}}_{[j,best]})$;
    end if
    $\lambda CR = (1-c) \cdot \lambda CR + c \cdot mean_A(S_{CR})$;
    $\lambda F = (1-c) \cdot \lambda F + c \cdot mean_L(S_F)$;
end while
Result: The best solution $\mathbf{w}^{\{y\}}_{(best)}$ corresponding to minimum function $f(\mathbf{w})$ value from $P_p UM^{[sec]}$ in the optimization.

2.2. Reflected Adaptive Differential Evolution with Two External Archives (RJADE/TA)

RJADE/TA [50] is an adaptive DE variant. Its main idea is to archive comparatively best solutions of the population at regular interval of optimization process and reflect the overall poor solutions. RJADE/TA inserts the following techniques in JADE. The techniques are presented in Table 1.

Table 1. Algorithmic parameters.

$M^{[first]}$	First archive	$M^{[Sec]}$	Second archive
P_p	Primary population	$N^{[pop]}$	Population size
FEs	Function evaluations	$MaxFEs$	Maximum function evaluations
λ	FEs of RJADE/TA	κ	Gap between two successive updates of $M^{[sec]}$
λCR	Crossover probability	λF	Mutation scaling factor
S_{CR}	Set of successful crossover probabilities	S_F	Set of successful mutation factors
w	No. of iterations of DFP	r	Number of migrated solutions to $M^{[sec]}$
$\mathbf{w}^{\{y\}}_{[j,new]}$	j^{th} New candidate/solution at iteration y	$\mathbf{w}^{\{y\}}_{[j,best]}$	j^{th} Ever best candidate/solution at iteration y

To prevent premature convergence and stagnation, the best solution, $\mathbf{w}^{\{y\}}_{[j,best]}$ is replaced by its reflection in RJADE/TA and is then shifted to the second archive $M^{[sec]}$.

The reflected solution replaces $\mathbf{w}^{\{y\}}_{[j,best]}$ in the population and the ever best candidate $\mathbf{w}^{\{y\}}_{[j,best]}$ by itself is migrated to the second archive $M^{[sec]}$. RJADE/TA maintains two archives, termed as $M^{[first]}$ and $M^{[sec]}$ for convenience. After half of available resources are utilized ($MaxFEs$), the first archive update of the second archive, $M^{[sec]}$, is made. Afterwards, $M^{[sec]}$ is updated adaptively with a continuing intermission of generations (see Algorithm 1).

The overall best candidates are transferred to $M^{[sec]}$, whereas $M^{[first]}$ records the recently explored poor solutions. The size of $M^{[first]}$ is fixed, equal to population size $N^{[pop]}$, while the size of $M^{[sec]}$ may exceed $N^{[pop]}$. As $M^{[sec]}$ keeps information of all best solutions found, no solution is deleted from it. $M^{[sec]}$ records only one solution of the current iteration, it may be a child or a parent, whereas $M^{[first]}$ makes a history of more than one inferior "parent solutions" only. $M^{[first]}$ is updated at every iteration and $M^{[sec]}$, initialized as $\emptyset$, is updated with a gap of κ iterations adaptively. The recorded history of $M^{[first]}$ is utilized in reproduction later on. In contrast, in $M^{[sec]}$, the recorded best individual is reflected with a new solution, which is then sent to the population. Once a candidate solution is posted to $M^{[sec]}$, it remains passive during the whole optimization. When the search procedures are terminated, then the recoded information contributes towards the selection of the best candidate solution.

2.3. Davidon–Fletcher–Powell (DFP) Method

The DFP method is a variable metric method, which was first proposed by Davidon [51] and then modified by Powell and Fletcher [52]. It belongs to the class of gradient dependent LS methods. If a right line search is used in DFP method, it will assure convergence (minimization) [49]. It calculates the difference between the old and new points, as given in Equation (1). Then, it finds the difference of the gradients at these points as calculated in Equation (2).

$$\mathbf{t}_{\{w\}} = \mathbf{w}^{\{y\}}_{[j+1]} - \mathbf{w}^{[j]} \tag{1}$$

$$\mathbf{t}_{\{g\}} = \nabla f(\mathbf{w}^{[j+1]}) - \nabla f(\mathbf{w}^{[j]}). \tag{2}$$

It then updates the Hessian matrix $\mathbf{H}$ as presented in Equation (3). Afterwards, it locates the optimal search direction $\mathbf{s}^{[j]}$ with the help of the Hessian matrix information as calculated in Equation (4). Finally, the output solution $\mathbf{w}^{[j+1]}$ is computed by Equation (5), where $\alpha^{[j]}$ is calculated by a line search method; golden section search method is used in this work.

$$\mathbf{H}^{[j+1]} = \mathbf{H}^{[j]} + \frac{(\mathbf{t}'_{\{w\}}\mathbf{t}_{\{w\}})}{\mathbf{t}'_{\{w\}}\mathbf{t}_{\{g\}}} - \frac{(\mathbf{H}^{[j]}\mathbf{t}'_{\{g\}}\mathbf{t}_{\{g\}}\mathbf{H}^{[j]})}{\mathbf{t}_{\{g\}}\mathbf{H}^{[j]}\mathbf{t}'_{\{g\}}} \tag{3}$$

$$\mathbf{s}^{[j]} = -\mathbf{H}^{[j]} \nabla f(\mathbf{w}^{[j]}) \tag{4}$$

$$\mathbf{w}^{[j+1]} = \mathbf{w}^{[j]} + \alpha^{[j]} \mathbf{s}^{[j]} \tag{5}$$

3. Related Work

To fix the above-mentioned weaknesses of DE, many researchers merged various LS techniques in DE. Nelder–Mead LS is hybridized with DE [53] to improve the local exploitation of DE. Recently, two new LS strategies are proposed and hybridized iteratively with DE in [1,31]. These hybrid designs show performance improvement over the algorithms in comparison. Two LS strategies, Trigonometric and Interpolated, are inserted in DE to enhance its poor exploration. Two other LS techniques are merged in DE along with a restart strategy to improve its global exploration [54]. This algorithm is statistically sound, as the obtained results are better than other algorithms. Furthermore, alopex-based LS is merged in DE [55] to improve its diversity of population. In another experiment, DE's slow convergence is enhanced by combining orthogonal design LS [56] with it. To avert local optima in DE, random LS is hybridized [57] with it. On the other hand, some researchers borrowed DE's mutation and crossover in traditional LS methods (see, e.g., [58,59]).

To the best of our knowledge, none of the reviewed algorithms in this section integrate DFP into DE's framework. Further, the proposed work here maintains two archives: the first one stores inferior solutions and the second one keeps information of best solutions migrated to it by the global search. Furthermore, the second archive improves the solutions quality further by implementing DFP there. Hence, our proposed work has the advantage that the second archive keeps complete information of the solution before and after LS. This way, any good solution found is not lost. It also adopts a population decreasing mechanism.

4. Developed Algorithm

As discussed in the literature review, LS techniques, due to their demerits, should not be used alone to solve optimization problems [2]. The global optimality of global evolution techniques is very high, but they can get stuck in local regions and cannot fine tune the solution at hand. Thus, motivated by above issues of global/local techniques, we hybridize a global optimizer RJADE/TA with DFP to enhance the convergence in both regions. The new design is named as RJADE/TA-ADP-LS. We specifically handle unconstrained, nonlinear, continuous, and single objective optimization problems in the current work.

RJADE/TA-ADP-LS

The initial population is evolved globally by RJADE/TA [50] until λ% of the function evaluations; that is, after RJADE/TA's iterative mutation, crossover, selection and $M^{[first]}$ process, as shown in Algorithm 1, the population is sorted and the current best solution $\mathbf{w}^{[k]}_{(i,best)}$ is translated to $M^{[sec]}$. This best solution may be a parent or a child solution. The DFP is applied to the shifted elements for w iterations. After implementation of DFP, a new improved solution $\mathbf{w}^{[k]}_{(i,new)}$ is produced from an old migrant. Then, the previously explored best solution and this new solution are posted to archive $M^{[sec]}$. Unlike our perviously proposed archive $M^{[sec]}$ in RJADE/TA, where the archive keeps the record of best solutions only and no LS is implemented, $M^{[sec]}$, as mentioned above, in this method maintains information of both solutions, i.e., the migrated best solution and its improved version, if any, after implementation of DFP.

The archive $M^{[sec]}$ is updated after regular intervals of κ generations (20 here). The migrated solutions and those explored by DFP remain there during the entire evolution process. When the evolution process completes, the overall best candidate is selected from $P_pUM^{[sec]}$. The novelty of RJADE/TA-ADP-LS is that it employs DFP to the archived solutions only, unlike all hybrid designs reviewed in Section 3.

In the proposed hybrid mechanism, we implement DFP to the migrated best solution to obtain its improved form, but without reflection, as displayed in the flowchart given in Figure 1, unlike in our recently proposed work [60]. Moreover, in this model, we propose adaptively decreasing population (ADP) mechanism different from the fixed population approach of Khanum et al. [60]. We refer to this new hybrid as RJADE/TA-ADP-LS throughout this work. The idea of RJADE/TA-ADP-LS is novel in proposing the ADP approach, because, in the literature, majority of the evolutionary algorithms (as reviewed in Section 3) maintain a fixed population throughout the searching process.

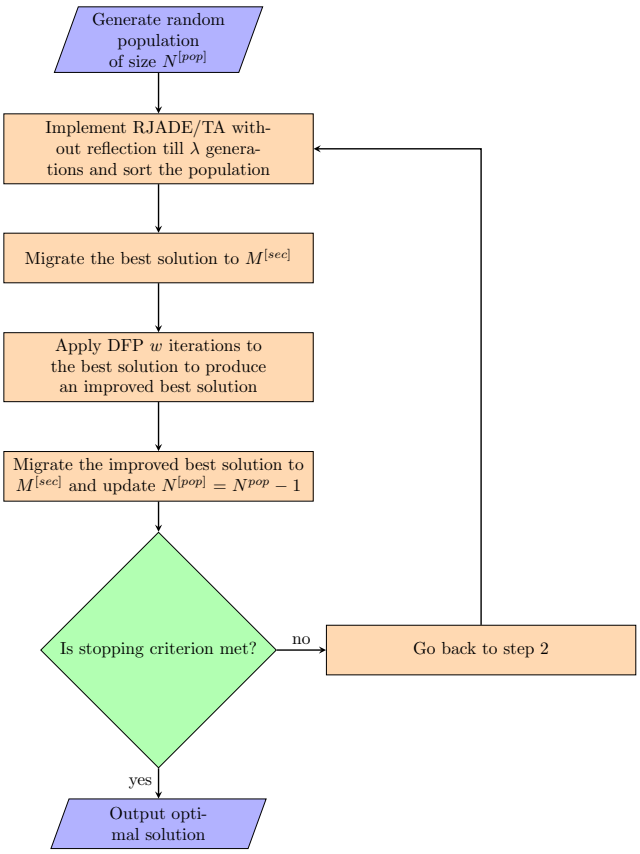


Figure 1. Flowchart of RJADE/TA-ADP-LS.

In this design, when the first update of $M^{[sec]}$ is made after half of the available resources are spent, DFP is applied to the archive members. The implementations of DFP and ADP are shown in Algorithm 2. Both the previously located best solution, $\mathbf{w}^{\{y\}}_{[j,best]}$, and the one exploited by DFP, $\mathbf{w}^{\{y\}}_{[j,new]}$, are propagated to $M^{[sec]}$. No reflection is made here to compensate the decreasing population. The ADP approach (Algorithm 2, Lines 6–8) is implemented as:

$$P_p = P_p - \mathbf{w}^{\{y\}}_{[j,best]}. \tag{6}$$

Hence,

$$P_p = \{\mathbf{w}^{\{y\}}_{[j,s_1]}, \mathbf{w}^{\{y\}}_{[j,s_2]}, \ldots, \mathbf{w}^{\{y\}}_{[j,s_{N^{[pop]}-1}]}\} \tag{7}$$

$$f(P_p) = \{f(\mathbf{w}^{\{y\}}_{[j,s_1]}), f(\mathbf{w}^{\{y\}}_{[j,s_2]}), \ldots, f(\mathbf{w}^{\{y\}}_{[j,s_{N^{[pop]}-1}]})\} \tag{8}$$

Every time $M^{[sec]}$ is updated, the migrated element is removed from the current population P_p (see Equation (6)), and the population is decreased by one. Thus, after each break of κ generations, $r(=$ the number of times the κ breaks occur) solutions are removed from $N^{[pop]}$, and the population

size is updated to $N^{[pop]} - r$, as demonstrated on Line 11 of Algorithm 2. Furthermore, the function values are updated accordingly (see Equations (7) and (8)). In ADP approach, the algorithm begins with a maximum population and terminates with a minimum population.

Algorithm 2 RJADE/TA-ADP-LS.

1: Update $M^{[sec]}$ as follows.
2: **if** $k = \kappa$ **then**
3: $\mathbf{w}^{\{y\}}_{[j,best]} \rightarrow M^{[sec]}$;
4: Apply DFP to $\mathbf{w}^{\{y\}}_{[j,best]}$ to prduce $\mathbf{w}^{\{y\}}_{[j,new]}$;
5: $\mathbf{w}^{\{y\}}_{[j,new]} \rightarrow M^{[sec]}$;
6: $P_p = \{\mathbf{w}^{\{y\}}_{([j,s_1]}, \mathbf{w}^{\{y\}}_{[j,s_2]}, \ldots, \mathbf{w}^{\{y\}}_{[j,s_{N^{[pop]}-1}]}\}$;
7: $f(P_p) = \{f(\mathbf{w}^{\{y\}}_{[j,s_1]}), f(\mathbf{w}^{\{y\}}_{[j,s_2]}), \ldots, f(\mathbf{w}^{\{y\}}_{[j,s_{N^{[pop]}-1}]})\}$;
8: $N^{[pop]} = N^{[pop]} - 1$;
9: **end if**
10: Terminate the iteration;
11: Repeat the process r number of times and update $N^{[pop]} = N^{[pop]} - r$.

5. Validation of Results

In this section, first we briefly illustrate the five algorithms used for comparison and then the experimental results are presented.

5.1. Global Search Algorithms in Comparison

Among the five algorithms for comparison, the first two, RJADE/TA and RJADE/TA-LS, are our recently proposed hybrid algorithms, while the remaining three, jDE, jDEsoo and jDErpo, are non-hybrid, but adaptive and popular DE variants.

5.1.1. RJADE/TA

RJADE/TA [50], similar to RJADE/TA-ADP-LS, utilizes two archives for information. One of the archives stores inferior solutions, while the other keeps a record of superior solutions. However, in RJADE/TA-ADP-LS, the second archive stores elite solutions, which are then improved by DFP. Further details of RJADE/TA can be seen in Section 2.2.

5.1.2. RJADE/TA-LS

RJADE/TA-LS [60] is a very recently proposed hybrid version of global and local search. However, it is different from RJADE/TA-ADP-LS in the sense that it utilizes reflection mechanism and a fixed population, while RJADE/TA-ADP-LS uses DFP as LS without reflection and a population decreasing approach.

5.1.3. jDE

jDE [61] is an adaptive version of DE, which is based on self-adaption of control parameters F and CR. In jDE, the parameters F and CR keep changing during the evolution process, while the population size $N^{[pop]}$ is kept unchanged. Every solution in jDE has its own F and CR values. Better individuals are produced due to better values of F and CR. Such parameter values translate to upcoming generations of jDE. Because of its unique mechanism and simplicity, jDE has gained popularity among researchers in the field of optimization. Since its establishment, people use it to compare with their own algorithms.

5.1.4. jDEsoo and jDErpo

jDEsoo [62] is a new version of DE that deals with single-objective optimization. jDEsoo subdivides the population and implements more than one DE strategies. To enhance diversity of population, it removes those individuals from population that remain unchanged in the last few generations. It was primarily developed for CEC 2013 competition.

jDErpo [61] is an improvement of jDE. It is based on the following mechanisms. Firstly, it incorporates two mutation strategies, different from jDE, DE and RJADE/TA. Secondly, it uses adaptively increasing strategy for adjusting the lower bounds of control parameters. Thirdly, it utilizes two pairs of control parameters for two different mutation strategies in contrast to one pair of parameters used in jDE, classic DE and RJADE/TA. jDErpo was also specially designed for solving CEC 2013 competition problems.

5.2. Parameter Settings/Termination Criteria

Experiments were performed on 28 benchmark test problems of CEC 2013 [63]. They are referred as BMF1–BMF28. The parameters' settings were kept the same as demanded in [63]. The dimension n of each problem was set to 10, population size $N^{[pop]}$ to 100, and the $MaxFEs$ to $10,000 \times n$. The number of elite solutions r was kept as 1. The iterations number w of DFP was set to 2. The reduction of population per archive update r was also chosen as 1. The gap κ between successive updates of $M^{[sec]}$ was kept as 20. The optimization was terminated if either $MaxFEs$ were reached or the difference between the means of function error values was less than 10^{-8}, as suggested in [50,63].

Table 2. Comparison of RJADE/TA-ADP-LS with Well Established Algorithms.

Bench Marks	jDE	jDEsoo	jDErpo	RJADE/TA	RJADE/TA-ADP-LS
BMF1	0.0000e + 0=	0.0000e + 0=	0.0000e + 0=	0.0000e + 0=	0.0000e + 0
BMF2	7.6534e − 05−	1.7180e + 03−	0.0000e + 0=	0.0000e + 0=	0.0000e + 00
BMF3	1.3797e + 0+	1.6071e + 0+	3.7193e − 05+	1.2108e + 02+	2.0350e + 02
BMF4	3.6639e − 08+	1.2429e − 01+	0.0000e + 0+	1.1591e + 02+	2.9749e + 02
BMF5	0.0000e + 0=	0.0000e + 0=	0.0000e + 0=	0.0000e + 0=	0.0000e + 00
BMF6	8.6581e + 0−	8.4982e + 04−	5.3872e + 0+	7.8884e + 0−	5.4656e + 00
BMF7	2.7229e − 03+	9.4791e − 01−	1.6463e − 03+	1.5927e − 01+	2.3707e − 01
BMF8	2.0351e + 01=	2.0348e + 01+	2.0343e + 01+	2.0366e + 01−	2.0352e + 01
BMF9	2.6082e + 0+	2.7464e + 0+	6.4768e − 01+	4.4593e + 0+	4.6182e + 00
BMF10	4.5263e − 02−	7.0960e − 02−	6.4469e − 02−	3.5342e − 02−	3.2488e − 02
BMF11	0.0000e + 0=	0.0000e + 0=	0.0000e + 0=	0.0000e + 0=	0.0000e + 0
BMF12	1.2304e + 01−	6.1144e + 0+	1.3410e + 01−	7.7246e + 0−	7.0574e + 00
BMF13	1.3409e + 01−	7.8102e + 0+	1.4381e + 01−	6.7571e + 0+	9.7072e + 00
BMF14	0.0000e + 0+	5.0208e − 02−	1.9367e + 01−	1.1994e − 02−	5.3105e − 03
BMF15	1.1650e + 03−	8.4017e + 02−	1.1778e + 03−	6.6660e + 02+	7.3411e + 02
BMF16	1.0715e + 0−	1.0991e + 0−	1.0598e + 0−	1.1336e + 0−	1.0545e + 00
BMF17	1.0122e + 01=	9.9240e + 0+	1.0997e + 01−	1.0122e + 01=	1.0122e + 01
BMF18	3.2862e + 01−	2.7716e + 01−	3.2577e + 01−	2.2715e + 01+	2.4399e + 01
BMF19	4.3817e − 01−	3.1993e − 01−	7.4560e − 01−	4.4224e − 01−	4.2674e − 01
BMF20	3.0270e + 0−	2.7178e + 0−	2.5460e + 0+	2.5317e + 0+	2.6153e + 00
BMF21	3.7272e + 02+	3.5113e + 02+	3.7272e + 02+	3.9627e + 02+	4.0019e + 02
BMF22	7.9231e + 01−	9.1879e + 01−	9.7978e + 01−	2.7022e + 01−	1.3178e + 01
BMF23	1.1134e + 03−	8.1116e + 02−	1.1507e + 03−	7.0015e + 02−	4.8553e + 02
BMF24	2.0580e + 02−	2.0851e + 02−	1.8865e + 02−	2.0217e + 02−	1.0823e + 02
BMF25	2.0471e + 02−	2.0955e + 02−	1.9885e + 02−	2.0314e + 02−	1.7732e + 02
BMF26	1.8491e + 02−	1.9301e + 02−	1.1732e + 02+	1.2670e + 02−	1.2096e + 02
BMF27	4.7470e + 02−	4.9412e + 02−	3.0000e + 02+	3.0351e + 02+	3.0514e + 02
BMF28	2.9216e + 02−	2.8824e + 02−	2.9608e + 02−	2.8824e + 02−	2.8500e + 02
−	17	17	14	13	
+	6	8	10	10	
=	5	3	4	5	

5.3. Comparison of RJADE/TA-ADP-LS against Established Global Optimizers

The mean of function error values, the difference between known and approximated values, for jDE, jDEsoo, jDErpo, RJADE/TA and RJADE/TA-ADP-LS, are presented in Table 2. In Table 2,

+ indicates that the algorithm won against our algorithm, RJADE/TA-ADP-LS; − indicates that the particular algorithm lost against our algorithm; and = indicates that both algorithms obtained the same statistics. The comparison of RJADE/TA-ADP-LS with other competitors showed its outstanding performance against all of them. RJADE/TA-ADP-LS achieved higher mean values than jDE and jDEsoo on 17 out of 28 problems; the many − signs in columns 2 and 3 of Table 2 support this fact. In contrast, jDE and jDEsoo performed better on six and eight problems, respectively.

RJADE/TA-ADP-LS showed performance improvement against jDErpo and RJADE/TA algorithms as well. In general, RJADE/TA-ADP-LS performed better than all algorithms in comparison, especially in the category of multimodal and composite functions. The proposed mechanism is not only based on LS for local tuning with no reflection, but it also implements an ADP approach, which could be the reasons for its good performance.

5.4. Performance Evaluation of RJADE/TA-ADP-LS Versus RJADE/TA-LS

We empirically studied the performance of RJADE/TA-ADP-LS against RJADE/TA-LS. Table 3 presents the mean results achieved by both methods in 51 runs. The best results are shown in bold face. It is very clear from the results in Table 3 that the proposed RJADE/TA-ADP-LS performed higher than RJADE/TA-LS on 13 out of 28 problems. Furthermore, on five problems, they obtained the same results. RJADE/TA-LS showed performance improvement on 10 test problems.

Table 3. Comparing RJADE/TA-ADP-LS with RJADE/TA-LS.

		BMF1	**BMF2**	**BMF3**	**BMF4**	**BMF5**	**BMF6**	**BMF7**
RJADE/TA-LS		**0.0000e+00**	**0.0000e+00**	2.5750e + 02	**3.9511e+01**	**0.0000e+00**	6.9264e + 00	**2.3707e-01**
RJADE/TA-ADP-LS	Mean	**0.0000e+00**	**0.0000e+00**	2.0350e + 02	**2.9749e+02**	**0.0000e+00**	**5.4656e+00**	2.3707e − 01
		BMF8	**BMF9**	**BMF10**	**BMF11**	**BMF12**	**BMF13**	**BMF14**
RJADE/TA-LS		**2.0342e+01**	4.4888e + 00	**3.2488e-02**	**0.0000e+00**	**6.8613e+00**	**7.9039e+00**	7.3105e − 003
RJADE/TA-ADP-LS	Mean	2.0352e + 01	**4.6182e+00**	**3.2488e-02**	**0.0000e+00**	7.0574e + 00	9.7072e + 00	**5.3105e-03**
		BMF15	**BMF16**	**BMF17**	**BMF18**	**BMF19**	**BMF20**	**BMF21**
RJADE/TA-LS		**6.6733e+02**	1.0855e + 00	**1.0122e+01**	**1.0122e+01**	4.4752e − 01	**2.5707e+00**	**3.9627e+02**
RJADE/TA-ADP-LS	Mean	7.3411e + 02	**1.0545e+00**	**1.0122e+01**	2.4399e + 01	**4.2674e-01**	2.6153e + 00	4.0019e + 02
		BMF22	**BMF23**	**BMF24**	**BMF25**	**BMF26**	**BMF27**	**BMF28**
RJADE/TA-LS		2.0589e + 01	6.7549e + 02	1.9809e + 02	2.0190e + 02	1.3596e + 02	**3.0033e+02**	3.0000e + 02
RJADE/TA-ADP-LS	Mean	**1.3178e+01**	**4.8553e+02**	**1.0823e+02**	**1.7732e+02**	**1.2096e+02**	3.0514e + 02	**2.8500e+02**

It is interesting to note that RJADE/TA-ADP-LS showed outstanding performance in the category of composite functions, where it solved BMF22–BMF28 better than RJADE/TA-LS. Again, the two different mechanisms, the ADP approach and the LS search with out reflection, of RJADE/TA-ADP-LS could be the reasons for its better performance. Among 28 problems, RJADE/TA-LS was better on 10 functions. Further, Table 4 presents the percentage performance of RJADE/TA-ADP-LS and RJADE/TA-LS. Since on five test problems, both algorithms showed equal results, thus we compared the percentage for the remaining 23 problems. As shown in Table 4, RJADE/TA-ADP-LS was able to solve 57% of problems against 43% of problems solved by RJADE/TA-LS out of 23 test instances.

Table 4. Comparing RJADE/TA-ADP-LS with RJADE/TA-LS.

Algorithms	**RJADE/TA-ADP-LS**	**RJADE/TA-LS**
Number of Problems solved in total of 23	13 of 23	10 of 23
% age	**57%**	43%

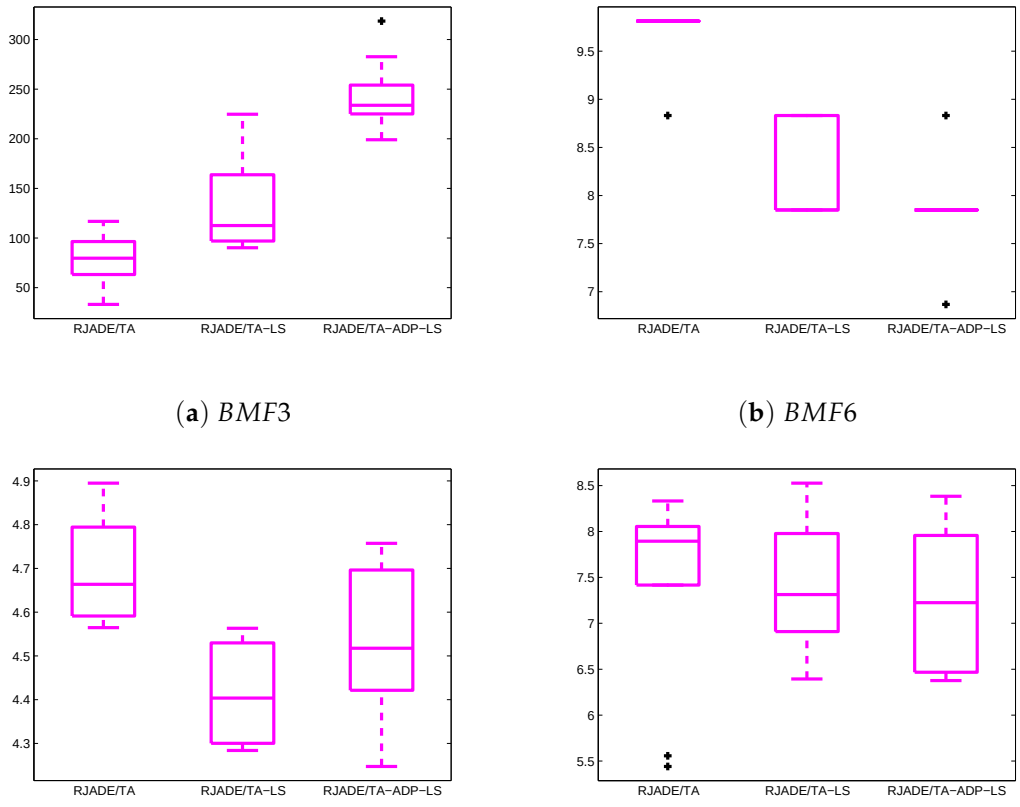
Furthermore, box plots were plotted from all means obtained in 25 runs of RJADE/TA, RJADE/TA-LS and RJADE/TA-ADP-LS. Figures 2 and 3 plot one function from each three functions.

Box plots are very good tools to show the spread of the data. Figure 2b–d shows that the boxes obtained by RJADE/TA-ADP-LS were lower than the other two boxes, indicating its better performance. Figure 2a presents the plot of BMF3, in which the two boxes in comparison were lower than RJADE/TA-ADP-LS, thus they were better.

Figure 3b,d,f shows that the boxes obtained by RJADE/TA-ADP-LS on BMF19, BMF25 and BMF27 were lower than the boxes of RJADE/TA and RJADE/TA-LS, indicating higher performance of RJADE/TA-ADP-LS. Figure 3a,c,e shows that the two other algorithms were better on the respective test instances.

5.5. Analysis/Discussion of Various Parameters Used

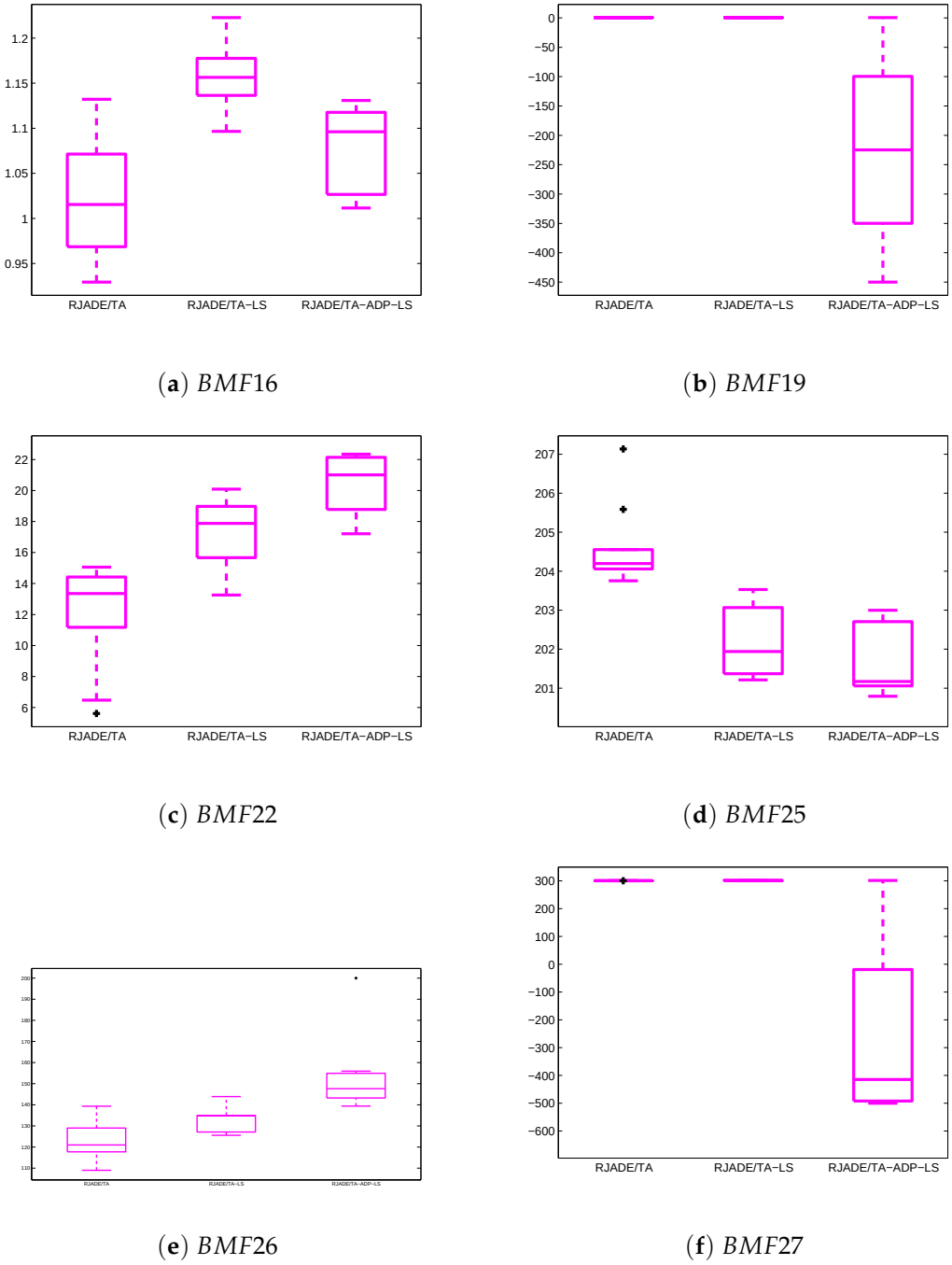
The number of solutions r to be migrated to archive and undergo DFP was kept as 1, since DFP is an expensive method due to gradient calculation. Further, its application to more than one solution might slow down the algorithm. The users may take two, but at most three is suggested. The number of iteration w of DFP to archive elements was kept as 2. DFP is a very good method; it could fine tune the solutions in only two iterations. Moreover, the decreasing number r of population per archive update was also chosen as 1. Since the archive was updated after regular gap of global evolution, each time population was decreased by one. However, if we reduced it by more than one solutions, then a stage would come where the diversity of the population would be decreased and the algorithm would either stop at local optima or converge prematurely. We suggest that the decreasing number be at most 3. In general, these parameters are user defined but should be chosen wisely to compliment the global and local search together, instead of premature convergence or stagnation.



(**a**) *BMF*3 (**b**) *BMF*6

(**c**) *BMF*9 (**d**) *BMF*13

Figure 2. Box plots of various algorithms in comparison.



(**a**) *BMF*16 (**b**) *BMF*19

(**c**) *BMF*22 (**d**) *BMF*25

(**e**) *BMF*26 (**f**) *BMF*27

Figure 3. Box plots of various algorithms in comparison.

6. Conclusions

This paper proposed a new hybrid algorithm, RJADE/TA-ADP-LS, where a LS mechanism, DFP is combined with a DE based global search scheme, RJADE/TA to benefit from their searching capabilities in local and global regions. Further, a population decreasing mechanism is also adopted. The key idea is to shift the overall best solution to archive at specified regular intervals of RJADE/TA, where it undergoes DFP for further improvement. The archive stores both the best solution and its improved form. Furthermore, the population is decreased by one solution at each archive update. We evaluated and compared our hybrid method with five established algorithms on test suit of CEC 2013. The results demonstrated that our new algorithm is better than other competing algorithms on majority of the tested problems, particularly our algorithm showed superior performance on hard

multimodal and composite problems of CEC 2013. In future, the present work will be extended to constrained optimization. As a second task, some other gradient free LS methods, global optimizers and archiving strategies will be tried to design more efficient algorithms for global optimization.

Author Contributions: Conceptualization, R.A.K. and M.A.J.; methodology, R.A.K., M.A.J., and W.K.M.; software, R.A.K., N.T. and H.S.; validation, H.U.K., M.S. and H.S.; formal analysis, R.A.K., M.A.J., and W.K.M.; investigation, R.A.K., M.A.J. and M.S.; resources, N.T. and H.S.; writing—original draft preparation, R.A.K., M.A.J.; writing—review and editing, H.U.K. and M.S.; project administration, N.T.; and funding acquisition, N.T. and H.S.

Funding: The authors would like to thank King Khalid University of Saudi Arabia for supporting this research under the grant number R.G.P.2/7/38.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Price, K.V. Eliminating drift bias from the differential evolution algorithm. In *Advances in Differential Evolution*; Springer: Berlin, Germany, 2008; pp. 33–88.
2. Xiong, N.; Molina, D.; Ortiz, M.L.; Herrera, F. A walk into metaheuristics for engineering optimization: principles, methods and recent trends. *Int. J. Comput. Intell. Syst.* **2015**, *8*, 606–636. [CrossRef]
3. Storn, R.; Price, K.V. Differential Evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
4. Storn, R. Differential evolution research—Trends and open questions. In *Advances in Differential Evolution*; Springer: Berlin, Germany, 2008; pp. 1–31.
5. Engelbrecht, A.; Pampara, G. Binary Differential Evolution Strategies. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2007), Singapore, 25–28 September 2007; pp. 1942–1947.
6. Kennedy, J.; Eberhart, R.C. Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; pp. 1942–1948.
7. Kennedy, J.; Eberhart, R. A Discrete Binary Version of the Partical Swarm Algorithm. In Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics, Orlando, FL, USA, 12–15 October 1997; pp. 4104–4109.
8. Eberhart, R.C.; Kennedy, J. A New Optimizer using Particle Swarm Theory. In Proceedings of the 6th International Symposium on Micromachine and Human Science, Nagoya, Japan, 4–6 October 1995; pp. 39–43.
9. Eberhart, R.C.; shi, Y. Guest Editorial: Special Issue on Particle Swarm Optimization. *IEEE Trans. Evol. Comput.* **2004**, *8*, 201–203. [CrossRef]
10. Dorigo, M. Ant colony optimization. *Scholarpedia* **2007**, *2*, 1461. [CrossRef]
11. Dorigo, M.; Birattari, M. Ant colony optimization. In *Encyclopedia of Machine Learning*; Springer: Berlin, Germany, 2011; pp. 36–39.
12. Al-Salami, N.M. System evolving using ant colony optimization algorithm. *J. Comput. Sci.* **2009**, *5*, 380. [CrossRef]
13. Cui, L.; Zhang, K.; Li, G.; Wang, X.; Yang, S.; Ming, Z.; Huang, J.Z.; Lu, N. A smart artificial bee colony algorithm with distance-fitness-based neighbor search and its application. *Future Gener. Comput. Syst.* **2018**, *89*, 478–493. [CrossRef]
14. Passino, K.M. Bacterial foraging optimization. *Int. J. Swarm Intell. Res. (IJSIR)* **2010**, *1*, 1–16. [CrossRef]
15. Gazi, V.; Passino, K.M. Bacteria foraging optimization. In *Swarm Stability and Optimization*; Springer: Berlin, Germany, 2011; pp. 233–249.
16. Moscato, P. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech Concurr. Comput. Prog. C3P Rep.* **1989**, *826*, 1989.
17. Fan, S.K.S.; Zahara, E. A hybrid Simplex Search and Partical Swarm optimization for unconstrained optimization. *Eur. J. Oper. Res.* **2007**, *181*, 527–548. [CrossRef]
18. Yuen, S.Y.; Chow, C.K. A Genetic Algorithm that Adaptively Mutates and Never Revisits. *IEEE Trans. Evol. Comput.* **2009**, *13*, 454–472. [CrossRef]
19. Koza, J.R. *Genetic Programming II, Automatic Discovery Of Reusable Subprograms*; MIT Press: Cambridge, MA, USA, 1992.

20. Koza, J.R. Genetic programming as a means for programming computers by natural selection. *Stat. Comput.* **1994**, *4*, 87–112. [CrossRef]
21. Koza, J.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*; MIT Press: Cambridge, MA, USA, 1992.
22. Yang, X.S.; Deb, S. Cuckoo search via Lévy flights. In Proceedings of the IEEE World Congress on Nature & Biologically Inspired Computing, Coimbatore, India, 9–11 December 2009; pp. 210–214.
23. Yang, X.S.; Deb, S. Engineering optimisation by cuckoo search. *arXiv* **2010**, arXiv:1005.2908.
24. Larrañaga, P.; Lozano, J.A. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*; Springer: Berlin, Germany, 2001; Volume 2.
25. Zhang, Q.; Sun, J.; Tsang, E.; Ford, J. Hybrid estimation of distribution algorithm for global optimization. *Eng. Comput.* **2004**, *21*, 91–107. [CrossRef]
26. Zhang, Q.; Muhlenbein, H. On the convergence of a class of estimation of distribution algorithms. *IEEE Trans. Evol. Comput.* **2004**, *8*, 127–136. [CrossRef]
27. Lozano, J.A.; Larrañaga, P.; Inza, I.; Bengoetxea, E. *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*; Springer: Berlin, Germany, 2006; Volume 192.
28. Hauschild, M.; Pelikan, M. An introduction and survey of estimation of distribution algorithms. *Swarm Evol. Comput.* **2011**, *1*, 111–128. [CrossRef]
29. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
30. Gupta, S.; Deep, K. Hybrid Grey Wolf Optimizer with Mutation Operator. In *Soft Computing for Problem Solving*; Springer: Berlin, Germany, 2019; pp. 961–968.
31. Leon, M.; Xiong, N. Eager random search for differential evolution in continuous optimization. In *Portuguese Conference on Artificial Intelligence*; Springer: Berlin, Germany, 2015; pp. 286–291.
32. Maučec, M.S.; Brest, J.; Bošković, B.; Kačič, Z. Improved Differential Evolution for Large-Scale Black-Box Optimization. *IEEE Access* **2018**, *6*, 29516–29531. [CrossRef]
33. Biswas, P.P.; Suganthan, P.; Wu, G.; Amaratunga, G.A. Parameter estimation of solar cells using datasheet information with the application of an adaptive differential evolution algorithm. *Renew. Energy* **2019**, *132*, 425–438. [CrossRef]
34. Sacco, W.F.; Rios-Coelho, A.C. On Initial Populations of Differential Evolution for Practical Optimization Problems. In *Computational Intelligence, Optimization and Inverse Problems with Applications in Engineering*; Springer: Berlin, Germany, 2019; pp. 53–62.
35. Wu, G.; Shen, X.; Li, H.; Chen, H.; Lin, A.; Suganthan, P. Ensemble of differential evolution variants. *Inf. Sci.* **2018**, *423*, 172–186. [CrossRef]
36. Awad, N.H.; Ali, M.Z.; Mallipeddi, R.; Suganthan, P.N. An improved differential evolution algorithm using efficient adapted surrogate model for numerical optimization. *Inf. Sci.* **2018**, *451*, 326–347. [CrossRef]
37. Al-Dabbagh, R.; Neri, F.; Idris, N.; Baba, M. Algorithm Design Issues in Adaptive Differential Evolution: Review and taxonomy. *Swarm Evol. Comput.* **2018**, *43*, 284–311. [CrossRef]
38. Betzig, L.L. *Despotism, Social Evolution, and Differential Reproduction*; Routledge: Abingdon, UK, 2018.
39. Opara, K.R.; Arabas, J. Differential Evolution: A survey of theoretical analyses. *Swarm Evol. Comput.* **2018**, *44*, 546–558. [CrossRef]
40. Das, S.; Mullick, S.S.; Suganthan, P. Recent advances in differential evolution An updated survey. *Swarm Evol. Comput.* **2016**, *27*, 1–30. [CrossRef]
41. Cui, L.; Huang, Q.; Li, G.; Yang, S.; Ming, Z.; Wen, Z.; Lu, N.; Lu, J. Differential Evolution Algorithm With Tracking Mechanism and Backtracking Mechanism. *IEEE Access* **2018**, *6*, 44252–44267. [CrossRef]
42. Cui, L.; Li, G.; Zhu, Z.; Ming, Z.; Wen, Z.; Lu, N. Differential evolution algorithm with dichotomy-based parameter space compression. *Soft Comput.* **2018**, *23*, 1–18. [CrossRef]
43. Meng, Z.; Pan, J.S.; Zheng, W. Differential evolution utilizing a handful top superior individuals with bionic bi-population structure for the enhancement of optimization performance. *Enterpr. Inf. Syst.* **2018**, 1–22. [CrossRef]
44. Fletcher, R. *Practical Methods of Optimization*, 2nd ed.; Wiley: Hoboken, NJ, USA, 1987; pp. 80–87.
45. Lozano, M.; Herrera, F.; Krasnogor, N.; Molina, D. Real-Coded Memetic Algorithms with Crossover Hill-Climbing. *Evol. Comput.* **2004**, *12*, 273–302. [CrossRef] [PubMed]
46. Zhang, J.; Sanderson, A.C. JADE: adaptive differential evolution with optional external archive. *IEEE Trans. Evol. Comput.* **2009**, *13*, 945–958. [CrossRef]

47. Khanum, R.A.; Jan, M.A.; Tairan, N.M.; Mashwani, W.K. Hybridization of Adaptive Differential Evolution with an Expensive Local Search Method. *J. Optim.* **2016**, *1016*, 1–14. [CrossRef]
48. Davidon, W.C. Variable metric method for minimization. *SIAM J. Optim.* **1991**, *1*, 1–17. [CrossRef]
49. Antoniou, A.; Lu, W.S. *Practical Optimization: Algorithms and Engineering Applications*; Springer: Berlin, Germany, 2007.
50. Khanum, R.A.; Tairan, N.; Jan, M.A.; Mashwani, W.K.; Salhi, A. Reflected Adaptive Differential Evolution with Two External Archives for Large-Scale Global Optimization. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*, 675–683.
51. Spedicato, E.; Luksan, L. Variable metric methods for unconstrained optimization and nonlinear least squares. *J. Comput. Appl. Math.* **2000**, *124*, 61–95.
52. Mamat, M.; Dauda, M.; bin Mohamed, M.; Waziri, M.; Mohamad, F.; Abdullah, H. Derivative free Davidon-Fletcher-Powell (DFP) for solving symmetric systems of nonlinear equations. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *332*, 012030. [CrossRef]
53. Ali, M.; Pant, M.; Abraham, A. Simplex Differential Evolution. *Acta Polytech. Hung.* **2009**, *6*, 95–115.
54. Khanum, R.A.; Jan, M.A.; Mashwani, W.K.; Tairan, N.M.; Khan, H.U.; Shah, H. On the hybridization of global and local search methods. *J. Intell. Fuzzy Syst.* **2018**, *35*, 3451–3464. [CrossRef]
55. Leon, M.; Xiong, N. A New Differential Evolution Algorithm with Alopex-Based Local Search. In *International Conference on Artificial Intelligence and Soft Computing*; Springer: Berlin, Germany, 2016; pp. 420–431.
56. Dai, Z.; Zhou, A.; Zhang, G.; Jiang, S. A differential evolution with an orthogonal local search. In Proceedings of the IEEE Congress on Evolutionary Computation, Cancun, Mexico, 20–23 June 2013; pp. 2329–2336.
57. Ortiz, M.L.; Xiong, N. Using random local search helps in avoiding local optimum in differential evolution. In Proceedings of the IASTED, Innsbruck, Austria, 17–19 February 2014; pp. 413–420.
58. Khanum, R.A.; Zari, I.; Jan, M.A.; Mashwani, W.K. Reproductive nelder-mead algorithms for unconstrained optimization problems. *Sci. Int.* **2015**, *28*, 19–25.
59. Zari, I.; Khanum, R.A.; Jan, M.A.; Mashwani, W.K. Hybrid (N)elder-mead algorithms for nonlinear numerical optimization. *Sci. Int.* **2015**, *28*, 153–159.
60. Khanum, R.A.; Jan, M.A.; Mashwani, W.K.; Khan, H.U.; Hassan, S. RJADETA integrated with local search for continuous nonlinear optimization. *Punjab Univ. J. Math.* **2019**, *51*, 37–49.
61. Brest, J.; Zamuda, A.; Fister, I.; Boskovic, B. Some Improvements of the Self-Adaptive jDE Algorithm. In Proceedings of the IEEE Symposium on Differential Evolution (SDE), Orlando, FL, USA, 9–12 December 2014; pp. 1–8.
62. Brest, J.; Boskovic, B.; Zamuda, A.; Fister, I.; Mezura-Montes, E. Real Parameter Single Objective Optimization using self-adaptive differential evolution algorithm with more strategies. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Cancun, Mexico, 20–23 June 2013; pp. 377–383.
63. Liang, J.; Qu, B.; Suganthan, P.; Hernández-Díaz, A.G. Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization, 2013. Available online: http://al-roomi.org/multimedia/CEC_Database/CEC2013/RealParameterOptimizationCEC2013_RealParameterOptimization_TechnicalReport.pdf (accessed on 22 April 2019).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).