Punjab University
Journal of Mathematics (ISSN 1016-2526)
Vol. 51(4)(2019) pp. 37-49

RJADE/TA Integrated with Local Search for Continuous Nonlinear Optimization

Rashida Adeeb Khanum
Jinnah College for Women,
University of Peshawar, Pakistan.
Email: rakhan@upesh.edu.pk

Muhammad Asif Jan
Department of Mathematics,
Kohat University of Science & Technology, Pakistan.
Email: majan@kust.edu.pk; majan.math@gmail.com

Wali Khan Mashwani
Department of Mathematics,
Kohat University of Science & Technology, Pakistan.
Email: mashwanigr8@gmail.com

Hidayat Ullah Khan
Department of Economics,
Kohat University of Science & Technology, Pakistan.
Email: masmaleo@yahoo.com

Saima Hassan
Institute of Computing,
Kohat University of Science & Technology, Pakistan.
Email: saimahassan@kust.edu.pk

Received: 25 April, 2018 / Accepted: 19 September, 2018 / Published online: 14 January, 2019

Abstract. This paper proposes a memetic algorithm by integrating adaptively a local search approach with a recently proposed variant of differential evolution, reflected adaptive differential evolution with two external archives (RJADE-TA). The main objective is to bring together the exploration factor of differential evolution and exploitative component of local search to solve continuous optimization problems. A novel hybrid local search mechanism is proposed and demonstrated leading to a crossbred version of RJADE-TA. In other words, the best solutions after a regular toll of global search are migrated to an archive, where Davidon Fletcher Powell local search method is implemented to the migrated solutions. Afterwards, the population is updated with new reflected so1utions to prevent premature convergence. The proposed approach is novel in the sense that most of the algorithms store only inferior or superior solutions in the archives. None of the algorithms implemented the local search inside the archive. Thus, this combination is a new one. To evaluate the merit of developed meme, a benchmark suite of complex 28 functions from CEC 2013 test problems is selected and implemented. The experimental results demonstrate that this integration of local search strategy can further improve the performance of RJADE-TA. They further reveal that the proposed meme outperforms differential evolution based algorithms on most of the tested problems.

AMS (MOS) Subject Classification Codes: 35S29; 40S70; 25U09
Key Words: Population Minimization, Local Search, Global Optimization, Memetic Algorithms, Adaptive Differential Evolution, External Archives.

1. INTRODUCTION

Continuous optimization is a popular research field, because various real world tasks can be formulated as a continuous function [32, 35]. Various population based search techniques have been proposed in last few decades to deal with this kind of continuous optimization problems. For instance, Genetic Algorithm (GA) [17, 33, 47], Particle Swarm Optimization (PSO) [15, 20, 21], Evolution Strategies (ES) [16], Cuckoo Search (CS) [45, 46], Ant Colony Optimization (ACO) [12, 13, 40], Bacterial Foraging Optimization (BFO) [19, 38] and Differential Evolution(DE) [39, 41, 42].

Among the above established algorithms, DE is a metaheuristics that can deal with complex problems effectively, without having specific information about problems [28]. In the past two decades, DE has been shown as a powerful optimizer for a wide variety of optimization problems [30]. DE has the advantage over PSO, GA, ES and ACO as it contains very few parameters to controll. It is user friendly and easy to implement. Because of these merits, we have chosen DE as a global optimizer in the proposed hybrid strategy. Due to its simple structure and few controlling parameters, DE has been widely applied [4, 5, 8, 10, 36] to practical optimization problems. However, DE does not guarantee fast convergence to the known optimum [28, 31]. Stagnation of DE has been observed in many experimental studies [28].

Local Search (LS) [18] techniques can be hybridized to improve DE's search capability. Merging of LS mechanism into a global search optimizer for fine tuning the solution is known as Memetic Algorithm (MA) [3, 28, 29]. Two of the most recent MAs are [32] and [28]. In both hybrids, LS mechanism is utilized to complement the global search algorithm. Very recently, Broyden Fletcher Goldfarb Shanno (BFGS) LS was hybridized with an adaptive DE variant, JADE [49], which results in an MA, DEELS [22]. In most of these hybrid designs, LS is incorporated to the best solution only. However, in our proposed hybrids, it is incorporated to the archived elements with population minimization.

In the current paper, we propose a hybrid design which inlaid Davidson-Fletcher-Powell's (DFP) [2, 11] LS strategy into our recently proposed algorithm, RJADE/TA [23] to improve further the performance of RJADE/TA. The key idea is to operate the archived elements by LS and keeping the record of both the perviously explored and new points to minimize the chance of loosing the ever best solution. For this purpose, a different LS mechanism is utilised resulting in a novel hybrid. Firstly, DFP is implemented to the archived information. Secondly, a reflection mechanism is proposed.

This study has the following structure: Basic DE, RJADE/TA and DFP are described in Section 2. Following the related work in Section 3, Section 4 presents the proposed hybrid. Section 5 presents the experimental results and its analysis obtained with the proposed algorithm. Finally, Section 6 concludes this paper and comments on some future research directions.

2. Basic DE, RJADE/TA and DFP

We already described canonical DE and JADE in detail in our previous work [22, 23], here we only review RJADE/TA and DFP method upon which the present work is mainly based.

2.1. **RJADE/TA.** RJADE/TA is an adaptive DE variant. The main idea of which is to archive elite solutions of the population at regular interval of optimization and reflect the overall poor solutions. RJADE/TA introduced the following two aspects in JADE:

2.1.1. *Mirroring the Elite Candidate.* To avoid stagnation and early convergence, RJADE/TA replaces the best solution, $\mathbf{z}^{[k]}_{(best,i)}$ by its mirror image of the search procedure and post it to the second archive $A^{[2]}$. The best individual $\mathbf{z}^{[k]}_{(best,i)}$ (the solution corresponding to minimum function value) of population P is shifted away through the centroid $\mathbf{x}_{c,G}$. The reflected candidate replaces $\mathbf{z}^{[k]}_{(best,i)}$ in the P and the ever best candidate $\mathbf{z}^{[k]}_{(best,i)}$ by itself is migrated to the second archive $A^{[2]}$ as shown in the Algorithm 1.

2.1.2. *Second Archive.* RJADE/TA has two archives, referred as $A^{[1]}$ and $A^{[2]}$ for the sake of convenience. When the optimization reaches the half of available resources (maximum function evaluations), the first archive update of $A^{[2]}$ is made. Afterwards, $A^{[2]}$ is updated adaptively with a continuing intermission of generations, refer to Algorithm 1. The characteristics of both archives can be listed as:

(1) The optimal solutions of the current search are posted to $A^{[2]}$, whereas $A^{[1]}$ stores the very recently found inferior solutions.
(2) The size of $A^{[1]}$ can not exceed population size $N^{[p]}$, if it does, few solution vectors are discarded arbitrarily; however, in $A^{[2]}$ the size might exceeds $N^{[p]}$. As it makes history of all found best individuals, none of the solutions is discarded here.
(3) $A^{[2]}$ records only single solution of the present generation, it might be a parent or an offspring solution. In contrast, $A^{[1]}$ maintains the substandard (more than one) "parents individuals " only.
(4) $A^{[1]}$ is updated at each generation and $A^{[2]}$, initialized as $\emptyset$, is updated adaptively with a gap of κ generations.
(5) The information in $A^{[1]}$ is employed in reproduction afterwards. Whereas in $A^{[2]}$ the stored best solution is reflected with a new solution; which is then sent to the population. Once a candidate solution is posted to $A^{[2]}$, it remains passive during the toll of whole optimization. When the search procedures are terminated,

Algorithm 1 Outlines of RJADE/TA Procedure [23]

Sample size $= N^{[p]}$; $FES =$ Function Evaluations; $\kappa =$ difference between 2 consecutive updates of $A^{[2]}$;
Generate $N^{[p]}$ uniform and random solutions, $\mathbf{z}^{[k]}_{(s_1,i)}, \mathbf{z}^{[k]}_{(s_2,i)}, \ldots, \mathbf{z}^{[k]}_{(s_{N^{[p]}},i)}$ to make the initial population P;
Initialize Archives $A^{[1]} = A^{[2]} = \emptyset$;
Set $\lambda CR = \lambda F = 0.5$; $p = 5\%$; $c = 0.1$;
Set $S_{CR} = S_F = \emptyset$;
Evaluate the members of P;
while $FES < MaxFES$ **do**
    Compute $F_i = rand(\lambda F, 0.1)$;
    Choose $\mathbf{z}^{[p,k]}_{(best)}$ randomly from $100p\%$ population;
    Make random selection of $\mathbf{z}^{[k]}_{(s_1,i)} \neq \mathbf{z}^{[k]}_{(s,i)}$ from P;
    Arbitrarily select $\tilde{\mathbf{z}}^{[k]}_{(s_2,i)} \neq \mathbf{z}^{[k]}_{(s_2,i)}$ from $P \cup A^{[1]}$;
    Produce trial vectors $\mathbf{w}^{[k]}_{(s,i)} = \mathbf{z}^{[k]}_{(s,i)} + F_i(\mathbf{z}^{[p,k]}_{(best)} - \mathbf{z}^{[k]}_{(s,i)}) + F_i(\mathbf{z}^{[k]}_{(s_1,i)} - \tilde{\mathbf{x}}^{[k]}_{(s_2,i)})$;
    for $j = 1$ to n **do**
        if $j < j_{rand}$ or $rand(0,1) < CR_i$ **then**
            $\mathbf{q}^{[k]}_{(j,i)} = \mathbf{w}^{[k]}_{(j,i)}$;
        else
            $\mathbf{q}^{[k]}_{(j,i)} = \mathbf{z}^{[k]}_{(j,i)}$;
        end if
    end for
    Choose the fittest amongst $\{\mathbf{z}^{[k]}_{(s,i)}, \mathbf{q}^{[k]}_{(s,i)}\}$;
    if $\mathbf{q}^{[k]}_{(s,i)}$ is the elite **then**
        $\mathbf{z}^{[k]}_{(s,i)} \to A^{[1]}$, $CR_i \to S_{CR}$, $F_i \to S_F$;
    end if
    if size of $A^{[1]} > N^{[p]}$ **then**
        Remove some solutions of $A^{[1]}$ arbitrarily;
    end if
    Update $A^{[2]}$ as;
    if $k = \kappa$ **then**
        $\mathbf{z}^{[k]}_{(best,i)} \to A^{[2]}$;
        Calculate the center of mass of $P - \mathbf{z}^{[\mathbf{k}]}{}_{(best,i)}$ as $\mathbf{z}^{[k]}_{c,i} = \frac{1}{N^{[p]}-1} \sum_{i=2}^{N^{[p]}} \mathbf{z}^{[k]}_{(c,i)}$
        Produce point of reflection as $\mathbf{z}^{[k]}_{(r,i)} = \mathbf{z}^{[k]}_{(c,i)} + (\mathbf{z}^{[k]}_{(c,i)} - \mathbf{z}^{[k]}_{(best,i)})$
    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
Outcome: The population member corresponding to minimum function value from $PUA^{[2]}$ in the optimization.

then the recoded information contribute towards the selection of the best candidate solution.

In the forthcoming section, we will describe the LS method which will be incorporated in the new algorithm.

2.2. **Davidon Fletcher Powell Method (DFP).** The DFP method is a variable metric method, which was first proposed by Davidon [11] and then modified by Powell and Fletcher. It belongs to the class of gradient dependent LS methods. The DFP procedure is outlined in Algorithm 2. If a right line search (given in line 16) is used in DFP method, it will assure convergence (minimization) [2].

Algorithm 2 Outlines of DFP Algorithm [44]

1: **Initialization**, $error$: demanded accuracy;
2: γ : iterations counter .
3: $\mathbf{z}^{[0]}$: the initial iterate .
4: $\mathbf{H}$: $\mathbf{I}_{n\times n}$ Initialization of Hessian matrix.
5: $j = 0$.
6: **while** $j < \gamma$ **do**
7: find the difference $\mathbf{d}_z = \mathbf{z}^{[j+1]} - \mathbf{z}^{[j]}$;
8: find the difference $\mathbf{d}_g = \nabla f(\mathbf{z}^{[j+1]}) - \nabla f(\mathbf{z}^{[j]})$;
9: **if** $\mathbf{d}_z \neq 0$ and $\mathbf{d}_g \neq 0$ **then**
10: $temp_1 = \mathbf{d}_g \mathbf{H}^{[j]} \mathbf{d}_g^{'}$;
11: $temp_2 = \mathbf{d}_z^{'} \mathbf{d}_g$;
12: Revise the Hessian matrix as:

$$\mathbf{H}^{[j+1]} = \mathbf{H}^{[j]} + \frac{(\mathbf{d}_x^{'}\mathbf{d}_x)}{temp_2} - \frac{(\mathbf{H}^{[j]}\mathbf{d}_g^{'}\mathbf{d}_g\mathbf{H}^{[j]})}{temp_1};$$

13: **end if**
14: locate the search direction $\mathbf{s}^{[j]}$ with the help of Hessian matrix $\mathbf{s}^{[j]} = -\mathbf{H}^{[j]}\nabla f(\mathbf{z}^{[j]})$;
15: calculate α_i by golden section search [44];
16: $\mathbf{z}^{[j+1]} = \mathbf{z}^{[j]} + \alpha^{[j]}\mathbf{s}^{[j]}$;
17: **end while**
Result: $\mathbf{z}^{[j+1]}$ is the output of the algorithm.

3. RELATED WORK

To heal the above mentioned weaknesses of DE, many researchers merged various LS techniques in DE. Nelder Mead Simplex (NMS), a LS method was hybridized with DE [1] to improve the local exploitation of DE. Recently, two new LS strategies were proposed and hybridized iteratively with DE in [28, 32]. These hybrid designs showed performance improvement over the algorithms in comparison. Two LS strategies, trigonometric and interpolated LS were inserted in DE to enhance it's poor exploration. Also, two LS techniques were merged in DE along with a restart to improve its global exploration [24]. This algorithm was statistically sound, as the obtained results were better than other algorithms. Furthermore, alopax dependent crossover LS was merged in DE [27] to improve its diversity of population. In another experiment, DE's slow convergence was enhanced by combing orthogonal design (OD) LS [9] with it. To avert local optima in DE random LS was hybridized [37] with it. On the other hand, some researchers borrowed DE's mutation and crossover in traditional LS methods, for example see [25,48]. To the best of our knowledge, none of the reviewed algorithms in this section hybridized DFP local search into DE framework. Further, the proposed work here maintains two archives, first one stores inferior solutions and second keeps information of elite 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 following advantages. Firstly, the second archive keeps complete information of the solutions before and after LS. This way any good solution is not lost. Secondly, the migrated solutions to second archive are regenerated by new reflected potential solutions which maintain the diversity of the population. This diversity is very important for the search to carry on and as a result it does not converge prematurely.

4. RJADE/TA INTEGRATED WITH LOCAL SEARCH

A good number of hybrid algorithms of LS with global search algorithms have been proposed in literature. However, hybridizing two or more technique together to perform efficiently is still a major issue. In this paper, we hybridize a LS method DFP with a global search method RJADE/TA. DFP is integrated in a different way into RJADE/TA, resulting in a different variant of RJADE/TA. We shall refer this as RJADE/TA-LS through out this work.

4.1. **RJADE/TA-LS.** The initial population is explored by RJADE/TA till 50% of the function evaluations. After regular mutation, crossover, selection and $A^{[1]}$ updates in RJADE/TA, as shown in Algorithm 1, the population is sorted and the current best solution $\mathbf{z}^{[k]}_{(best,i)}$ is migrated to $A^{[2]}$, where DFP LS will be incorporated to the migrated elements. After implementation of DFP, $\mathbf{z}^{[k]}_{(new,i)}$ is produced from old migrated solution. Here, the previously explored and the new solution as a result of implementation of DFP will be posted to archive $A^{[2]}$ (see Algorithm 3). Unlike our perviously proposed archive $A^{[2]}$ in RJADE/TA [23], where the archive was keeping the records of only best solutions and with no LS, $A^{[2]}$ in this experiment will maintain information of both solutions.

Due to migration of best solution to $A^{[2]}$, the current population size will be decreased by 1 at each $A^{[2]}$ update in RJADE/TA-LS. Thus to maintain fixed population, a replacement is newly generated and added to the population as follows. Infect a mirror image $\mathbf{z}^{[k]}_{(r,i)}$ of the migrated solution $\mathbf{z}^{[k]}_{(best,i)}$ is computed through the centroid of the $N^{[p]}-1$ population

$$\mathbf{z}^{[k]}_{c,i} = (N^{[p]}-1)^{-1}\sum_{i=2}^{N^{[p]}} \mathbf{z}^{[k]}_{(c,i)}. \tag{4. 1}$$

Hence ignoring the shifted solution by equation shown in line 7 of Algorithm 3. The current population is updated with this new candidate solution and iteration is terminated.

$$\mathbf{z}^{[k]}_{(r,i)} = (\mathbf{z}^{[k]}_{(c,i)} - \mathbf{z}^{[k]}_{(best,i)}) + \mathbf{z}^{[k]}_{(c,i)}, \tag{4. 2}$$

where $\mathbf{z}^{[k]}_{(r,i)}$ is the reflection [34] of $\mathbf{z}^{[k]}_{(best,i)}$.

The archive $A^{[2]}$ is updated after regular intervals of κ generations (20 here). The migrated solutions and those explored by DFP remain there during the entire optimization.

When the whole evolution is completed the overall best candidate is selected from $PUA^{[2]}$. The novelty of RJADE/TA-LS is that it employs the LS to the archived solutions only, different from all the hybrids designs reviewed in section 3 above.

Algorithm 3 RJADE/TA-LS

Update $A^{[2]}$ as;
if $k = \kappa$ **then**
  $\mathbf{z}^{[k]}_{(best,i)} \rightarrow A^{[2]}$;
  Apply DFP to $\mathbf{z}^{[k]}_{(best,i)}$ to prduce $\mathbf{z}^{[k]}_{(new,i)}$;
  $\mathbf{z}^{[k]}_{(new,i)} \rightarrow A^{[2]}$ and $\mathbf{z}^{[k]}_{(best,i)} \rightarrow A^{[2]}$;
  Calculate the centroid of $P - \mathbf{z^{[k]}}_{(best,i)}$ as $\mathbf{z}^{[k]}_{c,i} = \frac{1}{N^{[p]}-1}\sum_{i=2}^{N^{[p]}} \mathbf{z}^{[k]}_{(c,i)}$
  Produce point of reflection as $\mathbf{z}^{[k]}_{(r,i)} = \mathbf{z}^{[k]}_{(c,i)} + (\mathbf{z}^{[k]}_{(c,i)} - \mathbf{z}^{[k]}_{(best,i)})$
end if
Update the population with $\mathbf{z}^{[k]}_{(r,i)}$.
Terminate the iteration.

5. EXPERIMENTATION

In this section, first we discuss the experimental setup, and then RJADE/TA-LS is compared with global problem solvers available in the litrature.

5.1. **Experimental Setup.** Extensive experiments are conducted on 28 functions of CEC 2013 test problems [26], We will refer these as CP1, CP2, and so on in this experiment. We follow the parameter settings suggested for CEC 13 test suit [26]. The population size $N^{[p]}$ is kept 100 and problem dimension, n is considered 10 here. The demanded functions evaluations are $10000 \times n$. The stopping criteria are that if the maximum function evaluations are met or the difference between the Means of function errors is less than 10^{-8} as suggested in literature [23, 26].

5.2. **Comparison of RJADE/TA-LS with Global Problem Solvers in Literature.** Table 1 shows the experimental statistics (i.e., best, mean, median, worst and standard deviation) obtained by RJADE/TA-LS in 51 runs on 28 functions with dimensions $n = 10$ of the CEC 2013 test functions. Whereas Table 2 presents the "Mean values" of function errors ($f(\mathbf{z}^{[k]}_{(best,i)}) - f(\mathbf{z}^{[k]}_{(*,i)})$) obtained by proposed RJADE/TA-LS, where $f(\mathbf{z}^{[k]}_{(*,i)})$ is the known value to reach of a particular problem. The comparison is made with adaptive DE variants jDE, jDEsoo [6] and jDErpo [7] proposed in the literature. These algorithm were presented at special session of CEC 2013 competition. Moreover, we further compare the new algorithm against our previously proposed RJADE/TA [23] in the same table.

In Table 2, the **+** shows that the specific algorithm wins against our RJADE/TA-LS, the **-** indicates that the particular algorithm loses against our algorithm and **=** reveals that both the algorithms obtained same statistics. The outstanding performance of RJADE/TA-LS is clearly visible from Table 2, where many negative **-** signs made this fact evident. It is clearly visible that RJADE/TA-LS achieved significantly better results than jDE and

TABLE 1. EXPERIMENTAL STATISTICS OF RJADE/TA-LS ON 28 TEST FUNCTIONS OVER 51 RUNS WITH DIMENSION $n = 10$.

Test Problem	Best	Worst	Median	Mean	Std Dev
CP1	$0.0000e+0$	$0.0000e+0$	$0.0000e+0$	$0.0000e+0$	$0.0000e+0$
CP2	$0.0000e+0$	$0.0000e+0$	$0.0000e+0$	$0.0000e+0$	$0.0000e+0$
CP3	$2.2737e-13$	$3.3330e+03$	$1.1399e+02$	$2.5750e+02$	$5.0783E+02$
CP4	$0.0000e+0$	$2.0151e+003$	$0.0000e+0$	$3.9511e+001$	$2.8216e+02$
CP5	$0.0000e+0$	$0.0000e+0$	$0.0000e+0$	$0.0000e+0$	$0.0000e+0$
CP6	$0.0000e+0$	$9.8124e+0$	$9.8124e+0$	$6.9264e+0$	$4.5155e+0$
CP7	$7.7216e-04$	$3.3426e+0$	$5.0649e-02$	$2.3707e-01$	$5.2201e-01$
CP8	$2.0153e+01$	$2.0532e+01$	$2.0361e+01$	$2.0352e+01$	$7.9096E-02$
CP9	$2.7588e+0$	$5.6920e+0$	$4.5433e+0$	$4.4888e+0$	$6.0648e-01$
CP10	$1.0573e-11$	$6.5443e-02$	$3.2581e-02$	$3.2488e-02$	$1.4326e-02$
CP11	$5.6843e-14$	$9.0949e-13$	$1.7053e-13$	$2.2737e-13$	$1.7576e-13$
CP12	$3.0935e+0$	$1.0292e+01$	$7.1592e+0$	$6.8613e+0$	$1.5339e+0$
CP13	$3.1153e+0$	$1.4565E+e01$	$7.8074e+0$	$7.9039e+0$	$2.5344e+0$
CP14	$1.7909E-004$	$6.6399E-002$	$1.8731E-003$	$7.3105E-003$	$1.6904E-002$
CP15	$2.0294E+002$	$8.8731E+002$	$6.8075E+002$	$6.6733E+002$	$1.4778E+002$
CP16	$6.0593E-001$	$1.6032E+0$	$1.0913E+0$	$1.0855E+0$	$2.1696E-001$
CP17	$1.0122E+001$	$1.0122E+001$	$1.0122E+001$	$1.0122E+001$	$4.4260E-006$
CP18	$1.6924E+001$	$3.2454E+001$	$2.2826E+001$	$2.2884E+001$	$3.4018E+0$
CP19	$3.1028E-001$	$5.7275E-001$	$4.6508E-001$	$4.4752E-001$	$6.1314E-002$
CP20	$1.9544E+0$	$3.4657E+0$	$2.5479E+0$	$2.5707E+0$	$3.4632E-001$
CP21	$2.0001E+002$	$4.0019E+002$	$4.0019E+002$	$3.9627E+002$	$2.8031E+001$
CP22	$1.0508E-002$	$1.2329E+002$	$1.4179E+001$	$2.0589E+001$	$2.3414E+001$
CP23	$2.0008E+002$	$1.1802E+003$	$6.6229E+002$	$6.7549E+002$	$1.9405E+002$
CP24	$1.2132E+002$	$2.1316E+002$	$2.0043E+002$	$1.9809E+002$	$1.9196E+001$
CP25	$1.7553E+002$	$2.1085E+002$	$2.0063E+002$	$2.0190E+002$	$4.7984E+0$
CP26	$1.0540E+002$	$3.1497E+002$	$1.1413E+002$	$1.3596E+002$	$4.6023E+001$
CP27	$3.0001E+002$	$3.0223E+002$	$3.0022E+002$	$3.0033E+002$	$3.7837E-001$
CP28	$3.0000E+002$	$3.0000E+002$	$3.0000E+002$	$3.0000E+002$	$5.1662E-012$

jDEsoo algorithms on 14 and 15 out of 28 problems, respectively. In contrast, jDE got superior statistics on 8 problems against RJADE/TA-LS, which can be seen from Table 2.

In comparison with jDErpo, both the algorithms solved 12/28 problems, while on 4 problems they are equivalent. While in comparison with RJADE/TA, RJADE/TA-LS performs superior than RJADE/TA. It is interesting to note that on complex problems, CP21-CP28, RJADE/TA-LS performed better on majority of problems against most of the algorithms. This is surly due to the integration of LS, which is very good in fine tuning the solution.

The proposed work here incorporates a reflection mechanism which saves the search from going into bad direction. RJADE/TA-LS dependency on LS for fine tuning the solution could be the reason of its getting good solutions, while the algorithms in comparison which are not using LS did not get better solutions. Moreover, its archive records the shifted solutions and newly generated ones, so it has a very low chance to loose good solutions or to stuck into local optima. In general, it keeps balance between exploration and exploitation.

TABLE 2. Comparison of RJADE/TA-LS Obtained Results with Other Global Optimizers.

Prob	jDE	jDEsoo	jDErpo	RJADE/TA	RJADE/TA-LS
CP1	$0.0000e+0$=	$0.0000e+0$=	$0.0000e+0$=	$0.0000e+0$=	$0.0000e+0$
CP2	$7.6534e-05$-	$1.7180e+03$-	$0.0000e+0$=	$0.0000e+0$=	$0.0000e+00$
CP3	$1.3797e+0$+	$1.6071e+0$+	$3.7193e-05$+	$1.2108e+02$+	$2.5750e+02$
CP4	$3.6639e-08$+	$1.2429e-01$+	$0.0000e+0$+	$1.1591e+02$-	$3.9511e+01$
CP5	$0.0000e+0$=	$0.0000e+0$=	$0.0000e+0$=	$0.0000e+0$=	$0.0000e+00$
CP6	$8.6581e+0$-	$8.4982e+04$-	$5.3872e+0$+	$7.8884e+0$-	$6.9264e+00$
CP7	$2.7229e-03$+	$9.4791e-01$-	$1.6463e-03$+	$1.5927e-01$+	$2.3707e-01$
CP8	$2.0351e+01$-	$2.0348e+01$+	$2.0343e+01$+	$2.0366e+01$-	$2.0342e+01$
CP9	$2.6082e+0$+	$2.7464e+0$+	$6.4768e-01$+	$4.4593e+0$+	$4.4888e+00$
CP10	$4.5263e-02$-	$7.0960e-02$-	$6.4469e-02$-	$3.5342e-02$-	$3.2488e-02$
CP11	$0.0000e+0$=	$0.0000e+0$=	$0.0000e+0$=	$0.0000e+0$=	$2.2737e-13$
CP12	$1.2304e+01$-	$6.1144e+0$+	$1.3410e+01$-	$7.7246e+0$-	$6.8613e+00$
CP13	$1.3409e+01$-	$7.8102e+0$+	$1.4381e+01$-	$6.7571e+0$+	$7.9039e+00$
CP14	$0.0000e+0$+	$5.0208e-02$-	$1.9367e+01$-	$1.1994e-02$-	$7.3105e-003$
CP15	$1.1650e+03$-	$8.4017e+02$-	$1.1778e+03$-	$6.6660e+02$+	$6.6733e+02$
CP16	$1.0715e+0$+	$1.0991e+0$-	$1.0598e+0$+	$1.1336e+0$-	$1.0855e+00$
CP17	$1.0122e+01$=	$9.9240e+0$+	$1.0997e+01$-	$1.0122e+01$=	$1.0122e+01$
CP18	$3.2862e+01$-	$2.7716e+01$-	$3.2577e+01$-	$2.2715e+01$+	$2.2884e+01$
CP19	$4.3817e-01$+	$3.1993e-01$+	$7.4560e-01$-	$4.4224e-01$+	$4.4752e-01$
CP20	$3.0270e+0$-	$2.7178e+0$-	$2.5460e+0$+	$2.5317e+0$+	$2.5707e+00$
CP21	$3.7272e+02$+	$3.5113e+02$+	$3.7272e+02$+	$3.9627e+02$=	$3.9627e+02$
CP22	$7.9231e+01$-	$9.1879e+01$-	$9.7978e+01$-	$2.7022e+01$-	$2.0589e+01$
CP23	$1.1134e+03$-	$8.1116e+02$-	$1.1507e+03$-	$7.0015e+02$-	$6.7549e+02$
CP24	$2.0580e+02$-	$2.0851e+02$-	$1.8865e+02$+	$2.0217e+02$-	$1.9809e+02$
CP25	$2.0471e+02$-	$2.0955e+02$-	$1.9885e+02$+	$2.0314e+02$-	$2.0190e+02$
CP26	$1.8491e+02$-	$1.9301e+02$-	$1.1732e+02$+	$1.2670e+02$+	$1.3596e+02$
CP27	$4.7470e+02$-	$4.9412e+02$-	$3.0000e+02$+	$3.0351e+02$-	$3.0033e+02$
CP28	$2.9216e+02$+	$2.8824e+02$+	$2.9608e+02$+	$2.8824e+02$+	$3.0000e+02$
-	14	15	12	12	
+	8	10	12	10	
=	6	3	4	6	

The high performance of RJADE/TA-LS on last CP21-CP28 problems, being the most difficult ones in the CEC 13 test suit as these are formulated by combining two or several other complex problems [43], indicates that the new technique is more suited to such hard problems.

Further boxplots are plotted to study the behaviour of RJADE/TA-LS against RJADE/TA. The means obtained by both algorithms in 20 runs are represented by the boxplots. Since we considered minimization problems only, so if the boxplot in figure 1 is lower than the other, then it is considered as best. Thus, the boxplot of CP8 and CP10 in figure 1(a) and (b) reveals that means obtained by RJADE/TA-LS are lower than RJADE/TA, so RJADE/TA-LS is better than RJADE/TA. The boxplots for CP18, CP20 and CP24 are given in figure

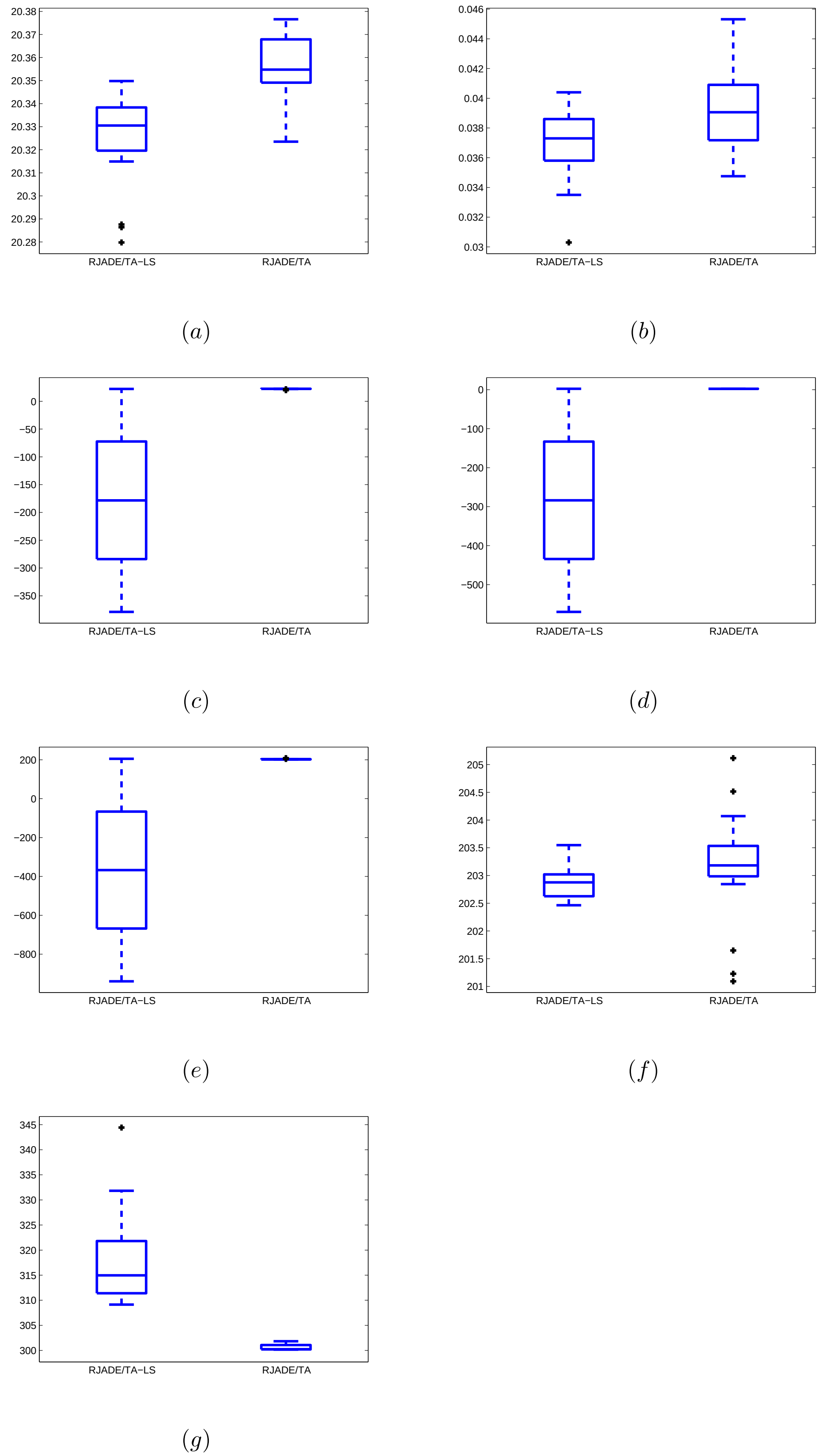

FIGURE 1. Boxplots of RJADE/TA-LS and RJADE/TA

1(c), (d) and (e). In all of these functions, RJADE/TA found same mean value in each run, so the boxplot is given as straight line, which means it was stuck in local optima. While RJADE/TA-LS solutions are lower than the horizontal line and are distinct in most cases, which are given in the same figure 1(c), (d) and (e). Which concludes that the new algorithm avoids to stuck in local optima.

In case of CP25, the boxplot in figure 1 (f) is again lower and means are spread in small area as compared to boxplot of RJADE/TA, Which has higher means and some data is out side the plot, the black spots in the graph. Considering CP27, RJADE/TA performed better than RJADE/TA-LS, as its box is lower than ours' one, and be seen in figure 1 (g).

6. CONCLUSIONS

In this study, we proposed a memetic algorithm by combining a DE version RJADE/TA with DFP, LS technique for continuous unconstrained optimization. The main idea behind this work is to bring together the local tuning of LS and global exploration of RJADE/TA to form new variant which should perform efficiently in both local/global regions. Hence, the shifting of elite solutions to archive and their exploitation by DFP was proposed. A diversity maintenance mechanism was also established to prevent premature convergence of the algorithm.

The proposed variant is tested on comparatively recent and hard test problems from CEC 2013, and the obtained experimental results are compared with well established algorithms. The results demonstrated that the new version is efficient than various algorithms in companions on majority of tested problems. It was also identified that this algorithm is more suitable for hard problems.

In future, the current work with some modifications will be extended to solve constrained optimization problems. Secondly, some other LS algorithms will be studied in this approach. Thirdly, instead of adopting DE, other global optimizers will be used. We also intend to propose other archive techniques.

REFERENCES

[1] M. Ali, M. Pant and A. Abraham, *Simplex Differential Evolution*, Acta Polytechnica Hungarica **6**, No. 5 (2009) 95-115.

[2] A. Antoniou and W. -S. Lu, *Practical optimization: algorithms and engineering applications*, Springer Science & Business Media 2007.

[3] M. Asim, W. K. Mashwani, M. A. Jan and J. Ali, *Derivative Based Hybrid Genetic Algorithm: A Preliminary Experimental Results*, Punjab Univ. j. math. **49**, No. 2 (2017) 89-99.

[4] N. H. Awad, M. Z. Ali, R. Mallipeddi and P. N. Suganthan *An improved differential evolution algorithm using efficient adapted surrogate model for numerical optimization*, Information Sciences, Elsevier **451**, (2018) 326-347.

[5] L. L. Betzig, *Despotism, social evolution, and differential reproduction*, Routledge, 2018.

[6] J. Brest, B. Boskovic, A. Zamuda, I. Fister and E. M. Montes, *Real Parameter Single Objective Optimization using self-adaptive differential evolution algorithm with more strategies*, IEEE Congress on Evolutionary Computation (CEC), IEEE (2013) 377-383.

[7] J. Brest, A. Zamuda, I. Fister and B. Boskovic, *Some Improvements of the Self-Adaptive jDE Algorithm*, 2014 IEEE Symposium on Differential Evolution (SDE), IEEE (2014) 1-8.

[8] R. D. A. -Dabbagh, F. Neri, N. Idris and M. S. Baba, *Algorithm Design Issues in Adaptive Differential Evolution: Review and taxonomy*, Elsevier, 2018.

[9] Z. Dai, A. Zhou, G. Zhang and S. Jiang, *A differential evolution with an orthogonal local search*, 2013 IEEE Congress on Evolutionary Computation, IEEE (2013) 2329-2336.

[10] S. Das, S. S. Mullick and P. N. Suganthan, *Recent advances in differential evolution An updated survey*, Swarm and Evolutionary Computation, Elsevier (2016) 1-62.
[11] W. C. Davidon, *Variable metric method for minimization*, SIAM Journal on Optimization, SIAM **1**, No. 1 (1991) 1-17.
[12] M. Dorigo, *Ant colony optimization*, Scholarpedia **2**, No. 3 (2007) 1461.
[13] M. Dorigo and M. Birattari, *Ant colony optimization*, Encyclopedia of machine learning (2011) 36-39.
[14] R. C. Eberhart and J. Kennedy, *A New Optimizer using Particle Swarm Theory*, 6th International Symposium on Micromachine and Human Science, Nagoya, Japan (1995) 39-43.
[15] R. C. Eberhart and Y. Shi, *Guest Editorial: Special Issue on Particle Swarm Optimization*, IEEE transactions on Evolutionary Computation **8**, No. 3 (2004) 201-203.
[16] A. Engelbrecht and G. Pampara, *Binary Differential Evolution Strategies*, Proceedings of IEEE Congress on Evolutionary Computation (CEC 2007) 1942-1947.
[17] Shu-Kai S. Fan and E. Zahara, *A hybrid Simplex Search and Partical Swarm optimization for unconstrained optimization*, European Journal of operational Research **811**, (2007) 527-548.
[18] R. Fletcher, *Practical Methods of Optimization*, Second ed., Wiley, 1987.
[19] V. Gazi and K. M. Passino, *Bacteria foraging optimization*, Swarm Stability and Optimization, Springer, (2011), 233-249.
[20] J. Kennedy and R. C. Eberhart, *Particle Swarm Optimization*, IEEE International Conference on Neural Networks **4**, (1995) 1942-1948.
[21] J. Kennedy and R. C. Eberhart, *A Discrete Binary Version of the Partical Swarm Algorithm*, Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics (1997) 4104-4109.
[22] R. A. Khanum, M. A. Jan, N. M. Tairan and W. K. Mashwani, *Hybridization of Adaptive Differential Evolution with an Expensive Local Search Method*, Journal of Optimization, Hindawi **2016**, (2016) 1-14.
[23] R. A. Khanum, N. M. Tairan, M. A. Jan, W. K. Mashwani and A. Salhi, *Reflected Adaptive Differential Evolution with Two External Archives for Large-Scale Global Optimization*, International Journal of Advanced Computer Science and Applications, Sciences and Information **7**, No. 2 (2016) 675-683.
[24] R. A. Khanum, M. A. Jan, W. K. Mashwani, N. M. Tairan, H. U. Khan and H. Shah, *On the hybridization of global and local search methods*, Journal of Intelligent & Fuzzy Systems, IOS Press **35**, No. 32 (2018) 3451-3464.
[25] R. A. Khanum, I. Zari, M. A. Jan and W. K. Mashwani, *Reproductive NELDER-MEAD ALGORITHMS FOR UNCONSTRAINED Optimization Problems*, Science International **28**, No. 1 (2016) 19-25.
[26] J. J. Liang, B. Y. Qu, P. N. Suganthan and A. G. Hernández-Díaz, *Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization*, IEEE 2013.
[27] M. Leon and N. Xiong, *A New Differential Evolution Algorithm with Alopex-Based Local Search*, International Conference on Artificial Intelligence and Soft Computing, Springer (2016) 420-431.
[28] M. Leon and N. Xiong, *Eager random search for differential evolution in continuous optimization*, Portuguese Conference on Artificial Intelligence, Springer (2015) 286-291.
[29] M. Lozano, F. Herrera, N. Krasnogor and D. Molina, *Real-Coded Memetic Algorithms with Crossover Hill-Climbing*, Evolutionary Computation **12**, No. 3 (2004) 273-302.
[30] M. S. Maucec, J. Brest, B. Boskovic and Z. Kacic, *Improved Differential Evolution for Large-Scale Black-Box Optimization*, IEEE Access, IEEE **6**, (2018) 29516-29531.
[31] Z. Meng, J. -S. Pan and W. -M. Zheng, *Differential evolution utilizing a handful top superior individuals with bionic bi-population structure for the enhancement of optimization performance*, Enterprise Information Systems, Taylor & Francis (2018) 1-22.
[32] D. Molina and F. Herrera, *Iterative hybridization of DE with local search for the CEC'2015 special session on large scale global optimization*, 2015 IEEE Congress on Evolutionary Computation (CEC), IEEE, (2015), 1974-1978.
[33] P. Moscato, *On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms*, Caltech Concurrent Computation Program, C3P Report, **826**, (1989).
[34] J. A. Nelder and R. Mead, *A simplex method for function minimization*, Comput. J. Br. Computer. Soc. **7**, No. 4 (1965) 308-313.
[35] K. Nouri, H. Ranjbar and L. Torkzadeh, *Two High Order Iterative Methods for Roots of Nonlinear Equations*, Punjab Univ. j. math. **51**, No. 3 (2019) 47-59.

[36] K. R. Opara and J. Arabas, *Differential Evolution: A survey of theoretical analyses*, Swarm and Evolutionary Computation, Elsevier 2018.
[37] M. L. Ortiz and N. Xiong, *Using random local search helps in avoiding local optimum in differential evolution*, Proc. IASTED (2014) 413-420.
[38] Passino and M. Kevin, *Bacterial foraging optimization*, International Journal of Swarm Intelligence Research (IJSIR) IGI Global, **1**, No. 1 (2010) 1-16.
[39] K. V. Price, *Eliminating Drift Bias from the Differential Evolution Algorithm*, Advances in Differential Evolution, Springer **143**, (2008) 33-88.
[40] A. Salami and M. A. Nada, *System evolving using ant colony optimization algorithm*, Journal of Computer Science **5**, No. 5 (2009) 380-387.
[41] R. Storn and K. V. Price, *Differential Evolution-A simple and efficient heuristic for global optimization over continuous spaces*, Journal of Global Optimization **11**, No. 4 (1997) 341-359.
[42] R. Storn, *Differential Evolution (DE) Research-Trends and open questions*, Advances in Differential Evolution, Springer, (2008) 1-31.
[43] R. Tanabe and A. Fukunaga, *Success-history based parameter adaptation for differential evolution*, 2013 IEEE Congress on Evolutionary Computation (CEC), IEEE (2013) 71-78.
[44] P. Venkataraman, *Applied Optimization with Matlab programming*, John Wiley and sons, New York 2002.
[45] X.-S. Yang and S. Deb, *Cuckoo search via Lévy flights*, World Congress on 2009 Nature & Biologically Inspired Computing (NaBIC), IEEE, (2009) 210-214.
[46] X.-S. Yang and S. Deb, *Engineering optimisation by cuckoo search*, arXiv preprint arXiv:1005.2908, 2010.
[47] S. Y. Yuen and C. K. Chow, *A Genetic Algorithm that Adaptively Mutates and Never Revisits*, IEEE Transactions on Evolutionary Computation **13**, No. 2 (2009) 454-472.
[48] I. Zari, R. A. Khanum, M. A. Jan and W. K. Mashwani, *Hybrid NELDER-MEAD ALGORITHMS FOR NONLINEAR NUMERICAL Optimization*, Science International **28**, No. 1 (2016) 153-159.
[49] J. Zhang and A. C. Sanderson, *JADE: adaptive differential evolution with optional external archive*, IEEE Transactions on Evolutionary Computation, IEEE, **13**, No. 5 (2009) 945-958.