Reusing the Past Difference Vectors in Differential Evolution—A Simple But Significant Improvement

Arka Ghosh, Swagatam Das, Asit Kr. Das, and Liang Gao

Abstract—**Differential evolution (DE) has established itself as a simple but efficient population-based, nonconvex optimization algorithm for continuous search spaces. Unlike the conventional real-coded genetic algorithms (GAs) and evolution strategies (ESs), DE uses a mandatory self-referential mutation for its population members, each of which are perturbed with the scaled difference(s) of the individuals from the current generation (iteration). These difference vectors determine the direction of the search moves for the individuals. However, unlike the better individuals, they are not retained in the elitist evolution cycle of DE. In this paper, we show that by archiving the most promising difference vectors from past generations and then by reusing them for generating offspring in the subsequent generations, we can strikingly improve the performance of DE. This strategy can be integrated with any classical or advanced DE variant with no serious overhead in time or space complexity. We demonstrate that when combined with the DE-based winners of the IEEE Congress on Evolutionary Computation (CEC) 2013, 2014, and 2017 competitions on real parameter optimization, the simple reuse strategy leads to a statistically significant performance improvement in the majority of test cases. We further showcase the efficacy of our proposal on a practical optimization problem concerning the design of circular antenna arrays with a prespecified radiation pattern.**

Index Terms—**Benchmarking, circular antenna array design, continuous optimization, difference vectors, differential evolution (DE), mutation.**

I. Introduction

SINCE its inception in 1995 [1] as a nonconvex, black-box optimization algorithm for continuous parametric spaces, differential evolution (DE) emerged as a prominent member of the evolutionary computing (EC) community due to its simplicity, only a few tunable control parameters, and yet a remarkable efficiency over diverse optimization scenarios both in terms of speed and accuracy [2], [3]. Unlike evolution strategies (ESs), DE is yet to possess a firm theoretical backbone. However, its remarkable performance has been evident from the high ranks obtained by its variants in various IEEE Congress on Evolutionary Computation (CEC) competitions on continuous optimization in complex scenarios (e.g., constrained, multimodal, dynamic, multiobjective, etc.). It is this consistent and impressive performance which enabled DE to stand the test of time in the EC community and beyond. More than a decade's research efforts have gone to further improve the scalability, accuracy, convergence speed, and robustness of DE on bound-constrained, single-objective optimization problems and the quests are very much alive to date. These attempts have been made by following a few major research directions, such as success-history-based parameter adaptation (like self-adaptive DE (SaDE) [4], JADE [5], and SHADE [6]); improved mutation and crossover strategies (like Pro-DE [7], MDE-pBX [8], DE with eigenvector-based crossover [9], and DE with multiple exponential crossover [10]); combining various offspring generation strategies (CoDE [11], EPSDE [12], and DE with multiple variant coordination [13]); and improved parameter control (see [13]–[16]).

In this paper, we hypothesize that preserving the successful directions of perturbation in the differential mutation of DE can be very effective in black-box optimization scenarios. Probabilistic reuse of the previously successful search directions can discover more promising solutions in future generations with a high probability, while still preserving considerable population diversity. These directions are essentially given by the difference vectors formed by subtracting one current individual from the other in the vector sense. Although DE is elitist in nature due to its greedy selection strategy, it cannot preserve the most promising difference vectors over generations due to the random choice of the vectors forming the differences. In this paper, we propose a simple archiving and reusing strategy for the promising difference vectors from earlier generations. Instead of any individual, we store only the difference vectors and reuse them in a controlled manner to generate offspring in the subsequent generations. The resulting framework can be integrated with any DE variant irrespective of the nature of the mutation, crossover, and parameter adaptation used. Our experiments conducted on the IEEE CEC 2013, 2014, and 2017 test suites, along with the circular antenna array design problem,

Manuscript received August 10, 2018; revised December 10, 2018 and May 30, 2019; accepted June 3, 2019. The work of S. Das and A. K. Das was supported by the GPU Grant from NVIDIA Corporation. The work of L. Gao was supported by the National Natural Science Foundation for Distinguished Young Scholars of China under Grant 51825502. This paper was recommended by Associate Editor F. Herrera. *(Corresponding author: Swagatam Das.)*

A. Ghosh and A. K. Das are with the Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology Shibpur, Howrah 711101, India (e-mail: 9007900477a@gmail.com; akdas@cs.iiests.ac.in).

S. Das is with the Electronics and Communication Sciences Unit, Indian Statistical Institute Kolkata, Kolkata 700108, India (e-mail: swagatam.das@isical.ac.in).

L. Gao is with the State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: gaoliang@mail.hust.edu.cn).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2019.2921602

indicate that the reuse of successful individuals can improve the performance of classical as well as state-of-the-art DE variants in a statistically meaningful fashion, without incurring any significant computational overhead.

The rest of this paper is organized in the following way. Section II provides a brief overview of the conventional DE algorithm and reviews some of the pertinent existing approaches to improve DE. Section III describes our proposed method in sufficient detail. Experimental results are presented and duly discussed in Section IV. Finally, Section V concludes this paper and unearths some important future research avenues.

II. Background

We begin with a brief outline of the DE algorithm and then discuss some of the related approaches to improve DE by preserving promising parameter values and directional information.

A. Conventional DE

Initialization, mutation, crossover, and selection—these four steps are present in a DE algorithm, among which only the last three steps are repeated in the succeeding generations of DE until some predefined criterion [such as an exhaustion of a maximum number of fitness evaluations (FEs)] is satisfied.

DE starts with the standard uniformly random initialization of a population of Np real-valued solution vectors, each dimension of them representing one of the D decision variables of the optimization problem at hand. Initial values of each variable are drawn uniform at random from a prespecified interval between some lower and upper bounds.

Subsequently, in every G^{th} generation, a parent or target vector $\overrightarrow{X}_{i,G}$ is perturbed to create a corresponding mutant or donor vector $\overrightarrow{V}_{i,G}$ in the mutation step. Out of the many strategies proposed so far, only the two most commonly used mutation strategies are shown below

$$\text{DE/rand/1: } \overrightarrow{V}_{i,G} = \overrightarrow{X}_{r_1,G} + F.\overrightarrow{\Delta}_{i,G} \tag{1a}$$

$$\text{DE/best/1: } \overrightarrow{V}_{i,G} = \overrightarrow{X}_{\text{best},G} + F.\overrightarrow{\Delta}_{i,G} \tag{1b}$$

where $\overrightarrow{V}_{i,G}$ is the resulting mutant vector and $\overrightarrow{\Delta}_{i,G}$ is a difference vector of the form $(\overrightarrow{X}_{r_2,G} - \overrightarrow{X}_{r_3,G})$. The indices $r_1, r_2, r_3 \in \{1, 2, \ldots, Np\}$ such that $r_1 \neq r_2 \neq r_3 \neq i$ and these are randomly selected for each mutant vector. $\overrightarrow{X}_{\text{best},G}$ is the best solution vector of the current generation G, and $F(\in (0.4, 2])$ is a positive control parameter for scaling the difference vector(s).

Next, a crossover operation (which may be binomial, exponential, or arithmetic) takes place to create the final offspring vector $\overrightarrow{U}_{i,G}$. We elaborate only the very commonly used binomial crossover process as the same is used in our experiments and in the DE variants that we will compare against. In the binomial crossover scheme, each component of $\overrightarrow{U}_{i,G}$ can either come from $\overrightarrow{X}_{i,G}$ or the mutant vector $\overrightarrow{V}_{i,G}$ in the following way:

$$u_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{if } \text{rand}_{i,j} \leq Cr \text{ or } j = j_r \\ x_{j,i,g}, & \text{otherwise} \end{cases} \tag{2}$$

where $\text{rand}_{i,j}$ is a uniformly distributed random number from the interval [0, 1] and it is instantiated a new for every j^{th} component of the i^{th} population member. Cr is the crossover rate that controls the proportion of the components of $\overrightarrow{X}_{i,G}$ and $\overrightarrow{V}_{i,G}$ in $\overrightarrow{U}_{i,G}$. j_r is an index chosen randomly from $\{1, 2, \ldots, D\}$ and it ensures that the offspring gets at least one component from the mutant vector. Finally, a one-to-one competition-based selection process between the target and the trial vector is performed in the following way:

$$\overrightarrow{X}_{i,G+1} = \begin{cases} \overrightarrow{U}_{i,G}, & \text{if } f\left(\overrightarrow{U}_{i,G}\right) \leq f\left(\overrightarrow{X}_{i,G}\right) \\ \overrightarrow{X}_{i,G} & \text{otherwise} \end{cases} \tag{3}$$

where $f(.)$ is the cost or objective function to be minimized.

B. Previous DE Approaches Based on Preservation of Promising Parameters and Directions: Overview

DE has three control parameters: 1) the scale factor (F); 2) the crossover rate (Cr); and 3) the population size (Np). Of these, the first two affect the performance of the algorithm more significantly by governing the perturbation step-size and the severity of crossover. Most of the success history-based adaptation schemes that have so far been very successfully used in DE, try to either learn a parameterized distribution of the successful F and Cr or directly archive the successful values of these parameters (see [4]–[6]). Success, in this context, implies generation of a better offspring by using the given values of F and Cr. Similarly, methods like SaDE [4] invoke the same offspring generation strategy (out of a few, forming a pool) again and again if it continues to generate better individuals. Zhang and Luo [17] used a directional derivative based on the fitness improvement along the direction from a base vector to the current best individual to improve DE mutation. This method did not store any difference vector and required an extra FE, per mutation step.

Guo *et al.* [18] proposed a successful parent selection (SPS) framework in DE which stores successful solutions into an archive, and the parents are selected from the archive when a stagnation of the population is sensed. However, the SPS framework does not explicitly allow archiving the successful search directions given by the difference vectors and it uses older solutions from archive only when the solution does not update for a considerable period.

Wang and Xiang [19] proposed a mutation strategy referred to as "DE/rand/±mean," where the population is grouped into two clusters and then the difference vector is constructed by randomly selecting component vectors from the two clusters, respectively. Bi and Xiao [20] proposed a classification-based SaDE by using the direction information with the current best solution and the best previous solution of each individual. Cai and Wang [21] defined a neighborhood for each individual and extracted three types of directional information for mutating the individual. These were the directional attraction computed from the best neighboring individual, directional repulsion computed from the worst neighboring individual, and the directional convergence from the combination of the best and the worst individuals. Note that none of these methods used a historical archive of the difference vectors formed with individuals chosen randomly from a generation, to preserve promising directions of perturbation. Cai *et al.* [22] proposed a systematic framework of improving the DE mutation through the use of multiple index-based neighborhoods and their proposal also includes a parent selection mechanism in DE by locally estimating the descent directions within the neighborhoods. Extending on a similar line of thought, Sun *et al.* [23] suggested an effective ensemble of multiple population topologies to adaptively control the information flow across different strata of the DE population.

As an important precursor to our proposal, Zhang and Yuen [24] came up with a method of creating a pool of the estimated descent directions represented by the vector differentials of the form $(\overrightarrow{U}_{i,G} - \overrightarrow{X}_{i,G})$ given that $f(\overrightarrow{U}_{i,G}) \leq f_G^{\text{best}}$, where $f(.)$ is the function to be minimized and f_G^{best} denotes the best objective function value (i.e., the lowest cost for a minimization problem). The pool of vector differentials formed in this fashion is used to guide the mutation of individuals in the next [i.e., the $(G+1)^{\text{th}}$] generation. Zhang and Zhang [25] further extended the approach of [24] by adding four repair methods when a component of the trial vector violates the boundary constraints and by using SHADE [6]-type parameter adaptation strategies for F and Cr.

Although [24] and [25] use a pool of vector differentials for creating offspring in a forthcoming generation, in what follows, we will show that the use of past difference vectors bears a different meaning for our proposal and the way in which such archived differences are used for future offspring generation is different. In Section IV-B6, we will report a detailed performance comparison between ARDE [25] and our method, when both use SHADE-type parameter adaptation.

III. Proposed Method

A. Difference Vector Archiving and Reusing Scheme

Suppose a difference vector of the form $\overrightarrow{\Delta} = \overrightarrow{X}_{r_1,G} - \overrightarrow{X}_{r_2,G}$ is used to generate a successful offspring $\overrightarrow{U}_{i,G}$ such that $f(\overrightarrow{U}_{i,G}) \leq f(\overrightarrow{X}_{i,G})$ holds true. In the next generation, there is almost no guarantee that the same difference vector (implying the same direction) will be used to mutate any other solution as the solutions $\overrightarrow{X}_{r_1,G}$ and $\overrightarrow{X}_{r_2,G}$ or any one of them may get updated.

Even if both solutions exist in the population, the probability of selecting the same two indices for any given base index i is $\sim 1/Np^2$, which is very small for a standard population size of $Np = 100$ or more. Thus, the promising search directions are hardly preserved.

In our proposed framework, we create an archive A of size Np. Starting from the first generation, as soon as a mutation leads to a successful offspring that replaces its parent at the same index, we store the difference vector(s) used in that mutation in the archive. This means if the mutation is successful for the i^{th} individual at the G^{th} generation, we store $\overrightarrow{\Delta}_{i,G}$ in archive A. In subsequent generations, for mutating the i^{th} individual, we use one of the following two schemes with probabilities p (say) and $(1-p)$, respectively:

1) we reuse a randomly selected difference vector from the archive;
2) we use a difference vector formed with the randomly picked up solutions from the current generation only, as done in a regular DE mutation.

As will be evident from our experimental results in Section IV-D3, almost always $p = 0.5$ (i.e., when we give equal importance to the archived and the current difference vectors) gives better results in the DVR scheme compared to other values of p. Note that if we reuse the archived difference vectors too much, the chances of stagnation to some suboptimal basin of the functional landscape increases, as the exploration through discovery of new promising descent directions will be hampered. In fact, this will push the majority of the individuals to similar basins of the functional landscape. On the other hand, rare use of the archived difference vectors will make the performance almost similar to a conventional DE. Hence, in a black-box optimization scenario, when we do not have additional information about the objective function, it is intuitively appealing that we give equal chances to the archived difference vectors and those from the current generation. The creation of archive of the previously successful difference vectors does not introduce any computational overhead as those difference vectors are just calculated once at the time of mutation and they might get used in forthcoming generations also, which, in fact, reduces computational cost.

At the end of each generation, the archive is updated to keep its size fixed to Np. From second generation onward, near the end of a generation, when the size of the archive exceeds Np and stays somewhere between Np and $2Np$, we simply select Np individuals uniform at random and discard the rest. Again, in a black box scenario, always preferring older difference vectors in the archive over the newer ones may run the risk of eliminating some newly discovered promising directions. On the other hand, always choosing the difference vectors that gave higher levels of fitness improvement [i.e., greater Δf_i, where $\Delta f_i = |f(\overrightarrow{X}_{i,G}) - f(\overrightarrow{U}_{i,G})|$] can lead to premature convergence quickly. Our experiments (reported in Section IV-D5) confirm that preserving Np difference vectors uniformly at random in the archive at the end of each generation forms a balance between these aspects and provides better results over a wide variety of test functions compared to the other possible schemes of an archive update.

This simple procedure, which we refer to here as the difference vector reuse (DVR), does not require any parameter adaptation and can be integrated with any DE-variant that uses the typical differential mutation step of DE. A brief pseudocode of a generic DE with DVR is provided as Algorithm 1.

Note that in the above algorithm, we do not restrict the same difference vectors to be archived again, if they become successful. Thus, the archive may contain copies of difference vectors, especially for those which became successful several times in subsequent generations. Hence, if at any generation, there are k copies of a difference vector in the archive, the probability of its selection in a subsequent mutation is $p.(k/Np)$ (given that the probability that the archive will be used is p). Thus, this archiving strategy is exploitative and behaves like the parent selection strategy of an elitist evolutionary algorithm, with *fitter* difference vectors having

Algorithm 1 DE/rand/1/bin With DVR

1: Initialize a population $\boldsymbol{P_0}$ of *Np* *D*-dimensional vectors within the pre-specified upper and lower limits. Set the initial difference vector archive $\boldsymbol{A_0} = \varnothing$.
2: **for** $G = 1$ to G_{max}
  for $i = 1$ to *Np* **do**
    if $G < 2$ **do**
      Sample a base vector $\vec{X}_{r1,G}$ and a difference vector $\vec{\Delta}_{i,G}$ from the current population $\boldsymbol{P_G}$.
      Generate the mutant vector as $\vec{V}_{i,G} = \vec{X}_{r1,G} + F.\vec{\Delta}_{i,G}$.
    end do
      Generate a uniform random number $rand_i \in [0,\ 1]$.
        if $(rand_i \le p)$
         Randomly sample a difference vector $\vec{\Delta}_k$ from $\boldsymbol{A_G}$ and set $\vec{\Delta}_{i,G} = \vec{\Delta}_k$.
        else sample $\vec{\Delta}_{i,G}$ and a base vector $\vec{X}_{r1,G}$ from the current population $\boldsymbol{P_G}$.
        end if
        Generate the mutant vector as $\vec{V}_{i,G} = \vec{X}_{r1,G} + F.\vec{\Delta}_{i,G}$.
    end if
    Generate the trial vector $\vec{U}_{i,G}$ by using binomial crossover as shown in (3).
    if $f\left(\vec{U}_{i,G}\right) \le f\left(\vec{X}_{i,G}\right)$ **do**
      Replace $\vec{X}_{i,G}$ in $\boldsymbol{P_G}$ with $\vec{U}_{i,G}$.
      Store the difference vector $\vec{\Delta}_{i,G}$ into archive $\boldsymbol{A_G}$.
    end if
  end for
    if $|\boldsymbol{A_G}| > Np$ **do**
     Discard $|\boldsymbol{A_G}| - Np$ difference vectors, selected uniform at random from $\boldsymbol{A_G}$ and set $\boldsymbol{A_{G+1}} = \boldsymbol{A_G}$.
    end if
end for
3: Return the best individual $\vec{X}_{best,G_{max}}$ from the final population and the corresponding objective function value.

more chances to take part in mutation. Such (parent) selective pressure during mutation can improve the performance of DE on nonseparable problems in concurrence with the observation made by Sutton *et al.* [26]. The greediness of this approach is, however, counterbalanced by the random pruning of the archive to maintain its size at *Np*. During the pruning process, the better difference vectors, which have been reused quite a number of times (and which have a chance to stipulate the population to suboptimal basins), are equally likely to be eliminated. Our experiments reported in Section IV-D5 confirm the effectiveness of these schemes by comparing against a modified DVR where we do not allow the same difference vector to enter the archive again.

B. Space and Time Complexity Issues

As can be perceived from Algorithm 1, a generic DE variant with the DVR scheme does not impose any serious computational burden since storing difference vectors to an archive, randomly accessing the archive, and discarding vectors from the archive are all $O(1)$ operations. The spatial complexity is also only up to the extent of storing a vector archive of length *Np*, and is similar to other archive-based DE variants like JADE and SHADE. Note that improved DE variants like JADE, SHADE, and L-SHADE [27] store the unsuccessful individuals (and not any difference vector) in the archive and reuse them later to enhance the explorative power of the population.

C. Relation and Difference From the Prior DE Variants

The DVR scheme as described earlier, considerably differs from its precursors like [24] and [25]. According to [24], the vector differentials ($\vec{U}_{i,G} - \vec{X}_{i,G}$) are archived only if the cost of a newly generated offspring vector $\vec{U}_{i,G}$ can meet or beat the current best cost. Due to such a strong condition on determining the "successful" offspring as well as the archival descent directions, the vector differentials from the archive may become fairly greedy or exploitative. On the other hand, in the proposed DVR scheme, only the difference vectors formed with the current generation vectors are stored in an archive provided that the difference vectors are able to create a successful offspring, which replaces its parent at the same index. Note that unlike [24] here, the success is not defined in terms of competing against the current best fitness in the population. The relaxed condition for archiving difference vectors and allowing them to be randomly picked up for perturbing a base vector which may have a different index (compared to the one for which the difference vector was used earlier) in the population allows for a balance between exploration and exploitation as will also be explained in Fig. 1 in Section III-D1.

In addition, according to [24] and [25], the pool used to store vector differentials is used once only in a generation and becomes invalid after the set of mutant vectors is generated. A new pool is created only if the best fitness of the population is improved at generation $G + 1$; otherwise, the original mutation operator of DE is used in the next generation and the difference vector pool is empty. However, in our proposal, the difference vector archive may contain successful perturbing differences from several earlier generations (instead of being from the immediate past generation), the difference vector archive is never empty, and it is occasionally pruned on the basis of a random elimination to keep its size constant.

D. Empirical Analysis and Justification of the DE With the DVR Scheme

In this section, we lay out an empirical analysis of the DE algorithm with the DVR mechanism with reference to certain desirable aspects of an efficient evolutionary optimizer. In particular, we explain how DVR may help achieve a better balance between the explorative and exploitative search tendencies of DE, a better contour fitting property of DE, an enhanced level of population diversity in DE, and more efficient saddle crossing behavior of the DE population.

1) Exploration—Exploitation Tradeoff: An efficient EA should strike a nice balance between exploration, which means diversification of the search to all corners of the feasible search volume and exploitation, which amounts to intensifying the search in specific basins of attraction. Based on the location of the base vector, which will be perturbed by a previously successful difference vector from the past generations per the DVR rule, the new offspring may be either pushed more toward the optima of the current basin

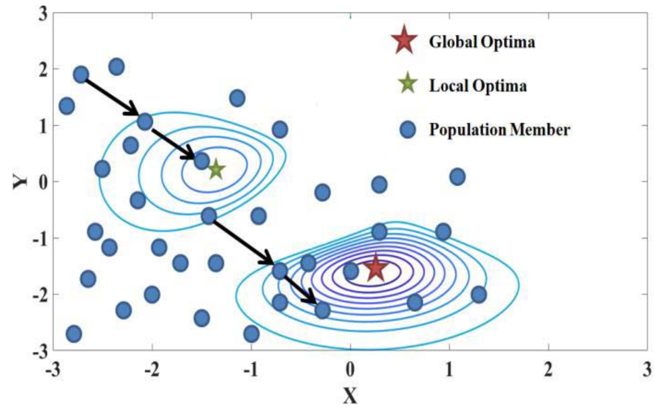

Fig. 1. Effect of past DVR on different base vectors, demonstrated on the contours of a multimodal function.

of attraction, thus favoring exploitation. On the other hand, since in DVR, we are not necessarily adding a past difference vector with a base vector of the same index (for which the difference vector was used earlier), if a previously successful difference vector drives a base vector to a direction away from a locally optimal basin, it works as an explorative search move. From Fig. 1, we observe that if a search direction was useful in an earlier generation, then that direction may drive other base vectors situated near the older one, to the optimum, thus intensifying the search in the current basin. However, if the past difference vector is added to such a base vector that lies in a different region, then we may end up with new points away from a local optimal basin and, in probability, more toward a new basin of attraction, thus facilitating the exploration of various portions of the search volume. Thus, the use of past difference vectors is likely to favor both explorative and exploitative search moves based on the distribution of the current population over the fitness landscape. Combined with the greedy and exploitative selection rules of DE, the past difference vectors reusage can also maintain the randomness of the offspring distribution and, as such, the chances of stagnation may decrease. Thus, simple reuse of these difference vectors can reduce the chances of stagnation without the need to introduce new mutation strategies or randomize the control parameters which are quite well studied in DE.

2) Preserving the Contour Matching Property: Contour matching [27], [28] is a very interesting property of DE and it refers to the gradual adaptation of the distribution of difference vectors in DE to the local shapes of the functional landscape. In this context, Storn (one of the inventors of DE) comments in [27] that—"It is intriguing to see the difference vector distribution adapts to the landscape of the objective function." On a multimodal functional landscape, when various basins of attraction around the different optima have different landscape properties, the perturbing difference vectors of the form $(\overrightarrow{X}_{r_1,G} - \overrightarrow{X}_{r_2,G})$, which had been successful in earlier generations, have a chance to retain important landscape information, in concurrence to the *contour matching* property of DE. In fact, DVR helps the difference vectors to quickly adapt to the local contours of the fitness landscape, as has been illustrated in Figs. 5 and 6 in the supplementary material for the Rosenbrock's function. As can be seen from Figs. 6A and 6B in the supplementary material, when the population of DE variants drives along a curved valley, for DE with DVR, the archived difference vectors can guide the population with the previous landscape information and, thus, can adapt to the local shape of the objective function much earlier than the canonical DE.

3) Better Saddle Crossing in DE: Saddle crossing [29] refers to the ability of a generational EA to transfer the population members from one local optimal basin to another, while crossing a ridge-like saddle between the basins. Such basin-to-basin transfer is a desirable property of DE since it helps to quickly capture the global minima over a multimodal fitness landscape. In the proposed DVR scheme, the archived difference vectors can steer a base vector of a different index (compared to the base vector for which this difference vector was used earlier) to a region far away from the nearest local minimum (see Fig. 1). Thus, in probability, DVR promotes saddle crossing behavior in DE. This fact has been illustrated on the peaks function with two minimal basins of attraction through Figs. 8 and 9 in the supplementary material. In fact, we can observe that starting from an asymmetric initialization (where the initial population is far from the actual optimal basin), DE/rand/1/bin+DVR can transfer its population members to the globally minimal basin earlier than DE/rand/1/bin. For more details on these figures, see the content description in the supplementary material.

4) Preserving the Population Diversity: The maintenance of population diversity over generations for an EA refers to the ability of the solutions to remain well-separated from each other, enabling them to uniformly cover the feasible search volume over the course of a search instead of accumulating in a smaller basin of attraction due to a small range of mutation or excessive selection pressure. In canonical DE, the distribution of difference vectors completely depends on the distribution of the current population members. However, when coupled with DVR, the reuse of the archived difference vectors during mutation dissociates the tight correspondence between the distributions of current population members and difference vectors. Thus, even when a major part of the population may tend to collapse to some very small neighborhood, some of the archived difference vectors may generate a large mutation step, thus creating fitter individuals around a different local minima and again spreading the population. In Section IV-D2, we experimentally verify that DE with DVR can better preserve the population diversity than only DE without the need to adapt the scale factor and despite the usual greedy selection strategy of DE.

IV. Experiments and Results

In this section, we elaborate on all of the simulation experiments undertaken to validate the DVR versions of DE with respect to several state-of-the-art DE variants. We also report benchmarking experiments designed to gain a deeper insight into the working principles of DVR along with its various aspects. This section is divided into five broad sections. Section IV-A sets the right background by describing the used benchmark suites, peer algorithms, and their intrinsic parameters; the computational platform used; and the simulation strategies. Section IV-B presents and discusses all of the comparative studies between DVR-induced DE variants and other state-of-the-art algorithms based on DE and covariance matrix adaptation ES (CMA-ES) on the widely known benchmark functions. The runtime complexity of DE with DVR is experimentally investigated in Section IV-C following the standard benchmarking procedures. Section IV-D experimentally analyzes the behavior of DVR and justifies its archiving and archive pruning strategies. Finally, Section IV-E showcases the efficiency of DVR in solving a real-world optimization problem involving the design of circular antenna arrays.

A. Experimental Setup

1) Benchmark Suites Used: For the comparative study reported in this section, we considered the well-accepted benchmark suites proposed for the IEEE CEC 2013 [30], 2014 [31], and 2017 [32] competition and special session on single-objective real parameter numerical optimization. While the CEC 2013 test suite consists of 28 numerical benchmarks, CEC 2014 and 2017 test suites consist of 30 functions each. These functions present various challenging traits like nonseparability, asymmetrical distribution of local optima, high degree of multimodality, shifted optima, mixture of characteristics of several basic functions (in the hybrid and composite functions), and so on to their optimizers. All of the functions are scalable and depending on the benchmarking protocols, we tested them in 30, 50, and 100 dimensions.

2) Algorithms Compared and Their Parametric Settings: Three traditional DE variants: 1) DE/rand/1/bin; 2) DE/best/1/bin; and 3) DE/current-to-best/1, and two very recent DE variants jSO [33] and LSHADE-cnEpSIn [34] were compared on the CEC 2017 test suite. For the CEC 2017 competition, jSO was the winner and LSHADE-cnEpSin ranked second among the DE-based participants. We also report the performances of DE/rand/1/bin, DE/best/1/bin, and DE/current-to-best/1 with the recently developed SPS scheme of Guo *et al.* [18] since this scheme archives successful parents and reuses them upon sensing stagnation. We compared the performance of SHADE [6] and ARDE [25] on the CEC 2013 benchmark suite [30] and L-SHADE [35] on the CEC 2014 benchmark suite [31] against the respective DVR versions to maintain the consistency of the previously reported comparative studies. To see whether the DVR version of a state-of-the-art DE variant can match the performance of an optimizer of a different breed, we undertake a comparison between jSO+DVR and a state-of-the-art CMA-ES variant, called RB-IPOP-CMA-ES [36] on the CEC 2017 functions. A real-world optimization problem involving the design of circular antenna arrays [37] is considered here to demonstrate the usefulness of the proposed scheme. For this problem, DVR is coupled with L-SHADE, and its performance is compared with the original L-SHADE along with the genetic algorithm with the multiparent recombination (GA-MPC) [38] algorithm. We selected GA-MPC as it won the 2011 CEC competition on testing evolutionary algorithms on real-world optimization problems [39].

The parameters of SHADE, L-SHADE, jSO, ARDE, and LSHADE-cnEpSin were set as described in the respective literature. For DE/rand/1, DE/best/1, DE/current-to-best/1, and their counterparts with DVR as well as SPS, we took the population size $Np = 100$, the scale factor $F = 0.8$, and the crossover rate $Cr = 0.5$ over all functions, as these are quite standard settings used in [2] and [3]. Also, we set the probability parameter $p = 0.5$ in DVR as it gives the best overall results compared to other values (as will be verified experimentally in Section IV-D3).

3) Simulation Platform: A workstation equipped with Intel Core i7, 8700k 6 cores, and 12 threads CPU running at 4.5 GHz coupled with a 16 GB of DDR4 RAM at 3200 MHz, running Microsoft Windows 10 operating system was used to execute all of the simulations reported here. The algorithms were coded in the MATLAB r2017a platform.

4) Computational Protocols and Simulation Strategy: Results were tabulated in terms of mean *best-of-the-run* error value over 51 independent runs of each of the compared algorithms on each problem following the CEC competition protocols (see [32]). The error value was recorded as $|f_{\text{best}} - f^*|$, where f_{best} denotes the best objective function value returned by the algorithm in a run and f^* is the actual optimum value of the function. In the simulation process, a result less than or equal to 10^{-8} was rounded to 0.0. The maximum number of FEs was taken as $D \times 10\,000$ following [32]. Only for the circular antenna array design problem, maximum FEs for all competing algorithms was taken to be 1.5×10^5.

In order to gauge the statistical significance of our results with respect to the other competitors, the nonparametric Wilcoxon's rank sum test [40] was conducted for independent samples at the 5% significance level. For the sake of space economy, detailed comparative results along with the outcome of the statistical tests (in terms of the P-values) are included in the supplementary material. For significance level $\alpha = 0.05$, if a P-value is less than 0.05, then the result difference should be considered as statistically significant. The summary results (Tables I–VI) show a Win/Tie/Loss (W/T/L) analysis, where a win depicts that the algorithm is the sole best performer. If the best mean error was achieved by more than one method, then the tie count for all of those methods was increased and the win count was kept unaltered. For every other case, a loss was recorded.

B. Main Results and Discussions

1) Comparison of Classical DE Variants Against Their SPS and DVR Counterparts Using CEC 2017 Functions: For space constraints, we show the summary results with maximum dimensionality, that is, $D = 100$ (which corresponds to the hardest problem instances) for the comparative study between DE/rand/1/bin, DE/best/1/bin, and DE/current-to-best/1 along with their SPS and DVR variants in Table I. For the detailed results, please see Tables I-S and I-S-P in the supplementary material. Note that in all 30 test cases, the basic DE variants with DVR significantly outperform the original and the SPS-based DE variants. Such superior performance of the DVR-equipped classical DE variants, can be attributed to the effectiveness of reusing the past difference vectors

TABLE I
W/T/L COMPARISON AMONG DE/RAND/1/BIN, DE/BEST/1/BIN, DE/CURRENT-TO-BEST/1 SCHEME, AND THEIR DVR AND SPS COUNTERPARTS ON THE CEC 2017 100-D PROBLEMS

Comparison on DE/rand/1 Scheme			
	Win	Tie	Loss
DE/rand/1	0	0	30
DE/rand/1+DVR	**30**	**0**	**0**
SPS-DE/rand/1	0	0	30
Comparison on DE/best/1 Scheme			
DE/best/1	0	0	30
DE/best/1 + DVR	**30**	**0**	**0**
SPS- DE/best/1	0	0	30
Comparison on DE/Current-to-best/1Scheme			
DE/Current-to-best/1	0	0	30
DE/Current-to-best/1+DVR	**30**	**0**	**0**
SPS-DE/Current-to-best/1	0	0	30

which can preserve the promising search directions and can be effective in maintaining a high level of population diversity throughout the search. Our experiments indicate that for lower dimensions (which are actually simpler instances of the same functions), the same conclusion consistently holds true.

DVR performs better than the SPS variants for all test instances. This indicates that storing and reusing only the successful difference vectors should suffice to drive the population toward more promising regions, and we do not really have to sense the stagnation (thus, avoiding the requirement of more tunable parameters) or store parent vectors. Statistically, a significant performance improvement can be noted for unimodal (F_1–F_3), multimodal (F_4–F_{10}), hybrid (F_{11}–F_{20}), and composition (F_{21}–F_{30}) functions. Also, the versatility of the DVR scheme is evident from the fact that it performs equally well when integrated with DE/rand/1, DE/best/1, and DE/current-to-best/1 strategies using the same values of F, Cr, and Np, respectively.

2) Comparative Performances of Two DE-Based Winners of CEC 2017 Competition and Their DVR Counterparts: In case of comparison against the two front-ranking DE variants that participated in the CEC 2017 competition, we show results for $D = 50$ and $D = 100$ in Table II (see Tables II-S and II-S-P in the supplementary material for detailed results). As indicated in this table, the DVR-equipped LSHADE-cnEpSin is able to statistically outperform the original LSHADE-cnEpSin in 23 out of 30 instances of the 50-D (i.e., 50 dimensional) functions and 18 out of 30 instances of the 100-D functions. Performances of the LSHADE-cnEpSin and its DVR version remain statistically equivalent for five 50-D functions. Only for two 50-D functions F_{16} and F_{29}, the original LSHADE-cnEpSin could yield statistically better results than its DVR counterpart. Note that F_{16} is a hybrid function mixing the properties of four basic multimodal functions whereas F_{29} is a composition function with properties like asymmetrical distribution of optima, different geometric properties around different local optima, and different properties for different variable groups besides multimodality and nonseparability. However, markedly improved performance of the DVR variant can be noted on F_{30}, which is also

TABLE II
W/T/L COMPARISON AMONG JSO AND LSHADE-CNEPSIN AND THEIR DVR COUNTERPARTS ON THE CEC 2017 100-D PROBLEMS

Algorithms	50D			100D		
	Win	Tie	Loss	Win	Tie	Loss
LSHADE-cnEpSin	2	5	23	9	3	18
LSHADE-cnEpSin +DVR	**23**	**5**	**2**	**18**	**3**	**9**
jSO	2	6	22	5	4	21
jSO+DVR	**22**	**6**	**2**	**21**	**4**	**5**

a composition function with similar complex properties. The performance of DVR on other hybrid functions indicates nontrivial improvement over the original LSHADE-cnEpSin. Coming to the 100-D functions, LSHADE-cnEpSin with DVR outperforms the original algorithm over 18 instances containing all three types of functions, namely, shifted and rotated basic functions, hybrid functions, and composition functions from the CEC 2017 test suite. A somewhat more consistent improvement of performance is noticed for jSO when equipped with the DVR scheme, as this variant could significantly outperform original jSO on 22 instances in 50-D and 21 instances in the 100-D functions. LSHADE-cnEpSin rotates the original donor vectors by using an eigencoordinate system and due to such rotation, some of the previously promising search directions may deteriorate. However, for jSO, the application of the difference vector-based perturbation is more direct and free of any coordinate rotation. Thus, for both 50 and 100-D functions, the improvement induced by DVR seems to be more consistent for jSO.

3) Comparative Performances of CMA-ES Variant and jSO with DVR on the CEC 2017 Functions: Table III (detailed results in Tables III-S and III-S-P in the supplementary material) shows the relative performance of a state-of-the-art DE variant enhanced with DVR over a state-of-the-art CMA-ES variant, called RB-IPOP-CMA-ES [36]. We chose jSO as the base algorithm for DVR and the comparison was carried out on 50-D and 100-D problems from the CEC 2017 benchmark suite. Among the 30 test problems, in 50-D simulation, jSO+DVR outperforms its contender in 22 cases and gives statistically equivalent results in three cases, but loses in five cases. For 100-D problems, jSO+DVR gives better results in 20 cases and loses to RB-IPOP-CMA-ES in 10 cases. Interestingly, in 50-D test cases, jSO+DVR outperforms RB-IPOP-CMA-ES on all unimodal functions (F_1–F_3). However, in 100-D, the latter performs better on F_2 and F_3 than jSO+DVR. In both 50-D and 100-D, jSO+DVR loses to the CMA-ES variant on the shifted and rotated Schwefel's function F_{10}. For this function, the second better local optimum is far from the global optimum [32] and the better performance of RB-IPOP-CMA-ES can be attributed to the general proficiency of CMA-ES in driving the solutions monotonically by estimating second-order information (through approximation of the inverse Hessian of the objective function). Similarly, for the composition function F_{26} in 50-D, which presents asymmetrical level sets and different properties along different local optima, RB-IPOP-CMA-ES yields better results compared to jSO+DVR. However, we also note that for the same function in 100-D, the performance

TABLE III
W/T/L COMPARISON OF jSO+DVR AND RB-IPOP-CMA-ES ON CEC 2017 50-D AND 100-D PROBLEMS

	50D			100D		
Algorithms	Win	Tie	Loss	Win	Tie	Loss
RB-IPOP-CMA-ES	5	3	22	10	0	20
jSO +DVR	**22**	**3**	**5**	**20**	**0**	**10**

TABLE IV
W/T/L COMPARISON OF L-SHADE AND ITS COUNTERPART WITH DVR FRAMEWORK ON THE CEC 2014 50-D PROBLEMS

	50D		
Algorithms	WIN	TIE	LOSS
L-SHADE	5	2	23
L-SHADE+DVR	**23**	**2**	**5**

TABLE V
W/T/L COMPARISON OF SHADE AND ITS COUNTERPART WITH DVR FRAMEWORK ON THE CEC 2013 50-D PROBLEMS

	50D		
Algorithms	WIN	TIE	LOSS
SHADE	3	2	22
SHADE+DVR	**22**	**2**	**3**

of RB-IPOP-CMA-ES significantly deteriorates compared to jSO+DVR, which indicates poor scalability of the CMA-ES variants even on favorable functional landscapes. Note that on several other functions of diverse characteristics, jSO+DVR maintains superior performance, a characteristic that is desirable in most of the black-box optimizers in the absence of specific knowledge about the functional landscape.

4) Comparative Performances of L-SHADE and Its DVR Counterpart on the CEC 2014 Test Suite: Table IV shows a summary comparison between L-SHADE and its DVR counterpart on the 50-D IEEE CEC 2014 suite (for detailed results, see Tables IV-S and IV-S-P in the supplementary material). Here, among the 30 test problems, the DVR counterpart gives statistically better results than original L-SHADE in 23 cases and is tied with it in 2 cases. The original L-SHADE algorithm gives a better result than the DVR scheme in only five cases.

5) Comparative Performances of SHADE and Its DVR Counterpart on the CEC 2013 Test Suite: In Table V, we show the performance comparison between SHADE and its DVR counterpart on the 28 benchmark functions in 50-D from the CEC 2013 suite. Elaborated results can be found from Tables V-S and V-S-P in the supplementary material. For 23 out of a total 28 cases, our DVR counterpart outperforms the original algorithm. For F_{11} and F_{14}, SHADE gives better results, however, the performance gap is relatively small from a practical point of view. In the other three cases (F_1, F_5, and F_{19}), both end with statistical ties.

6) Comparative Performances of SHADE and Its DVR Counterpart Against ARDE on the CEC 2013 Test Suite in 30-D: In Table VI, we summarize the comparative performance between SHADE, its DVR version, and the ARDE [25] algorithm on the 28 benchmark functions in 30-D from the CEC 2013 suite, since in [25], Zhang and Zhang also used the same test suite for ARDE. Detailed results can be found in Table VI-S and VI-S-P in the supplementary material. In this case, out of 28 test functions, in 17 cases, SHADE+DVR outperforms both original SHADE and ARDE. SHADE+DVR gives statistically equivalent results in eight cases and performs worst in only three cases. Note that ARDE gives better results compared to the DVR enhanced SHADE on the shifted and rotated version of the Zakharov function F_3 which is unimodal and nonseparable.

TABLE VI
W/T/L COMPARISON OF SHADE, ITS COUNTERPART WITH DVR FRAMEWORK, AND ARDE ON THE CEC 2013 30-D PROBLEMS

	30D		
Algorithms	WIN	TIE	LOSS
SHADE	1	5	22
ARDE	1	10	17
SHADE+DVR	**17**	**8**	**3**

As it seems, the strongly exploitative search moves generated by using the strict (estimated) descent directions of the form $(\vec{U}_{i,G} - \vec{X}_{i,G})$ help the population of ARDE to quickly converge to the flat optimal basin of F_3 transgressing through its monotonic and wide slope (see [32, Fig. 3]). Also, on the hybrid function F_{17}, the results of ARDE and SHADE are mutually equivalent and better than those of SHADE+DVR. ARDE performs relatively better on the unimodal functions and such performance improvement is likely to occur by driving the subsequent individuals through sharper descent directions. However, SHADE+DVR is able to maintain its superior performance on a wider variety of problems from this test suite including all of the composition functions which mix properties of various individual benchmarks and present different landscape structures around different local optima. Promising difference vectors used for perturbation in the early generations can learn some important landscape properties and, thus, reusing them in the future can help to achieve basin-to-basin transfer in DE, a property that results from contour matching, as was discussed in Section III-D.

C. Algorithm Complexity

To illustrate the fact that DVR does not induce any significant computational cost on the underlying DE variant, we measure the complexity of DE/best/1/bin with DVR as instructed in [32]. In Table VII, *T0* is the time needed to run the following test problem:

$$
\begin{aligned}
&\text{for } \ i = 1{:}1000000\\
&\quad x = x + x;\ \ x = x/2;\ \ x = x * x\\
&\quad x = \mathrm{sqrt}();\ \ x = \log(x)\\
&\quad x = \exp(x);\ \ x = x/(x+2)\\
&\text{end.}
\end{aligned}
$$

We computed the complexity for $D = 10$, 30, and 50. *T1* is the time required to execute 200 000 FEs of F_{18} from the CEC 2017 benchmark suite and *T2* is the time to execute DE/best/1/bin with DVR for 200 000 FEs on the same F_{18} in D dimensions. *T2_O* is the time to execute DE/best/1/bin with 200 000 FEs on F_{18} in dimension D. The entries of Table VII

TABLE VII
SAMPLE ALGORITHM COMPLEXITY RESULTS FOR DE/BEST/1/BIN AND DE/BEST/1/BIN+DVR ON CEC 2017 FUNCTION NO. 18 PER [30]

D	*T0*	*T1*	*T2*	*T2_o*	*(T2_o-T1)/T0*	*(T2-T1)/T0*
10	0.0958	0.7978	2.1256	1.9863	12.4060	13.8601
30		1.7974	3.3695	3.2154	14.8016	15.4102
50		3.3947	5.4521	5.1236	18.0469	20.4759

indicate that the complexity of DE/best/1/bin with DVR is only marginally higher than the original algorithm.

D. Different Aspects of the DVR Scheme—Deeper Experimental Analysis

In this section, we report the representative experimental results to demonstrate how DVR enhances the production of more successful offspring in a DE population, leading to quicker fitness improvement. We also experimentally analyze the population diversity of the DVR versions of classical DE, the effect of parameter p on DVR, and the effect of possible archiving and archive pruning strategies in DVR.

1) Effect of DVR on Classical DE Variants in Terms of Successful Replacement and Fitness Improvement: The number of successful offspring replacements over generations can be considered as an indicator of the progress of the search toward a fruitful direction. Fig. 2 shows a sample plot of the average successful replacement versus FEs for DE/rand/1/bin, DE/best/1/bin, and their counterparts employing the DVR scheme for a few representative functions of the CEC 2017 test suite. Among these functions, F_5, F_6, and F_{10} are multimodal in nature; F_{15} belongs to hybrid function group; and F_{21}, F_{25}, F_{28}, and F_{30} are composite in nature. Except for F_5, due to space constraints, the other plots are moved to the supplementary material (as Fig. 2A and 2B). In all of these cases, it can be clearly seen that the average number of successful replacements is always higher for algorithms coupled with DVR compared to their original form. This implies that reusing the past difference vectors in a controlled manner can induce a higher probability of generating better offspring by utilizing the already learned promising search moves in a complicated fitness landscape.

Fig. 3 shows the average absolute difference between the offspring (trial) and target costs by increasing FEs for function F_5 of the CEC 2017 suite, while the same plots for the remaining representative functions corresponding to the above-mentioned function groups have been shown as Fig. 3A and 3B in the supplementary material. It can be clearly seen that for both DE variants, the one with DVR always has an increasing average absolute difference between the offspring cost and the target cost, thus indicating greater variance of the evolutionary process. On the other hand, the original DE strategies stagnate earlier; as for them, the difference between the offspring cost and the target cost is that it is gradually and uniformly reduced, suggesting that the overall search gets stuck at a locally optimal basin. Thus, Figs. 2 and 3 empirically indicate that the proposed DVR has a positive impact on the search performance of DE.

TABLE VIII
W/T/L COMPARISON FOR DE/RAND/1/BIN+DVR WITH DIFFERENT VALUES OF p ON ALL OF THE CEC 2017 100-D PROBLEMS. BEST RESULTS ARE MARKED IN BOLDFACE

	100D		
Parameter Settings	**WIN**	**TIE**	**LOSS**
$\boldsymbol{p = 0.1}$	0	0	30
$\boldsymbol{p = 0.3}$	0	0	30
$\boldsymbol{p = 0.5}$	**30**	**0**	**0**
$\boldsymbol{p = 0.7}$	0	0	30
$\boldsymbol{p = 0.9}$	0	0	30

Note that Figs. 2–4 (in Section IV-D2) have been generated from the median of 51 independent runs of each algorithm on the designated test functions when the runs were ranked on the basis of their "best-of-the-run" error values.

2) Population Diversity Analysis for the DVR Scheme: The DVR scheme can conserve the diversity of a DE population to a considerable extent. As proposed in [41], the diversity of a given population P_G at generation G can be measured by the average distance of the population from the mean vector, and it is defined as

$$\text{diversity}(P_G) = \frac{1}{N_p \times L} \sum_{i=1}^{N_p} \sqrt{\sum_{j=1}^{D} \left(x_{i,j} - \bar{x}_j\right)^2} \tag{4}$$

where the length of the longest diagonal in the current search space is L, and $\bar{x}_j = (1/N_p) \sum_{i=1}^{N_p} x_{i,j}$ is the average value of the jth dimension of the members of a population. Fig. 4 shows the population diversity comparison between DE/rand/1/bin and its DVR counterpart on a few CEC 2017 functions with diverse characteristics that are also used for the illustrations explained in Section IV-D1. Like Figs. 2 and 3, the plots for functions F5 and F15 are kept in the main paper, while the remainder are moved to the supplementary material as Fig. 4A. It can be seen that for all cases, the DVR variant retains much more diversity in the population compared to the original scheme and despite the usual selection pressure of DE. As DVR succeeds in preserving the diversity, it can be concluded that it has better chances of avoiding local optima and, thus, yields better search performance when compared to a standard DE/rand/1/bin scheme.

3) Effect of p on the DVR Scheme: In this section, we study the effect of different values of p, the probability parameter which decides whether the mutation will involve a difference vector from the archive or will proceed like a conventional DE mutation, on the overall search performance of the algorithm. We consider five different values of p which are 0.1, 0.3, 0.5, 0.7, and 0.9. To show the sample results, we consider all of the 30 100-D test functions from the CEC 2017 benchmark suite. Table VIII summarizes the results in terms of the statistical Win/Tie/Loss with respect to mean best-of-the-run error and standard deviations on a set of 51 independent runs of DE/rand/1/bin+DVR with the different values of p. Detailed results with statistical test outcomes are provided in Tables VIII-S and VIII-S-P in the supplementary material.

Table VIII indicates that taking $p = 0.5$ produces the best result among all other considerations. Although we took DE/rand/1/bin as the base algorithm, our conclusions remain

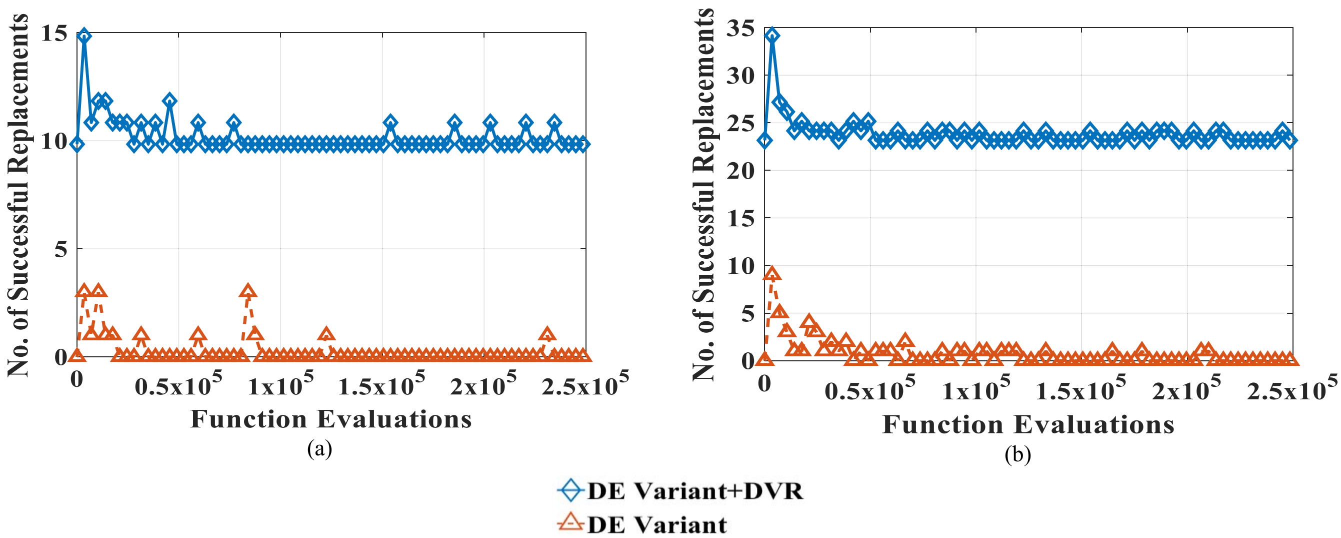

Fig. 2. Comparison of (a) DE/rand/1/bin and (b) DE/best/1/bin and their counterparts with the proposed DVR strategy in terms of the number of successful replacements with FEs on function F_5 of CEC 2017 suite.

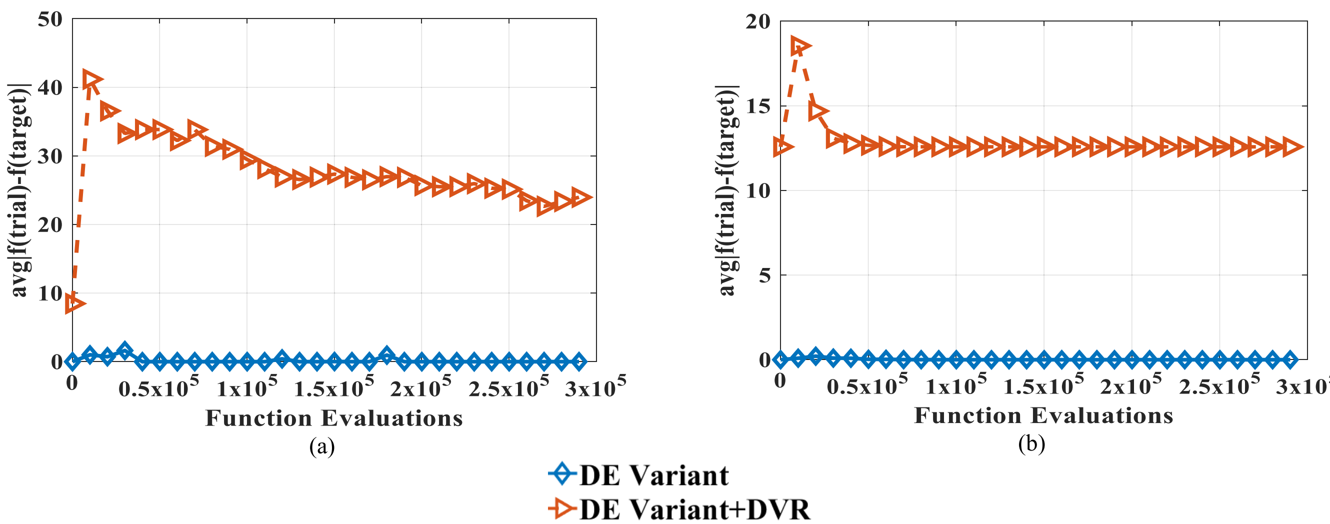

Fig. 3. Comparison of (a) DE/rand/1/bin and (b) DE/best/1/bin and their counterparts with the proposed DVR strategy in terms of the average absolute difference of offspring (trial) cost and target cost with FEs on function F_5 of CEC 2017 suite.

unchanged for other basic DE strategies like DE/best/1 and DE/current-to-best/1 as well. Better results obtained for $p = 0.5$ concur with our remark in Section III-C that too much use of archived difference vectors (for large p) can lead to false convergence, whereas rare use of the difference vectors (for small p) can make the performance similar to conventional DE.

4) Experimental Analysis of the Archiving Strategies for DVR: In this section, we provide a comparative view of two difference vector archiving strategies. Of these, one is the original DVR where we allow multiple copies of the same difference vector to be archived as they become successful in multiple generations. The other archiving scheme (modified DVR) is the one which allows only one copy of the difference vector in the archive. This scheme can be obtained by slightly modifying Algorithm 1 in Section III-A. Based on the probability p, if a difference vector is sampled from the current generation, then only on generating a successful offspring, it will be allowed to enter the archive. However, if the difference vector is sampled from the archive, despite generating a successful individual, it will not be permitted to enter the archive as another duplicate copy. Table IX summarizes the results (in terms of statistical Win/Tie/Loss) on all of the 30 100-D benchmark functions from the CEC 2017 test suite with the two archiving strategies used in the DVR integrated with the DE/rand/1/bin scheme. Detailed results can be found in Tables IX-S and IX-S-P in the supplementary material.

Table IX indicates that the archiving strategy used in DVR is more suitable in improving the performance of a DE variant on a diverse set of problems, as in all of the tested cases, modified DVR was unable to yield statistically superior results compared to the original DVR.

5) Empirical Analysis of Different Archive Pruning Strategies: In order to demonstrate the appropriateness of the considered archive pruning strategy in DVR (as given in

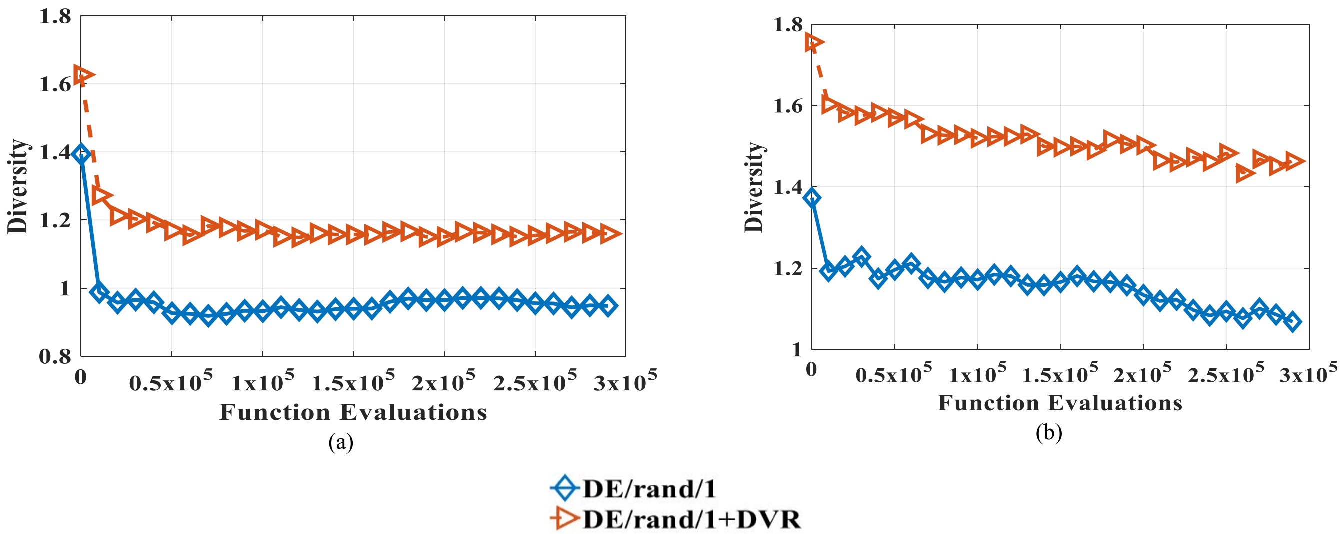

Fig. 4. Population diversity comparison among DE/rand/1/bin and its counterpart equipped with the DVR strategy on functions (a) F_5 and (b) F_{15} of CEC 2017.

TABLE IX
W/T/L COMPARISON FOR DE/RAND/1+DVR WITH DIFFERENT DVR CONFIGURATION ON ALL THE CEC 2017 100-D PROBLEMS. BEST RESULTS ARE MARKED IN BOLDFACE

	100D		
Parameter Settings	**WIN**	**TIE**	**LOSS**
DE/RAND/1 + DVR	**30**	**0**	**0**
MODIFIED DVR	0	0	30

TABLE X
COMPARISON OF DIFFERENT ARCHIVE PRUNING STRATEGIES WITH WIN/TIE/LOSS OVER 3 CEC COMPETITION BENCHMARKS

	CEC 2013			CEC 2014			CEC 2017		
	S1	S2	S3	S1	S2	S3	S1	S2	S3
W/T/L	**22/5/0**	1/5/0	0/5/0	**20/3/6**	7/3/20	0/0/0	**26/3/1**	1/3/26	0/0/0

Algorithm 1) over a few other possible strategies, we conduct a study over all three CEC test suites and use DE/best/1/bin with DVR separately involving the following three archive pruning strategies. Note that S1 is the strategy that is used with the DVR scheme discussed in this paper. In the S2 scheme, inclusion of the difference vector, which is already present in the archive, is prohibited.

S1: When the archive size exceeds a predefined size (= Np here), we randomly select Np members from the present archive and create a new archive.

S2: When the archive size exceeds a predefined size, we sort the current archive members on the basis of maximum fitness improvement from parent to offspring, and then select the top Np members from the present archive to create a new archive.

S3: When the archive size exceeds the predefined size, we start deleting those members which were used the least number of times until the new archive size becomes acceptable. For this, we keep a counter for each member to monitor how many times it is called to create mutants during the DE mutation.

In this section, we consider $D = 50$ for CEC 2013 and $D = 100$ for CEC 2014 and 2017 test suites. Table X clearly suggests for all cases, our opted strategy S1 gives the best result. Hence, strategy S1 was selected to prune the archive in DVR. S2 is more incumbent toward greedy search moves as the direction of maximum fitness improvement may not necessarily drive a solution to a globally optimal basin, but may cause entrapment in some funnel-like local optimal basin. On the other hand, S3 can cause elimination of some truly good directions of perturbation which were not considered in the recent past due to the randomness of selecting difference vectors from the archive.

E. Performance of the DVR Scheme on the Circular Antenna Array Design Problem

In the era of digital communication, antenna arrays are exhaustively used in various mobile and wire-free communication systems. To handle regularly expanding the channel capacities of these communication systems, the optimum design of such antenna arrays is of vital importance. In long-distance communication, it is quite hard to satisfy the gain requirement along with highly directive radiation pattern conditions. Many individual antennas are arranged by obeying certain electrical and geometrical configurations in antenna arrays. There are many applications of antenna arrays like sonar [42], radar [43], global positioning system (GPS) [44], and in various generation wireless communication systems like 3G, 4G, and recently 5G. The objective of this optimization problem is to obtain the optimum parameter set contributing to the geometry of antenna arrays by finding the appropriate positions of the array elements. Optimization of the position of the elements in the antenna array is very crucial for producing a desired radiation pattern.

Fig. 5 [37] shows the constituent elements of a circular antenna array. Let us consider N antenna elements spaced on a circle having radius r in the x–y plane. Equation (5) shows the expression for the array factor (AF) value corresponding

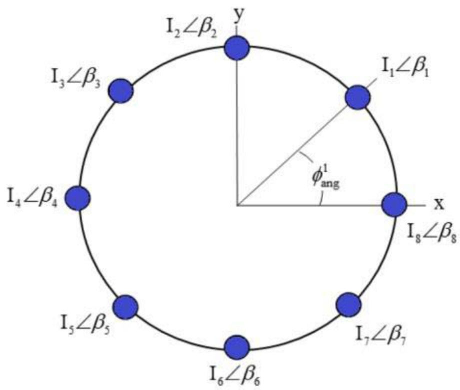

Fig. 5. Geometry of the circular antenna array (adopted from [35]).

to that N element circular antenna array

$$\mathrm{AF}(\phi) = \sum_{n=1}^{N} I_n \cdot \exp\Big[j \cdot k \cdot r \cdot \Big(\cos\Big(\phi_0 - \phi_{\mathrm{ang}}^n\Big) - \cos\Big(\phi_0 - \phi_{\mathrm{ang}}^n\Big)\Big) + \beta_n\Big] \quad (5)$$

where

- ϕ_{ang}^n $2 \cdot \pi \cdot [(n-1)/N]$ nth element angular position on the x–y plane;
- k wave-number, here $k \cdot r = N \cdot d$ and d denote angular spacing between the antenna elements;
- r radius of the circle in which antennas are arranged;
- ϕ_0 maximum radiation direction;
- ϕ angle of incidence of the plane wave;
- I_n current excitation;
- β_n nth element phase excitement.

Here, we try to suppress side-lobes, minimize beam width, and achieve null control at the desired direction by varying the current and phase excitations of the antenna elements. Equation (6) defines the constraints between elements of the designed circular antenna array

$$\begin{aligned} I_{\frac{n}{2}+1}\angle\beta_{\frac{n}{2}+1} &= \mathrm{conj}(I_1\angle\beta_1) \\ I_{\frac{n}{2}+1}\angle\beta_{\frac{n}{2}+1} &= \mathrm{conj}(I_2\angle\beta_2), \ldots \\ I_n\angle\beta_n &= \mathrm{conj}\Big(I_{\frac{n}{2}}\angle\beta_{\frac{n}{2}}\Big). \end{aligned} \quad (6)$$

In this paper, the cost function for the circular antenna array design is defined per [35] in the following way:

$$\begin{aligned} F &= w_1 \cdot \Big|\mathrm{AR}\Big(\phi_{sll}, \vec{I}, \vec{\beta}, \varphi_0\Big)\Big| / \Big|\mathrm{AR}\Big(\phi_{\max}, \vec{I}, \vec{\beta}, \varphi_0\Big)\Big| \\ &+ w_2 \cdot \frac{1}{\mathrm{DIR}\Big(\phi_0, \vec{I}, \vec{\beta}\Big)} + w_3 \cdot |\phi_0 - \phi_{\mathrm{des}}| \\ &+ w_4 \cdot \sum_{k=1}^{\mathrm{num}} \Big|\mathrm{AR}\Big(\phi_{sll}, \vec{I}, \vec{\beta}, \varphi_0\Big)\Big| \end{aligned} \quad (7)$$

where we considered $w_1 = w_2 = w_3 = w_4 = 1$. In (7), the first component minimizes the side lobes. ϕ_{sll} is the angle where the maximum level of side lobe is achieved. Directivity minimization is handled by the second component. The third component dictates the maxima of the array pattern close to the desired maxima ϕ_{des}. The fourth components acts as a penalty term to the cost function if adequate null control is not attained by the array. The number of null control directions is given by num and ϕ_k describes the direction of kth null control. In this paper, we consider a circular antenna array with 12 elements ($N = 12$). All test parameters are given in Table XI.

We consider L-SHADE as the base algorithm to our DVR scheme as many recent DE variants for CEC competitions are based on L-SHADE. The performance of L-SHADE+DVR is compared with the original L-SHADE with its parameter sets defined in [35] and the GA-MPC algorithm [38]. The performance comparison among these algorithms is summarized in Table XI in terms of the mean cost and standard deviations over 51 independent runs. In Table VIII, the statistical test results are summarized in the following way. If the final error yielded by an algorithm is statistically significantly different from that of the best-performing algorithm on a particular function, then the mean error of the former is marked with a † symbol. If the difference of the error values

TABLE XI
PARAMETERS OF THE CIRCULAR ANTENNA ARRAY DESIGN PROBLEM USED IN TEST

Test #	low	high	Element #	ϕ_{des}	null control
Test 1	[0.2, 0.2, 0.2, 0.2, 0.2, 0.2, -180, -180, -180, -180, -180, -180]	[1, 1, 1, 1, 1, 1, 180, 180, 180, 180, 180, 180]	12	180^0	-

TABLE XII
PERFORMANCE IN TERMS OF COST ± (STANDARD DEVIATION) OF L-SHADE, GA_MPC, AND L-SHADE+DVR ON CIRCULAR ANTENNA ARRAY DESIGN PROBLEM

Element #	L-SHADE	GA-MPC	L-SHADE + DVR
12	-2.05E+01† (1.23E+00)	-2.16E+01† (4.36E+00)	**-2.19E+01 (5.65E-07)**
P-value	1.77E-12	5.67E-16	**NA**
W/T/L	0/0/1	0/0/1	**1/0/0**

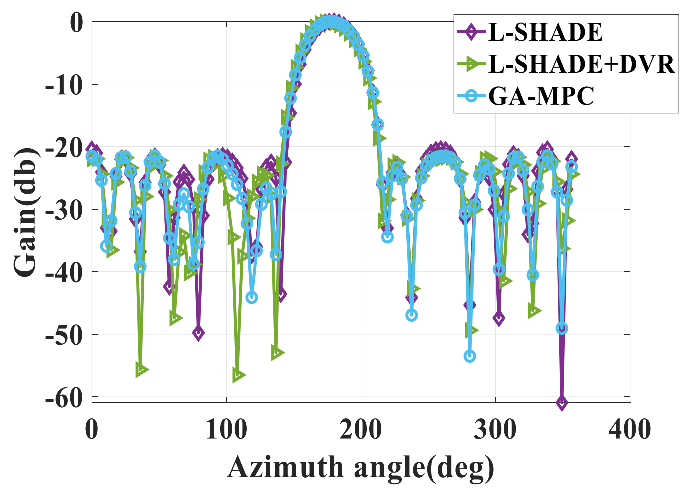

Fig. 6. Radiation pattern comparisons for the 12 element circular antenna array design problem.

found by one algorithm is not significantly different from the best algorithm, then the mean of this algorithm is marked with the sign $\approx$. The P-values obtained with the rank-sum test are also indicated in the second row. For a 5% significance-level test, if a P-value is less than 0.05, then the resulting difference should be considered as statistically significant.

Table XII suggests that the DVR variant gives better results than the other two competitors with statistical significance. Fig. 6 also depicts that DVR enhanced L-SHADE provides the most desired radiation pattern when compared to the other two algorithms. The radiation patterns obtained with these three algorithms are shown in Fig. 6, which reveals better directivity and null control obtained by the L-SHADE+DVR scheme.

V. Conclusion

DE has been subjected to several modifications aiming to boost its performance on complex optimization problems. Many of these variants introduced additional parameters and/or necessitated costly operations for strategy and parameter control. In this paper, we showed that a simple strategy of storing the successful difference vectors and reusing them in future generations in a stochastic fashion can significantly improve the performance of DE as suggested by our experiments on the well-accepted benchmark suites from CEC 2013, 2014, and 2017 competitions on real parameter bound-constrained optimization problems. The performance of the DVR scheme is also tested on a circular antenna array design problem. Storing the successful difference vectors can induce stochastic learning of the better descent directions on a fitness landscape, thus providing better guidance to future offspring vectors. The scheme can be integrated with any DE variant without incurring a serious computational burden.

This paper can be extended in several ways. The scheme of pruning the archive can be made more adaptive. Our tested DE variants used the binomial crossover which generates a final offspring or trial vector at one of the vertices of a simplex formed with the current target and donor as two diagonally opposite points and the DVR fits quite well to this scheme. The effect of the DVR integrated with other kinds of crossovers (e.g., exponential and arithmetic) can be further investigated. Since as per our benchmarking experiments, DVR can improve a diverse set of DE mutations (DE/rand/1, DE/best/1, DE/current-to-best/1, DE/current-to-*p*best-w/1 in jSO, DE/current-to-*p*best/1 of LSHADE-cnEpSin, etc.), the effect of DVR can be further investigated in the recent constructive multimutation DE frameworks like [45]. The fundamental idea of DVR can be used in conjunction with other metaheuristics like particle swarm optimization (PSO) where the most promising velocity directions can be stored and reused to guide the flight of particles in future iterations.

References

[1] R. Storn and K. Price, "Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces," Int. Comput. Sci. Inst., Berkeley, CA, USA, Rep. TR-95-012, 1995.

[2] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[3] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution—An updated survey," *Swarm Evol. Comput.*, vol. 27, pp. 1–30, Apr. 2016.

[4] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.

[5] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.

[6] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Cancún, Mexico, Jun. 2013, pp. 71–78.

[7] M. G. Epitropakis, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis, "Enhancing differential evolution utilizing proximity-based mutation operators," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 99–119, Feb. 2011.

[8] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 482–500, Apr. 2012.

[9] S.-M. Guo and C.-C. Yang, "Enhancing differential evolution utilizing eigenvector-based crossover operator," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 31–49, Feb. 2015.

[10] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 55–66, Feb. 2011.

[11] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Appl. Soft. Comput.*, vol. 11, no. 2, pp. 1679–1696, Mar. 2011.

[12] S. X. Zhang, S. Y. Zheng, and L. M. Zheng, "An efficient multiple variants coordination framework for differential evolution," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2780–2793, Sep. 2017.

[13] Y.-Z. Zhou, W.-C. Yi, L. Gao, and X.-Y. Li, "Adaptive differential evolution with sorting crossover rate for continuous optimization problems," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2742–2753, Sep. 2017.

[14] M. Z. Ali, N. H. Awad, P. N. Suganthan, and R. G. Reynolds, "An adaptive multipopulation differential evolution with dynamic population reduction," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2768–2779, Sep. 2017.

[15] L. Tang, Y. Dong, and J. Liu, "Differential evolution with an individual-dependent mechanism," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 560–574, Aug. 2015.

[16] X. Qiu, K. C. Tan, and J.-X. Xu, "Multiple exponential recombination for differential evolution," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 995–1006, Apr. 2017.

[17] J. Zhang and W. Luo, "Directed differential evolution based on directional derivative for numerical optimization problems," in *Proc. Int. Conf. Hybrid Intell. Syst. (HIS)*, Malacca, Malaysia, 2011, pp. 340–345.

[18] S.-M. Guo, C.-C. Yang, P.-H. Hsu, and J. S.-H. Tsai, "Improving differential evolution with a successful-parent-selecting framework," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 717–730, Oct. 2015.

[19] Y.-X. Wang and Q.-L. Xiang, "Exploring new learning strategies in differential evolution algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2008, pp. 204–209.

[20] X. J. Bi and J. Xiao, "Classification-based self-adaptive differential evolution with fast and reliable convergence performance," *Soft Comput.*, vol. 15, no. 8, pp. 1581–1599, Aug. 2011.

[21] Y. Cai and J. Wang, "Differential evolution with neighborhood and direction information for numerical optimization," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2202–2215, Dec. 2013.

[22] Y. Cai, G. Sun, T. Wang, H. Tian, Y. Chen, and J. Wang, "Neighborhood-adaptive differential evolution for global numerical optimization," *Appl. Soft Comput.*, vol. 59, pp. 659–706, Oct. 2017.

[23] G. Sun, Y. Cai, T. Wang, H. Tian, C. Wang, and Y. Chen, "Differential evolution with individual-dependent topology adaptation," *Inf. Sci.*, vol. 450, pp. 1–38, Jun. 2018.

[24] X. Zhang and S. Y. Yuen, "A directional mutation operator for differential evolution algorithms," *Appl. Soft Comput.*, vol. 30, pp. 529–548, May 2015.

[25] X. Zhang and X. Zhang, "Improving differential evolution by differential vector archive and hybrid repair method for global optimization," *Soft Comput.*, vol. 21, no. 23, pp. 7107–7116, 2017.

[26] A. M. Sutton, M. Lunacek, and L. D. Whitley, "Differential evolution and non-separability: Using selective pressure to focus search," in *Proc. Conf. Genet. Evol. Comput. (GECCO)*, 2007, pp. 1428–1435.

[27] R. Storn, “Differential evolution research: Trends and open questions,” in *Advances in Differential Evolution*, U. K. Chakraborty, Ed. Heidelberg, Germany: Springer, 2008, pp. 1–32.

[28] K. R. Opara and J. Arabas, “The contour fitting property of differential mutation,” *Swarm Evol. Comput.*, to be published. [Online]. Available: https://doi.org/10.1016/j.swevo.2018.09.001

[29] J. Arabas, L. Bartnik, and K. R. Opara, “DMEA—An algorithm that combines differential mutation with the fitness proportionate selection,” in *Proc. IEEE Symp. Differ. Evol.*, 2011, pp. 49–56.

[30] 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,” Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, and Nanyang Technol. Univ., Singapore, Rep. 201212, 2013.

[31] J. J. Liang, B. Y. Qu, and P. N. Suganthan, “Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization,” Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, and Nanyang Technol. Univ., Singapore, Rep. 201311, 2014.

[32] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, and P. N. Suganthan, “Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization,” Nanyang Technol. Univ., Singapore, Rep., Nov. 2016.

[33] J. Brest, S. M. Maučec, and B. Bošković, “Single objective real-parameter optimization: Algorithm jSO,” in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2017, pp. 1311–1318.

[34] N. H. Awad, M. Z. Ali, and P. N. Suganthan, “Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems,” in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 372–379.

[35] R. Tanabe and A. S. Fukunaga, “Improving the search performance of SHADE using linear population size reduction,” in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Beijing, China, Jul. 2014, pp. 1658–1665.

[36] R. Biedrzycki, “A version of IPOP-CMA-ES algorithm with midpoint for CEC 2017 single objective bound constrained problems,” in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2017, pp. 1489–1494.

[37] A. Mandal, H. Zafar, S. Das, and A. V. Vasilakos, “Efficient circular array synthesis with a memetic differential evolution algorithm,” *Progr. Electromagn. Res. B*, vol. 38, pp. 367–385, 2012.

[38] S. M. Elsayed, R. A. Sarker, and D. L. Essam, “GA with a new multi-parent crossover for solving IEEE-CEC2011 competition problems,” in *Proc. IEEE Congr. Evol. Comput. (CEC)*, New Orleans, LA, USA, Jun. 2011, pp. 1034–1040.

[39] S. Das and P. N. Suganthan, “Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems,” Jadavpur Univ., Kolkata, India, and Nanyang Technol. Univ., Singapore, Rep., 2010.

[40] J. Derrac, S. García, D. Molina, and F. Herrera, “A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms,” *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, 2011.

[41] J. Hu, J. Zeng, and Y. Tan, “A diversity-guided particle swarm optimizer for dynamic environments,” in *Proc. Int. Conf. Life Syst. Model. Simul.*, 2007, pp. 239–247.

[42] A. Bellettini and M. A. Pinto, “Theoretical accuracy of synthetic aperture sonar micronavigation using a displaced phase-center antenna,” *IEEE J. Ocean. Eng.*, vol. 27, no. 4, pp. 780–789, Oct. 2002.

[43] M. Akcakaya and A. Nehorai, “Adaptive MIMO radar design and detection in compound-Gaussian clutter,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 3, pp. 2200–2207, Jul. 2011.

[44] A. A. Heidari, M. Heyrani, and M. Nakhkash, “A dual-band circularly polarized stub loaded microstrip patch antenna for GPS applications,” *Progr. Electromagn. Res.*, vol. 92, pp. 195–208, Jan. 2009.

[45] Y. Cai, C. Shao, Y. Zhou, S. Fu, H. Zhang, and H. Tian, “Differential evolution with adaptive guiding mechanism based on heuristic rules,” *IEEE Access*, vol. 7, pp. 58023–58040, 2019.

Arka Ghosh received the M.Tech. degree in information technology from the Indian Institute of Engineering Science and Technology Shibpur, Howrah, India, in 2015, where he is currently pursuing the Ph.D. degree in computer science and technology.

He has published research articles in peer-reviewed journals and international conference proceedings. His current research interests include evolutionary algorithms and data mining.

Swagatam Das received the B.E.Tel.E., M.E.Tel.E. (control engineering specialization), and Ph.D. degrees in engineering from Jadavpur University, Kolkata, India, in 2003, 2005, and 2009, respectively.

He is currently serving as an Associate Professor with the Electronics and Communication Sciences Unit, Indian Statistical Institute Kolkata, Kolkata. He has published over 300 research articles in peer-reviewed journals and international conferences. He has over 16 500 Google Scholar citations with an *H*-index of 61. His current research interests include machine learning and nonconvex optimization.

Dr. Das was a recipient of the 2012 Young Engineer Award from the Indian National Academy of Engineering and the 2015 Thomson Reuters Research Excellence India Citation Award as the Highest Cited Researcher from India in the Engineering and Computer Science Category from 2010 to 2014. He is the Founding Co-Editor-in-Chief of *Swarm and Evolutionary Computation*.

Asit Kr. Das received the B.Tech. and M.Tech. degrees in computer science and technology from Calcutta University, Kolkata, India, in 1996 and 2002, respectively, and the Ph.D. degree in engineering from Bengal Engineering and Science University Shibpur, Howrah, India, in 2011.

He is currently serving as a Professor with the Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology Shibpur, Howrah. He has published 1 research monograph, 3 edited books, and over 100 research articles in peer-reviewed journals and international conferences. His current research interests include data mining and pattern recognition, evolutionary computing, and audio and video processing.

Dr. Das has been associated with the international program committees and organizing committees of several regular international conferences, including International Conference on Computational Intelligence in Data Mining, Soft Computing in Data Analytics, International Conference on Emerging Technologies in Data Mining and Information Security, and Computational Intelligence in Pattern Recognition. He has acted as a Guest Editor for special issues on “Nature Inspired Optimization and Its Application to Engineering” in *Evolutionary Intelligence* (Springer); “Hybrid Intelligent Techniques: Foundations, Applications and Challenges” in the *International Journal of Automation and Control* (Inderscience), and “Advances and Challenges of Soft Computing in Data Mining” in the *International Journal of Computational Systems Engineering* (Inderscience).

Liang Gao received the B.Sc. degree in mechatronic engineering from Xidian University, Xi’an, China, in 1996 and the Ph.D. degree in mechatronic engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2002.

He is a Professor of the Department of Industrial and Manufacturing System Engineering, School of Mechanical Science and Engineering, and the Vice Director of the State Key Laboratory of Digital Manufacturing Equipment, HUST. He has published over 220 refereed papers. His current research interests include optimization in design and manufacturing.

Dr. Gao currently serves as the Editor-in-Chief of *IET Collaborative Intelligent Manufacturing*, an Associate Editor of *Swarm and Evolutionary Computation* and the *Journal of Industrial and Production Engineering*, and an Editorial Board Member of *Operations Research Perspectives*.