Soft Comput
DOI 10.1007/s00500-015-1911-2

METHODOLOGIES AND APPLICATION

Differential Evolution with Grid-Based Parameter Adaptation

Vasileios A. Tatsis[1] · **Konstantinos E. Parsopoulos**[1]

Abstract The reduction of human intervention in tuning metaheuristic optimization algorithms has been an ongoing research pursuit. Differential Evolution is a very popular algorithm that counts a large number of variants. However, its efficiency has been shown to depend on the type of its crossover operators (binomial or exponential), mutation operators, as well as on the two parameters that dominate these procedures. Making proper decisions on these parameters has proved to be a laborious, problem-dependent task. We propose a parameter adaptation technique that allows the algorithm to dynamically determine the most suitable crossover type and parameter values during its execution. The technique is based on a search procedure in the discretized parameter search space, using estimations of the algorithm's performance. The proposed approach is tested and statistically validated on an established high-dimensional test suite. Also, comparisons with other algorithms are reported, verifying the competitiveness of the proposed approach.

Keywords Metaheuristic Optimization · Dynamic Parameter Adaptation · Parameter Control · Differential Evolution

Communicated by V. Loia.

✉ Konstantinos E. Parsopoulos
kostasp@cs.uoi.gr

Vasileios A. Tatsis
vtatsis@cs.uoi.gr

[1] Department of Computer Science and Engineering, University of Ioannina, GR-45110 Greece, Ioannina

1 Introduction

Metaheuristics are widely acknowledged as essential tools for solving difficult optimization problems in diverse scientific fields Bäck (1996). Albeit solution optimality is not guaranteed, they can provide suboptimal solutions of complex real-world problems in reasonable time. This renders metaheuristics a valuable alternative especially in cases where traditional optimization tools or analytical approaches fail.

The performance of metaheuristics typically depends on their control parameters and the particular problem instance Bäck (1996). The calibration and fine-tuning of the control parameters is a major issue in metaheuristics modeling. To this end, there are two major approaches, namely the (offline) tuning of parameters prior to the algorithm's application and the (online) dynamic adaptation of parameters during its run Eiben et al. (1999); Eiben and Smit (2011).

The first approach is based on a preprocessing phase that attempts to identify the most suitable parameter setting of the algorithm on the specific problem instance through preliminary experimentation. The detected promising settings are then adopted for complete experimentation. More sophisticated techniques such as F-Race, Sequential Model-Based Optimization, and ParamILS Hoos (2011) have been used with promising results at the cost of additional implementation effort or computational burden.

Alternatively, dynamic parameter adaptation allows the algorithm to identify suitable parameter settings during its run. These approaches are based on the algorithm's hitherto performance, as well as on possible subsequent performance estimations Eiben and Smit (2011); Hoos (2011). Typically, they need minimal user intervention, although at the cost of additional computational requirements in terms of running time.

Published online: 04 November 2015

In the present paper, we propose a grid-based technique for dynamically adjusting the control parameters of a popular metaheuristic algorithm, namely Differential Evolution (DE) Price et al. (2005). The proposed technique is applicable in any similar optimization algorithm. Nevertheless, DE was selected mainly due to its recognized sensitivity in parameter values and operator type Qing (2009), which results in challenging parameter control problems.

The main goal of the proposed technique is the dynamic adaptation of the control parameters and crossover type of the DE algorithm, during its run. This is achieved by discretizing the parameters' search space, forming a grid. Then, a descent local search is conducted on the grid, based on estimations of DE's performance under the corresponding parameters. The most promising parameter setting is then adopted for a number of iterations in the DE algorithm. The same steps are iteratively applied in order to identify promising parameter settings in different stages of the optimization procedure. The search can be conducted either serially or in parallel, taking advantage of modern computer systems.

The resulting DE variants were validated on the established test suite of the Special Issue on Large Scale Continuous Optimization Problems of the Soft Computing journal Lozano et al. (2011). This test suite consists of high-dimensional (up to dimension 500) benchmark problems, including problems from other test suites (CEC 2008) as well as composite functions. Since such cases of complex and demanding problems, where parameter tuning is laborious, constitute the most appealing field of application for the proposed approaches, we selected the specific test suite for experimentation and comparisons.

The rest of the paper is organized as follows: Sect. 2 offers the necessary background information, i.e., description of the basic DE algorithm as well as literature review on adaptive DE variants. Section 3 is devoted to the proposed approach, while Sect. 4 presents the experimental setting and results, along with comparisons with other algorithms. Finally, Sect. 5 concludes the paper.

2 Background Information

In the following paragraphs, the basic DE algorithm is described along with its main operator types. Subsequently, an overview of relevant developments are given.

2.1 Differential Evolution

Differential Evolution (DE) was introduced by Storn and Price (1997) as a population-based, stochastic optimization algorithm for numerical optimization problems. Along with Particle Swarm Optimization Parsopoulos and Vrahatis (2010), DE employs a different mechanism for producing new candidate solutions than the dominant probabilistic mechanisms of Evolutionary Algorithms (EAs). Specifically, it uses differences of the population's members to perturb existing candidate solutions. Similarly to EAs, mutation, crossover, and selection operators are applied to evolve the population. Its simplicity, efficiency, and effectiveness has placed DE among the most popular metaheuristics, counting a large number of research works Das and Suganthan (2011).

Algorithm 1 Pseudocode of the DE algorithm.

```
1:  INPUT: Population $P$; Population size $N$; Parameters $F$, $CR$; Maximum iterations $t_{max}$
2:  $initialize(P)$
3:  while $t < t_{max}$ do
4:      for $i = 1 : N$ do
5:          Choose mutually different indices $r_s$, $2 \leqslant s \leqslant 5$
6:          $u_i^{(t+1)} \leftarrow mutation\left(x_{r_s}^{(t)}, F\right)$   /* Use Eqs. (1)-(5) */
7:          $v_i^{(t+1)} \leftarrow crossover\left(x_i^{(t)}, u_i^{(t+1)}, CR\right)$   /* Use Eq. (6) */
8:          $evaluate\left(v_i^{(t+1)}\right)$
9:          $x_i^{(t+1)} \leftarrow selection\left(x_i^{(t)}, v_i^{(t+1)}\right)$   /* Use Eq. (7) */
10:     end for
11:     Update index of best individual $g$
12:     $t \leftarrow t + 1$
13: end while
```

Let $f : X \rightarrow \mathbb{R}$ be the objective function under consideration and $X \subset \mathbb{R}^n$ be its domain. DE employs a population,

$$P = \{x_1, x_2, \ldots, x_N\},$$

of size N to probe the search space. Each individual x_i is an n-dimensional vector,

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{in})^\top \in X, \quad i \in I = \{1, 2, \ldots, N\},$$

and constitutes a candidate solution of the problem. The population is randomly initialized in X, typically following the uniform distribution.

The population P is iteratively evolved by applying two operators, namely *mutation* and *crossover*, on each individual. Then, a *selection* phase takes place, where each individual of the new population competes with its corresponding original individual, and the best one passes to the next iteration. These operators are iteratively applied until a termination condition is satisfied. The steps are outlined in Algorithm 1.

According to the mutation operator, a new vector $u_i^{(t+1)}$ is derived for each individual $x_i^{(t)}$ at iteration t, through combinations of existing individuals. Various forms have been proposed for this task, the most common ones being:
DE/Best/1

$$u_i^{(t+1)} = x_g^{(t)} + F\left(x_{r_1}^{(t)} - x_{r_2}^{(t)}\right), \quad (1)$$

DE/Rand/1

$$u_i^{(t+1)} = x_{r_1}^{(t)} + F\left(x_{r_2}^{(t)} - x_{r_3}^{(t)}\right), \tag{2}$$

DE/Current-to-Best/2

$$u_i^{(t+1)} = x_i^{(t)} + F\left(x_g^{(t)} - x_i^{(t)} + x_{r_1}^{(t)} - x_{r_2}^{(t)}\right), \tag{3}$$

DE/Best/2

$$u_i^{(t+1)} = x_g^{(t)} + F\left(x_{r_1}^{(t)} - x_{r_2}^{(t)} + x_{r_3}^{(t)} - x_{r_4}^{(t)}\right), \tag{4}$$

DE/Rand/2

$$u_i^{(t+1)} = x_{r_1}^{(t)} + F\left(x_{r_2}^{(t)} - x_{r_3}^{(t)} + x_{r_4}^{(t)} - x_{r_5}^{(t)}\right), \tag{5}$$

where F is a fixed, user-defined parameter, also called *scale factor* and it usually lies in the range (0, 1] Das and Suganthan (2011); g denotes the index of the best individual, i.e., the one with the lowest function value; and r_s are mutually different integers from the set I, also different than i.

After mutation, crossover is applied to produce a *trial vector*,

$$v_i = (v_{i1}, v_{i2}, \ldots, v_{in})^\top, \quad i \in I,$$

for each individual x_i. There are two main types of crossover operator. The most popular one is the *binomial crossover* where,

$$v_{ij}^{(t+1)} = \begin{cases} u_{ij}^{(t+1)}, & \text{if } \mathcal{R} \leqslant CR \text{ or } j = RN(n), \\ x_{ij}^{(t)}, & \text{otherwise,} \end{cases} \tag{6}$$

where $j \in \{1, 2, \ldots, n\}$; $\mathcal{R}$ is the realization of a uniformly distributed random variable in the range [0, 1]; $CR \in (0, 1]$ is the second control parameter of the algorithm, called *crossover rate*; and $RN(n)$ is an integer randomly selected from the set $\{1, 2, \ldots, n\}$. The two conditions in the first branch of Eq. (6) ensure that at least one component of the mutated vector $u_i^{(t+1)}$ is inherited to the trial vector.

The alternative, *exponential crossover* operator initially copies $x_i^{(t)}$ into the trial vector $v_i^{(t+1)}$. Subsequently, it randomly selects a component index $k \in \{1, 2, \ldots, n\}$ and sets the corresponding component $v_{ik}^{(t+1)} = u_{ik}^{(t+1)}$. Then, starting from the index $k + 1$, a number of components of $v_i^{(t+1)}$ are assigned the corresponding component values of $u_i^{(t+1)}$, based on stochastic condition. After the first failure of the condition, the rest of the components retain the values initially copied from $x_i^{(t)}$ Price et al. (2005).

Finally, selection takes place where the trial vector $v_i^{(t+1)}$ competes with $x_i^{(t)}$ and the new individual for the next iteration of the algorithm is selected as follows,

$$x_i^{(t+1)} = \begin{cases} v_i^{(t+1)}, & \text{if } f\left(v_i^{(t+1)}\right) \leqslant f\left(x_i^{(t)}\right), \\ x_i^{(t)}, & \text{otherwise.} \end{cases} \tag{7}$$

The algorithm iteratively applies the same procedure until a termination condition is reached. Eventually, the individual x_g is reported as the best detected solution.

The parameters of DE (including its crossover operator type) have been shown to be crucial for its convergence Qing (2009). While proper parameter values can render DE a very efficient algorithm, mild perturbations may result in significantly inferior performance. Taking into consideration that proper parameter values are typically problem-dependent, it is reasonable to expect that parameter tuning of DE can be a laborious task.

2.2 Relevant Works

A number of DE variants with dynamically adjusted parameters have been proposed in literature. In Brest et al. (2006) proposed a self-adapting algorithm that assigns control parameter values with probability. Zhao et al. (2011) proposed a self-adaptive scheme with multi-trajectory search. Also, a distributed adaptive scheme with scale factor inheritance was proposed in Weber et al. (2010), where subpopulations are connected in a ring topology, each one having its own scale factor. Moreover, several serial DE adaptation mechanisms were proposed to improve its search ability in Eiben et al. (1999); Qin et al. (2009); Zaharie and Petcu (2005).

In the same vein, the JADE variant with optional external archive and adaptive control parameters was proposed by Zhang and Sanderson (2009). Poláková et al. (2014) introduced a controlled restart DE in which the restarting conditions are derived from the difference of extremal values of the objective function and the estimated maximum distance among the points in the current population.

Tanabe and Fukunaga (2013) proposed a new parameter adaptation technique (called SHADE), which uses a historical memory of successful control parameter settings to guide the selection of future values. Moreover, Tanabe and Fukunaga (2014) introduced an enhanced version of SHADE, called L-SHADE, which incorporates success-history based parameter adaptation.

Tvrdík also adapted the control parameters of the DE in Tvrdík (2006), while a new adaptive variant with twelve competing strategies was proposed in Tvrdík and Poláková (2013). A comparative analysis of the binomial and exponential crossover variants was conducted in Zaharie (2007) and Zaharie (2009), providing also theoretical results. Brest et al. (2012) proposed a self-adaptive DE with small, varying population size.

Also, LaTorre et al. (2012) have analyzed the behavior of a hybrid algorithm that combines two heuristics. An adaptive memetic DE with global and local neighborhood-based mutation operators was proposed by Piotrowski (2013). Finally, in Segura et al. (2015) studied the relation between exploration and exploitation with respect to the scale factor in DE.

Most of the aforementioned DE variants were shown to be superior to their non-adaptive predecessors, achieving satisfactory balance between exploration and exploitation through parameter adaptation.

3 The Proposed Approach

In the following paragraphs, the proposed technique is presented in detail. Its basic scheme for the parameter adaptation of DE is initially exposed, followed by an enhanced version that dynamically adjusts also the crossover operator type (binomial/exponential).

3.1 Basic Scheme

The first step in the proposed technique is the discretization of DE's parameter space. Recall that both parameters F and CR (scale factor and crossover rate) take real values in the range $(0, 1]$. Although different ranges for F have been studied with promising results, the specific one appears to be the most common default choice in relevant literature Das and Suganthan (2011) as well as in reusable source codes freely available on the internet Price and Storn (2009); Takahama (1997). Nonetheless, adaptation for different ranges is trivial.

Discretization step sizes λ_F and λ_{CR} are specified for the two parameters, respectively. Small step sizes offer more refined search in parameter space, albeit transitions become slower. Thus, higher computational budget (running time) is typically needed to identify proper parameter values. On the other hand, large discretization step sizes may result in surpassing appropriate parameter values. Moreover, the observed performance differences of DE for marginally different parameter values can be simply the outcome of random fluctuations.

Our experience has shown that the performance differences are marginal between DE variants with parameter difference smaller than 0.1. For this reason, in the present work we selected $\lambda_F = \lambda_{CR} = 0.1$, as a reference step size to build the discretized 2-dimensional parameter space,

$$\mathcal{G} = \{(F, CR);\ \ F, CR, \in \{0.0, 0.1, \ldots, 1.0\}\},$$

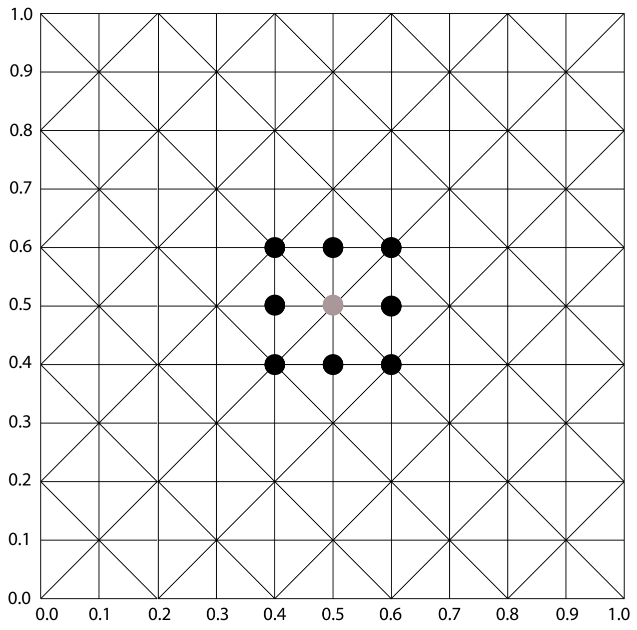


Fig. 1 The grid for $\lambda_F = \lambda_{CR} = 0.1$. Each interior point (grey node) has 8 immediate neighbors (black nodes)

which is henceforth called the *grid*. Each interior point in $\mathcal{G}$ has 8 immediate neighbors as illustrated in Fig. 1.

The algorithm starts with a randomly initialized population, called the *primary population*, which assumes the parameter pair at the center of the grid, i.e., $(F, CR) = (0.5, 0.5)$. This selection is reasonable in lack of additional domain knowledge suggesting better initial parameter values. Naturally, if such information is available, it can be easily exploited to accelerate the detection of suitable parameter values. For example, consider the case where a priori data is available suggesting that the objective function has a multitude of local minimizers, densely distributed on specific parts of its domain. In view of such information, the user can correspondingly bias the initial parameter pair in the grid towards lower values. This choice would be favorable, since large mutation and crossover steps may prohibit the algorithm from carefully probing the domain areas of interest. On the other hand, small number of sparsely distributed local minima suggests the choice of larger parameter values. Thus, further domain knowledge can be beneficial to convergence speed.

The primary population is evolved according to the standard DE procedure for t_p iterations. This is called the *dynamic's deployment* phase of the algorithm. The resulting population P is then copied in 8 *secondary populations* $P'_{a,b}$ with $a, b \in \{-1, 0, 1\}$, one for each neighboring parameter pair in the grid. If (F, CR) is the current parameter pair for the primary population P, then the secondary population $P'_{a,b}$ has a parameter pair (F', CR') with,

$$F' = F + a\,\lambda_F, \quad CR' = CR + b\,\lambda_{CR}, \quad a, b \in \{-1, 0, 1\}. \tag{8}$$

Obviously, the case $a = b = 0$ corresponds to the primary population and, thus, excluded.

Each secondary population is evolved according to the standard DE procedure for $t_s \ll t_p$ iterations. This is called the *performance estimation* phase of the algorithm, and gives information on the algorithm's performance for the current primary population, using the specific neighboring parameter values. Naturally, this step can be executed either serially or in parallel using one master node (primary population) and 8 slave nodes (secondary populations). Today, most modern desktop computers offer adequate resources for such implementations.

Subsequently, the primary and the secondary populations are compared in terms of their *average objective values* (A-OV), defined as follows:

$$\bar{f}_P = \frac{1}{N} \sum_{i=1}^{N} f(x_i).$$

The best-performing population (either the primary or a secondary one) is selected as the new primary population and the whole procedure is repeated again by conducting t_p iterations on the new primary population and so on.

A complete application of the procedure is henceforth referred as a *cycle* of the algorithm to avoid confusion with the iterations conducted during the evolution of the primary and secondary populations. Thus, a cycle c involves t_p iterations of the primary population and $8 \times t_s$ iterations of the secondary populations, i.e., a total number of,

$$q_c = \left(t_p + 8\,t_s\right)\, N,$$

function evaluations, where N stands for the population size. Thus, we can *a priori* determine the maximum number of complete cycles,

$$c_{\max} = \left\lfloor \frac{q_{\max}}{q_c} \right\rfloor, \tag{9}$$

that will be performed by the algorithm for a prescribed maximum computational budget of $q_{\max}$ function evaluations, where $\lfloor . \rfloor$ is the floor function.

Alternative performance metrics can be used instead of AOV. A typical alternative is the use of each population's overall best objective value. However, this metric may become misleading because, for practical purpose, the number t_s of performance-estimation iterations shall be typically kept low (5 to 10 iterations) in order to spare computational budget. Thus, using solely the overall best value renders the procedure vulnerable to locally optimal solutions that may be rapidly discovered. This was also verified in our experiments.

Algorithm 2 Pseudocode of the basic DEGPA algorithm.

```
1: $initialize(P)$
2: $M \leftarrow 8$   /* Number of secondary populations */
3: while (not termination) do
4:     /* Dynamic's deployment phase */
5:     Evolve primary population $P$ for $t_p$ iterations.
6:     /* Performance estimation phase */
7:     for $i = 1 : M$ do
8:         Copy $P$ to secondary population $P'_i$.
9:         Assign parameter pair to $P'_i$ according to Eq. (8).
10:        Evolve $P'_i$ for $t_s$ iterations.
11:    end for
12:    /* Update primary population */
13:    Find the best-performing secondary population $P'_{best}$.
14:    if ($\bar{f}_P - \bar{f}_{P'_{best}} \geqslant \varepsilon_{\min}$) then
15:        $P \leftarrow P'_{best}$
16:        Replace worst individuals of $P$ with the overall bests of
                all $P'_i$ (if they are better).
17:    end if
18: end while
```

Moreover, in order to take full advantage of the discoveries of all secondary populations, we also consider the utilization of the overall best individual of each secondary population in the new primary population. Thus, the worst 8 individuals of the new primary population are replaced by the 8 overall bests of the secondary populations, if they have better values.

In the context of parameter space $\mathcal{G}$, the proposed technique produces a trajectory of parameter pairs by tracking improvements of estimated performance. However, even marginal improvements offered by the secondary populations result in adopting new parameter pairs that may be recalled in subsequent cycles of the procedure. This effect can produce undesirable cyclic or oscillating trajectories, which can be alleviated by imposing a minimal improvement threshold $\varepsilon_{\min}$ for accepting a new parameter pair. Thus, a secondary population $P'_{a,b}$ replaces the primary population P only if it holds that,

$$\bar{f}_P - \bar{f}_{P'_{a,b}} \geqslant \varepsilon_{\min} > 0.$$

The value of $\varepsilon_{\min}$ can be set taking into consideration the desired solution accuracy.

The proposed approach is outlined in Algorithm 2 and it is henceforth called *Differential Evolution with Grid-based Parameter Adaptation* (DEGPA). Steps 3-18 constitute a complete cycle of the algorithm.

The number of iterations t_p and t_s for the primary and secondary populations, respectively, can be set to fixed values. As previously mentioned, small values of 5 to 10 iterations are adequate for t_s since only rough performance estimations are required for the subpopulations. On the other hand, t_p shall be assigned higher values in order to allow the primary population with the selected parameter pair deploy its dynamic. Based on suggestions in literature Storn and Price (1997), the value $t_p = 10 \times n$, with n being the problem's dimension is considered as the default choice.

An alternative and more sophisticated approach is the dynamic adaptation of t_p from a minimum value $t_p^{\min}$ to a maximum value $t_p^{\max}$ during the algorithm's execution. The rationale behind it lies on the fact that at the latest stages of optimization the population is expected to have identified promising regions of the search space and, hence, longer running time for the dynamic's deployment phase can be beneficial. The simplest adaptation scheme is the *linear* one. Thus, if c denotes the current cycle of the algorithm and $c_{\max}$ denotes the maximum number of cycles determined by Eq. (9), then t_p can be linearly adapted between $t_p^{\min}$ and $t_p^{\max}$ as follows:

$$t_p(c) = \left(t_p^{\max} - t_p^{\min}\right) \frac{c}{c_{\max}} + t_p^{\min}. \tag{10}$$

The extremal values $t_p^{\min}$ and $t_p^{\max}$ can be set by the user taking into consideration the problem at hand (especially its dimension). Our experience through experimentation on numerous problems has shown that setting $t_p^{\max}$ at values $40\% - 50\%$ higher than $t_p^{\min}$ is a good default choice. Thus, we can suggest the following default setting,

$$t_p^{\min} = 10 \times n, \qquad t_p^{\max} = 14 \times n,$$

where n is the problem's dimension.

Note that these parameters as well as the $\varepsilon_{\min}$ threshold are optional. Hence, they do not contradict the main goal of the proposed approach to exonerate the user from the burden of finding proper parameter values for the DE algorithm. The proposed approach can be straightforwardly parallelized in modern desktop computers by using a master node for the primary population and 8 slave nodes for the concurrent evolution of the secondary populations. Naturally, additional nodes can further expedite the procedure, especially when denser grids are used.

3.2 Enhanced Scheme

In addition to the dynamic parameter adaptation, the basic DEGPA scheme can be enhanced by considering also the dynamic adaptation of the crossover operator type between binomial and exponential. This can be achieved by using an additional secondary population, henceforth called the *bridge population*, in the performance-estimation phase of the algorithm.

More specifically, the bridge population executes t_s iterations (as the rest of the secondary populations) on the primary population, assuming the same parameter set as the primary population but with complementary crossover type. Thus,

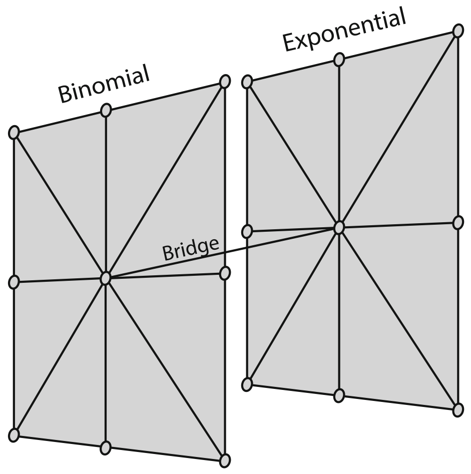


Fig. 2 Jumping from binomial to exponential grid

if the primary population P has binomial crossover operator and parameters (F, CR), the bridge population P'_{bridge} is initially a copy of P with parameters (F, CR) and the corresponding exponential crossover operator. After evolving P'_{bridge} for t_s iterations, the resulting population competes with the rest of the secondary populations and the primary one, according to the basic DEGPA scheme.

If the bridge population is the winner, then it becomes the primary population and its operator type becomes the current one for the subsequent cycles of the algorithm, until the next possible change. This procedure can be simply considered as a jump from the grid of the binomial to the grid of the exponential crossover operator and vice versa, as illustrated in Fig. 2. It is worth mentioning that jumping from one parameter space to the other normally requires recomputation of CR Zaharie (2007). This procedure automatically takes place in our scheme because, at every iteration, the algorithm recomputes both CR and F in the current parameter space.

The new variant is henceforth called *enhanced DEGPA* (eDEGPA) and it is outlined in Algorithm 3. The eDEGPA approach offers additional flexibility than DEGPA by further reducing the number of user-defined parameters in DE. The reason for presenting both DEGPA and eDEGPA is that, as we will see later in the experimental assessment, there are problem types where efficient crossover operator type has been identified in previous works. Thus, it is interesting and challenging to compare the performance between DEGPA incorporating this information and eDEGPA with no prior problem-dependent information.

4 Experimental Assessment

The performance of the proposed DEGPA and eDEGPA approaches was assessed on the established test suite pub-

Algorithm 3 Pseudocode of the eDEGPA algorithm.

```
1: $initialize(P)$
2: $M \leftarrow 9$   /* Number of secondary populations */
3: while not termination do
4:     /* Dynamic's deployment phase */
5:     Evolve primary population $P$ for $t_p$ iterations.
6:     /* Performance estimation phase */
7:     for $i = 1 : M$ do
8:         Copy $P$ to secondary population $P_i'$.
9:         if $i < M$ then
10:            /* Normal secondary population */
11:            Assign parameter pair to $P_i'$ according to Eq. (8).
12:        else
13:            /* Bridge population */
14:            Change crossover operator type.
15:        end if
16:        Evolve $P_i'$ for $t_s$ iterations.
17:    end for
18:    /* Update primary population */
19:    Find the best-performing secondary population $P_{best}'$.
20:    if $\bar{f}_P - \bar{f}_{P_{best}'} \geqslant \varepsilon_{\min}$ then
21:        $P \leftarrow P_{best}'$
22:        Replace worst individuals of $P$ with the overall bests of
               all $P_i'$ (if they are better).
23:    end if
24: end while
```

lished in the *Special Issue on Large Scale Continuous Optimization Problems* of the *Soft Computing* journal Lozano et al. (2011). The test suite consists of 19 scalable test problems, henceforth denoted as F1-F19. The problems F1-F6 come from the CEC 2008 test suite Tang et al. (2007), while F7-F11 are shifted functions and F12-F19 are hybrid composition functions Lozano et al. (2011).

The test suite recommends three algorithms, namely DE with exponential crossover, CHC Eshelman (1993), and GCMAES Auger and Hansen (2005) as the basic ones for comparisons. Also, averaged results are provided for 13 additional algorithms, namely SOUPDE Weber et al. (2011), DE-D^{40}+M^{m} García-Martínez et al. (2011), GaDE Yang et al. (2011), jDElscop Brest and Maucec (2011), SaDE-MMTS Zhao et al. (2011), MOS LaTorre et al. (2011), MA-SSW-Chains Molina et al. (2011), RPSO-vm García-Nieto and Alba (2011), Tuned IPSOLS de Oca et al. (2011), Evo-PROpt Duarte et al. (2011), EM323 Gardeux et al. (2011), VXQR1 Neumaier et al. (2011), and GODE Wang et al. (2011). The software package is freely available on the internet Lozano et al. (2010).

DEGPA and eDEGPA aim at relieving the user from the burden of parameter setting in DE. Thus, the main pursuance in our experiments was to achieve competitive average performance with the algorithms reported in the test suite, especially with the exponential DE approach, which was optimally tuned on the specific test problems. The DEGPA approach adopted the exponential crossover operator as well the mutation of Eq. (2), according to the setting of the base DE algorithm. On the other hand, eDEGPA was let to dynamically select between binomial and exponential crossover, while its mutation operator was identical to DEGPA.

We conducted experiments with both DEGPA and eDEGPA on all test problems, for dimensions $n = 50$, 100, 200, and 500. The available computational budget was determined, according to the test suite's requirements, as $5000 \times n$ function evaluations Lozano et al. (2011). At this point, we shall underline that this budget does not include the (usually significant) preliminary experiments that are necessary for identifying good parameter settings for all the algorithms of the test suite. This computational budget is typically neglected in all relevant studies, although it can be comparable or even higher than the reported budgets required for solving a problem. Our approaches do not require such preliminary experimentation. Although it would be fair to assign our approaches this additional preprocessing budget, we decided to push their performance to the limit and assess them with exactly the same budget as for the other algorithms.

The implementation was made in C programming language, and the OpenMPI library[1] was used for parallel evolution of the secondary populations under a typical master-slave model. Specifically, one subpopulation was assigned per slave node, while the master node was used for the main procedure of the algorithm. The parallelization does not interfere with the algorithm's dynamic but only expedites the experiments. Thus, the same results are received with the serial version of the algorithm under identical initial conditions and seeding. All experiments were conducted on Intel® i7 machines (each one providing 8 CPUs) with 8GB RAM, running under Ubuntu Linux 14.04.

The considered performance measure was the objective value error,

$$f\left(x^*\right) - f\left(x_{\text{opt}}\right),$$

where x^* is the solution achieved by the algorithm and x_{opt} is the optimal solution of the problem. For each algorithm, 25 independent experiments were conducted and the average errors were recorded. According to the setting of the DE algorithm in the employed test suite, the population size for both DEGPA and eDEGPA was set to $N = 60$.

Although the test suite includes only the exponential DE variant, for the sake of completeness, we evaluated also the corresponding binomial DE variant using the provided settings and source codes in Lozano et al. (2010). The control parameters for the two base DE variants were set to $(F, CR) = (0.7, 0.5)$ as suggested in Lozano et al. (2010). The two base DE variants are henceforth denoted as DE$_{\text{exp}}$ and DE$_{\text{bin}}$. On the other hand, both DEGPA and eDEGPA were initiated at the central parameter pair $(F, CR) = (0.5, 0.5)$, and the initial crossover operator type for eDEGPA

[1] http://www.open-mpi.org/

Table 1 Average errors and standard deviations for the proposed and base algorithms, for dimension $n = 50$ and 100

Problem	DEGPA		eDEGPA		DE$_{\text{bin}}$		DE$_{\text{exp}}$		CHC		GCMAES	
	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD
50-dimensional												
$F1$	$8.87e-14$	$3.31e-14$	$5.46e-14$	$1.14e-14$	$3.00e-17$	$7.69e-18$	$2.78e-17$	$6.29e-33$	$2.90e+02$	$5.69e+02$	$2.78e-17$	$6.29e-33$
$F2$	$4.48e+00$	$3.15e+00$	$3.19e+00$	$4.16e+00$	$3.87e+01$	$8.90e+00$	$3.31e-01$	$5.90e-02$	$7.72e+01$	$1.23e+01$	$7.69e-11$	$4.83e-11$
$F3$	$4.59e+01$	$1.18e+01$	$5.56e+01$	$2.59e+01$	$6.99e+01$	$3.58e+01$	$3.10e+01$	$8.65e+00$	$5.64e+07$	$1.42e+08$	$6.38e-01$	$1.49e+00$
$F4$	$1.30e-13$	$2.60e-14$	$6.60e-11$	$3.29e-10$	$3.21e+01$	$1.38e+01$	$4.79e-02$	$2.01e-01$	$1.12e+02$	$2.74e+01$	$3.72e+02$	$8.68e+01$
$F5$	$4.55e-14$	$1.64e-14$	$2.73e-14$	$5.68e-15$	$9.86e-04$	$2.76e-03$	$0.00e+00$	$0.00e+00$	$9.02e-01$	$1.82e+00$	$2.16e-01$	$5.64e-01$
$F6$	$2.07e-13$	$7.32e-14$	$1.47e-13$	$9.83e-14$	$7.16e-14$	$1.86e-14$	$1.39e-13$	$9.43e-15$	$3.23e+00$	$2.44e+00$	$1.90e+01$	$1.02e+00$
$F7$	$5.53e-14$	$2.09e-13$	$0.00e+00$	$0.00e+00$	$2.22e-15$	$1.17e-15$	$8.88e-17$	$1.96e-16$	$1.23e-09$	$1.45e-09$	$2.10e+01$	$1.38e+01$
$F8$	$5.94e+01$	$1.16e+02$	$1.86e+02$	$2.59e+02$	$9.02e+10$	$0.00e+00$	$9.02e+10$	$0.00e+00$	$9.02e+10$	$9.02e+06$	$9.03e+10$	$9.39e+07$
$F9$	$1.30e-04$	$6.07e-04$	$6.58e-05$	$2.98e-04$	$2.85e+02$	$5.30e+00$	$2.73e+02$	$7.40e-01$	$3.11e+02$	$4.98e+00$	$3.16e+02$	$7.03e+00$
$F10$	$3.30e-28$	$1.08e-27$	$9.12e-27$	$2.65e-26$	$1.53e+00$	$1.29e+00$	$6.50e-29$	$3.60e-29$	$7.72e+00$	$2.93e+00$	$9.25e+00$	$2.82e+00$
$F11$	$6.98e-05$	$2.81e-04$	$7.25e-05$	$3.05e-04$	$9.65e-01$	$2.02e+00$	$6.26e-05$	$1.30e-05$	$1.01e-02$	$1.26e-02$	$1.95e+02$	$3.65e+01$
$F12$	$1.08e-08$	$5.30e-08$	$1.71e-28$	$6.56e-28$	$5.82e+00$	$1.03e+01$	$5.26e-13$	$1.64e-13$	$8.23e+01$	$1.53e+02$	$1.14e+02$	$1.01e+01$
$F13$	$2.97e+01$	$4.79e+00$	$5.02e+01$	$3.29e+01$	$5.97e+01$	$2.22e+01$	$2.48e+01$	$1.31e+00$	$1.43e+07$	$3.29e+07$	$1.16e+02$	$1.43e+01$
$F14$	$3.15e-08$	$8.46e-08$	$3.34e-06$	$1.30e-05$	$3.35e+01$	$1.86e+01$	$3.55e-08$	$2.26e-08$	$6.76e+01$	$1.30e+01$	$2.71e+02$	$7.30e+01$
$F15$	$3.98e-13$	$1.99e-12$	$0.00e+00$	$0.00e+00$	$2.29e-01$	$6.07e-01$	$1.99e-24$	$3.22e-24$	$3.07e+00$	$5.32e+00$	$3.94e+01$	$1.25e+02$
$F16$	$1.04e-06$	$4.96e-06$	$2.72e-09$	$1.36e-08$	$5.64e+00$	$8.47e+00$	$1.56e-09$	$2.81e-10$	$5.60e+01$	$5.16e+01$	$2.23e+02$	$1.50e+01$
$F17$	$2.32e+00$	$2.84e+00$	$7.44e+00$	$3.19e+00$	$1.51e+01$	$1.43e+01$	$8.52e-01$	$4.92e-01$	$7.61e+06$	$2.44e+07$	$3.47e+02$	$2.18e+01$
$F18$	$9.50e-07$	$2.96e-06$	$4.67e-06$	$9.95e-06$	$5.73e+00$	$5.26e+00$	$1.28e-04$	$4.63e-05$	$6.76e+01$	$3.46e+01$	$3.59e+02$	$8.45e+01$
$F19$	$9.44e-23$	$4.71e-22$	$0.00e+00$	$0.00e+00$	$1.23e+00$	$9.26e-01$	$2.00e-24$	$1.50e-24$	$1.95e+02$	$5.01e+02$	$1.71e+03$	$5.84e+03$

Table 1 continued

Problem	DEGPA		eDEGPA		DE_{bin}		DE_{exp}		CHC		GCMAES	
	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD
100-dimensional												
$F1$	$2.34e-13$	$5.76e-14$	$5.68e-14$	$3.86e-29$	$1.12e-16$	$4.28e-17$	$7.77e-17$	$1.13e-17$	$4.67e+02$	$7.02e+02$	$5.55e-17$	$1.26e-32$
$F2$	$1.60e+01$	$7.54e+00$	$2.90e+01$	$1.50e+01$	$7.74e+01$	$7.77e+00$	$4.60e+00$	$4.24e-01$	$9.96e+01$	$1.16e+01$	$2.61e-03$	$1.30e-02$
$F3$	$1.10e+02$	$2.85e+01$	$1.36e+02$	$4.96e+01$	$4.43e+02$	$3.63e+02$	$8.01e+01$	$1.03e+01$	$1.52e+08$	$2.69e+08$	$1.23e+01$	$1.80e+01$
$F4$	$3.16e-13$	$5.93e-14$	$4.64e-13$	$8.17e-13$	$1.01e+02$	$2.25e+01$	$9.53e-03$	$4.76e-02$	$2.92e+02$	$5.16e+01$	$8.38e+02$	$1.39e+02$
$F5$	$1.15e-13$	$2.90e-14$	$3.30e-14$	$1.34e-14$	$2.93e-02$	$5.32e-02$	$2.55e-17$	$5.19e-18$	$5.95e+00$	$1.29e+01$	$2.68e+00$	$1.05e+01$
$F6$	$4.12e-13$	$9.61e-14$	$1.96e-13$	$1.50e-13$	$1.55e+00$	$3.88e-01$	$3.10e-13$	$1.62e-14$	$4.79e+00$	$1.87e+00$	$1.86e+01$	$2.45e+00$
$F7$	$6.10e-15$	$2.69e-14$	$1.53e-15$	$4.35e-15$	$1.39e-14$	$7.12e-15$	$3.80e-17$	$5.29e-17$	$8.67e-02$	$3.70e-01$	$6.35e+01$	$2.36e+01$
$F8$	$9.82e+02$	$1.46e+03$	$2.55e+03$	$3.34e+03$	$1.79e+11$	$0.00e+00$	$1.79e+11$	$0.00e+00$	$1.79e+11$	$1.92e+07$	$1.80e+11$	$3.54e+08$
$F9$	$3.85e-04$	$7.11e-04$	$1.07e-03$	$2.29e-03$	$5.43e+02$	$1.36e+01$	$5.06e+02$	$9.16e-01$	$5.87e+02$	$1.01e+01$	$6.08e+02$	$1.07e+01$
$F10$	$1.08e-26$	$5.39e-26$	$3.53e-30$	$1.55e-29$	$1.54e+01$	$3.31e+00$	$1.35e-28$	$3.86e-29$	$2.89e+01$	$1.01e+01$	$1.93e+01$	$5.10e+00$
$F11$	$6.60e-04$	$2.17e-03$	$1.01e-03$	$2.65e-03$	$4.31e+01$	$2.09e+01$	$1.25e-04$	$1.43e-05$	$2.80e+01$	$3.02e+01$	$4.82e+02$	$4.27e+01$
$F12$	$2.31e-07$	$9.00e-07$	$1.43e-02$	$7.15e-02$	$7.21e+01$	$3.21e+01$	$6.44e-11$	$1.52e-11$	$8.72e+02$	$2.55e+03$	$2.41e+02$	$1.23e+01$
$F13$	$7.23e+01$	$1.89e+01$	$9.07e+01$	$3.09e+01$	$2.76e+02$	$6.18e+01$	$6.13e+01$	$1.00e+00$	$9.37e+07$	$4.02e+08$	$2.59e+02$	$2.16e+01$
$F14$	$3.58e-08$	$8.99e-08$	$1.78e-06$	$2.78e-06$	$9.37e+01$	$1.56e+01$	$4.48e-02$	$2.24e-01$	$2.25e+02$	$4.59e+01$	$6.19e+02$	$9.25e+01$
$F15$	$2.09e-15$	$1.04e-14$	$5.97e-10$	$2.07e-09$	$3.67e+00$	$1.76e+00$	$7.10e-23$	$7.00e-23$	$5.99e+00$	$1.19e+01$	$5.57e+01$	$5.22e+01$
$F16$	$3.05e-06$	$1.16e-05$	$1.48e-08$	$4.17e-08$	$1.10e+02$	$3.80e+01$	$1.94e-02$	$9.70e-02$	$2.08e+02$	$1.49e+02$	$4.84e+02$	$2.08e+01$
$F17$	$2.27e+01$	$2.76e+01$	$3.43e+01$	$2.64e+01$	$1.78e+02$	$5.49e+01$	$1.19e+01$	$2.62e+00$	$4.36e+07$	$7.09e+07$	$7.04e+02$	$3.92e+01$
$F18$	$2.56e-06$	$1.23e-05$	$8.40e-06$	$2.40e-05$	$1.04e+02$	$4.39e+01$	$2.92e-04$	$6.77e-05$	$2.37e+02$	$7.02e+01$	$1.09e+03$	$4.15e+02$
$F19$	$2.69e-22$	$1.34e-21$	$1.18e-31$	$4.83e-31$	$1.17e+01$	$2.61e+00$	$4.79e-23$	$2.65e-23$	$4.70e+02$	$1.84e+03$	$5.83e+03$	$9.85e+03$

Table 2 Average errors and standard deviations for the proposed and base algorithms, for dimension $n = 200$ and 500

Problem	DEGPA		eDEGPA		DE_{bin}		DE_{exp}		CHC		GCMAES	
	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD
200-dimensional												
$F1$	$5.50e-13$	$1.07e-13$	$1.32e-13$	$8.32e-14$	$6.39e-16$	$5.32e-16$	$1.78e-16$	$1.60e-17$	$9.61e+02$	$1.65e+03$	$1.17e-16$	$1.13e-17$
$F2$	$3.67e+01$	$1.17e+01$	$6.22e+01$	$2.00e+01$	$1.01e+02$	$5.90e+00$	$1.89e+01$	$1.05e+00$	$1.17e+02$	$7.60e+00$	$7.47e-02$	$2.44e-01$
$F3$	$2.17e+02$	$3.18e+01$	$2.56e+02$	$8.22e+01$	$6.38e+02$	$3.80e+02$	$1.79e+02$	$8.89e+00$	$2.54e+08$	$3.97e+08$	$1.24e+02$	$8.85e+01$
$F4$	$8.00e-04$	$4.00e-03$	$4.90e-09$	$9.12e-09$	$4.21e+02$	$5.63e+01$	$8.52e-02$	$3.98e-01$	$6.32e+02$	$8.43e+01$	$1.57e+03$	$1.54e+02$
$F5$	$2.60e-13$	$2.80e-14$	$7.21e-07$	$3.61e-06$	$3.00e-01$	$7.73e-01$	$7.49e-17$	$6.94e-18$	$1.02e+01$	$1.59e+01$	$1.13e+00$	$2.84e+00$
$F6$	$8.39e-13$	$2.35e-13$	$3.73e-13$	$3.20e-13$	$5.28e+00$	$9.65e-01$	$6.46e-13$	$2.53e-14$	$8.14e+00$	$2.26e+00$	$1.93e+01$	$7.39e-01$
$F7$	$7.87e-14$	$3.74e-13$	$6.55e-15$	$2.35e-14$	$1.78e-11$	$4.65e-11$	$2.25e-16$	$1.92e-16$	$3.95e-01$	$1.21e+00$	$1.25e+02$	$1.92e+01$
$F8$	$1.10e+04$	$1.43e+04$	$2.79e+04$	$2.70e+04$	$8.33e+11$	$0.00e+00$	$8.33e+11$	$0.00e+00$	$8.33e+11$	$3.09e+08$	$8.56e+11$	$3.36e+09$
$F9$	$9.47e-05$	$3.00e-04$	$6.28e-03$	$1.48e-02$	$1.13e+03$	$1.87e+01$	$1.01e+03$	$1.30e+00$	$1.18e+03$	$8.29e+00$	$1.22e+03$	$1.79e+01$
$F10$	$1.62e-30$	$7.01e-30$	$1.63e-30$	$7.46e-30$	$5.52e+01$	$1.02e+01$	$2.77e-28$	$5.31e-29$	$7.34e+01$	$6.25e+01$	$3.76e+01$	$2.78e+01$
$F11$	$7.82e-04$	$2.26e-03$	$5.31e-03$	$1.24e-02$	$3.95e+02$	$6.11e+01$	$2.55e-04$	$3.20e-05$	$4.03e+02$	$8.45e+01$	$1.08e+03$	$8.25e+01$
$F12$	$1.49e-06$	$6.91e-06$	$3.72e-03$	$1.37e-02$	$2.84e+02$	$4.97e+01$	$9.97e-10$	$2.01e-10$	$8.11e+02$	$1.58e+03$	$5.87e+02$	$4.52e+02$
$F13$	$1.40e+02$	$1.68e+01$	$1.95e+02$	$6.12e+01$	$7.52e+02$	$2.71e+02$	$1.40e+02$	$1.26e+01$	$2.06e+08$	$3.51e+08$	$5.92e+02$	$1.08e+02$
$F14$	$1.55e-08$	$3.27e-08$	$1.38e-07$	$4.05e-07$	$3.11e+02$	$3.66e+01$	$8.08e-03$	$4.04e-02$	$4.90e+02$	$5.23e+01$	$1.26e+03$	$1.81e+02$
$F15$	$2.78e-18$	$1.39e-17$	$4.64e-14$	$2.07e-13$	$1.17e+01$	$2.85e+00$	$3.71e-24$	$2.32e-24$	$1.40e+01$	$9.80e+00$	$1.95e+02$	$1.66e+02$
$F16$	$1.71e-06$	$6.61e-06$	$1.33e-03$	$4.71e-03$	$5.58e+02$	$7.96e+01$	$7.85e-09$	$1.11e-09$	$6.77e+02$	$6.04e+02$	$9.56e+02$	$3.33e+01$
$F17$	$6.24e+01$	$3.58e+01$	$6.29e+01$	$3.10e+01$	$1.03e+03$	$1.11e+02$	$3.71e+01$	$8.30e-01$	$1.17e+07$	$1.70e+07$	$1.49e+03$	$8.00e+01$
$F18$	$1.06e-05$	$5.14e-05$	$1.67e-04$	$6.00e-04$	$7.53e+02$	$6.55e+01$	$5.10e-04$	$9.97e-05$	$7.67e+02$	$2.14e+02$	$3.94e+03$	$3.91e+03$
$F19$	$2.29e-17$	$1.15e-16$	$4.20e-02$	$2.10e-01$	$4.04e+01$	$7.71e+00$	$1.67e-22$	$7.58e-23$	$7.51e+02$	$1.76e+03$	$2.53e+04$	$2.45e+04$

Table 2 continued

Problem	DEGPA		eDEGPA		DE_{bin}		DE_{exp}		CHC		GCMAES	
	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD	Mean	StD
500-dimensional												
$F1$	$1.43e-12$	$1.58e-13$	$5.59e-13$	$2.32e-13$	$3.88e-05$	$7.93e-05$	$5.17e-16$	$1.36e-17$	$9.25e+02$	$1.27e+03$	n/a	n/a
$F2$	$7.97e+01$	$1.28e+01$	$7.31e+01$	$1.85e+01$	$1.25e+02$	$5.37e+00$	$5.38e+01$	$1.21e+00$	$1.35e+02$	$5.52e+00$	n/a	n/a
$F3$	$4.96e+02$	$2.01e+01$	$6.22e+02$	$1.69e+02$	$3.44e+04$	$1.62e+05$	$4.74e+02$	$1.48e+00$	$6.93e+08$	$1.72e+09$	n/a	n/a
$F4$	$2.08e-12$	$3.22e-13$	$3.98e-02$	$1.99e-01$	$2.35e+03$	$1.60e+02$	$7.12e-01$	$9.64e-01$	$2.11e+03$	$1.66e+02$	n/a	n/a
$F5$	$7.21e-13$	$1.00e-13$	$3.60e-13$	$1.79e-13$	$3.11e-01$	$5.07e-01$	$2.38e-16$	$1.18e-17$	$1.45e+01$	$2.52e+01$	n/a	n/a
$F6$	$2.08e-12$	$3.66e-13$	$1.24e-01$	$3.30e-01$	$1.49e+01$	$8.38e-01$	$1.64e-12$	$4.85e-14$	$1.27e+01$	$1.26e+00$	n/a	n/a
$F7$	$4.02e-18$	$2.01e-17$	$3.64e-03$	$1.82e-02$	$2.74e-03$	$6.49e-03$	$7.29e-16$	$3.58e-16$	$3.33e-05$	$1.12e-04$	n/a	n/a
$F8$	$9.55e+04$	$8.31e+04$	$1.77e+05$	$1.53e+05$	$4.94e+12$	$0.00e+00$	$4.94e+12$	$0.00e+00$	$4.94e+12$	$1.42e+08$	n/a	n/a
$F9$	$2.32e-03$	$4.18e-03$	$2.10e-02$	$6.44e-02$	$2.97e+03$	$3.17e+01$	$2.52e+03$	$2.10e+00$	$3.00e+03$	$1.64e+01$	n/a	n/a
$F10$	$0.00e+00$	$0.00e+00$	$1.45e-32$	$7.23e-32$	$1.36e+02$	$2.08e+01$	$9.79e-28$	$1.43e-28$	$1.64e+02$	$5.62e+01$	n/a	n/a
$F11$	$2.16e-03$	$4.28e-03$	$2.78e-03$	$6.39e-03$	$2.34e+03$	$9.22e+01$	$6.78e-04$	$3.60e-05$	$1.67e+03$	$1.44e+02$	n/a	n/a
$F12$	$4.24e-06$	$1.45e-05$	$1.66e-04$	$7.86e-04$	$1.02e+03$	$6.68e+01$	$6.80e-09$	$8.58e-10$	$1.62e+03$	$1.83e+03$	n/a	n/a
$F13$	$3.71e+02$	$2.14e+01$	$5.77e+02$	$1.77e+02$	$2.49e+03$	$3.13e+02$	$3.60e+02$	$9.23e+00$	$3.41e+08$	$4.29e+08$	n/a	n/a
$F14$	$1.60e-08$	$3.23e-08$	$3.30e-03$	$1.61e-02$	$1.67e+03$	$1.51e+02$	$3.93e-01$	$1.05e+00$	$1.59e+03$	$1.57e+02$	n/a	n/a
$F15$	$2.33e-15$	$8.81e-15$	$5.50e-01$	$1.77e+00$	$4.44e+01$	$5.59e+00$	$2.93e-18$	$7.16e-18$	$3.50e+01$	$1.20e+01$	n/a	n/a
$F16$	$1.71e-05$	$4.77e-05$	$1.46e+01$	$7.27e+01$	$2.02e+03$	$8.60e+01$	$2.05e-08$	$1.64e-09$	$1.92e+03$	$1.44e+03$	n/a	n/a
$F17$	$1.41e+02$	$3.61e+01$	$2.67e+02$	$5.51e+02$	$3.83e+03$	$1.41e+02$	$1.12e+02$	$1.02e+00$	$6.64e+08$	$1.64e+09$	n/a	n/a
$F18$	$7.67e-06$	$2.90e-05$	$1.19e-01$	$3.30e-01$	$3.37e+03$	$4.34e+02$	$1.25e-03$	$1.87e-04$	$2.74e+03$	$3.59e+02$	n/a	n/a
$F19$	$3.18e-27$	$9.97e-27$	$2.16e+00$	$3.41e+00$	$1.29e+02$	$2.34e+01$	$3.35e-21$	$2.15e-21$	$2.05e+03$	$4.03e+03$	n/a	n/a

was the exponential one (dynamically changing during execution).

The experimental analysis was divided in two phases. In the first phase, DEGPA and eDEGPA were compared against the base algorithms, namely DE_{exp}, DE_{bin}, CHC, and GCMAES. The available source code for each base algorithm was employed for conducting 25 independent experiments per problem, using the exact settings reported in the test suite. The achieved errors were recorded for each algorithm and experiment. The means and standard deviations of the obtained objective value errors for the proposed and the base algorithms are reported in Tables 1 and 2. In the 500-dimensional case, results could not be obtained with the provided GCMAES source code due to excessive computation time (this is reported as "n/a" in the Tables).

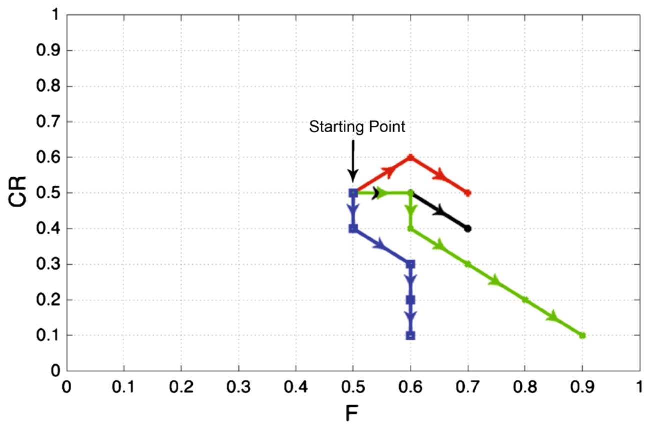
A close inspection of the results offers interesting information. Firstly, both the proposed approaches have competitive performance with the base algorithms. In fact, we can see that they clearly outperform DE_{bin}, CHC, and GCMAES, in almost all test problems, especially when dimension increases. Also, they show remarkably competitive performance against the best-performing base algorithm DE_{exp}. This behavior worths further attention if we take into consideration that DE_{exp} was used with its optimal parameter setting and crossover type provided in the test suite, for the same computational budget with our approaches.

Also, for lower dimensions ($n = 50$ and 100) eDEGPA has superior average performance than DEGPA. This can be attributed to eDEGPA's ability to decide on the most proper crossover type operator, since binomial crossover appears to be more beneficial in some test problems. However, the performance gradually declines as dimension increases. This is expected, since eDEGPA needs additional effort to evolve the extra subpopulation, in order to dynamically decide on the crossover operator type. Thus, as the problems become harder with dimension, eDEGPA exceeds the available budget more rapidly than DEGPA.

Table 3 Number of wins (denoted as "+"), loses (denoted as "-"), and ties (denoted as "=") of DEGPA and eDEGPA against the base algorithms of the test suite

Dimension	Algorithm	DEGPA			eDEGPA		
		+	−	=	+	−	=
50	DE_{bin}	14	3	2	14	3	2
	DE_{exp}	9	6	4	9	7	3
	CHC	19	0	0	19	0	0
	GCMAES	16	3	0	16	3	0
	eDEGPA	6	10	3			
100	DE_{bin}	17	1	1	17	1	1
	DE_{exp}	11	6	2	11	6	2
	CHC	19	0	0	19	0	0
	GCMAES	16	3	0	16	3	0
	eDEGPA	6	8	5			
200	DE_{bin}	17	1	1	17	1	1
	DE_{exp}	12	7	0	10	8	1
	CHC	19	0	0	19	0	0
	GCMAES	16	3	0	16	3	0
	eDEGPA	8	4	7			
500	DE_{bin}	19	0	0	19	0	0
	DE_{exp}	11	7	1	7	6	6
	CHC	19	0	0	19	0	0
	GCMAES	n/a	n/a	n/a	n/a	n/a	n/a
	eDEGPA	8	4	7			

In order to statistically verify the observed performance differences, pairwise comparisons using the Wilcoxon rank-sum test at confidence level 95 % were conducted for all test functions. Each positive comparison where DEGPA or eDEGPA outperformed another algorithm with statistical



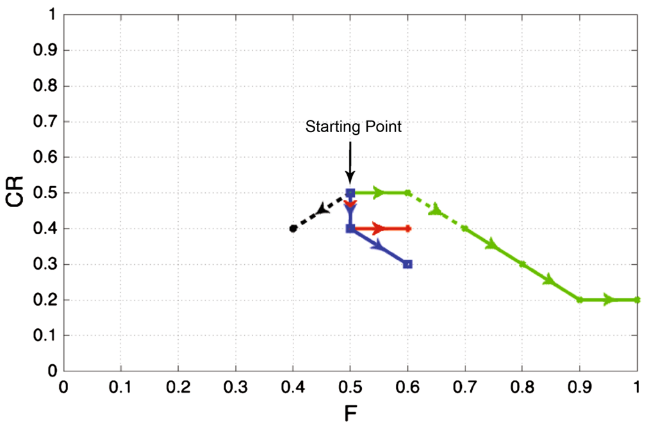


Fig. 3 Trajectories of parameter pairs for DEGPA (left) and eDEGPA (right) for four indicatively selected test problems. Different colors correspond to different problems. All trajectories start on the grid center $(F, CR) = (0.5, 0.5)$. For eDEGPA solid line indicates exponential crossover, while dashed line stands for binomial crossover

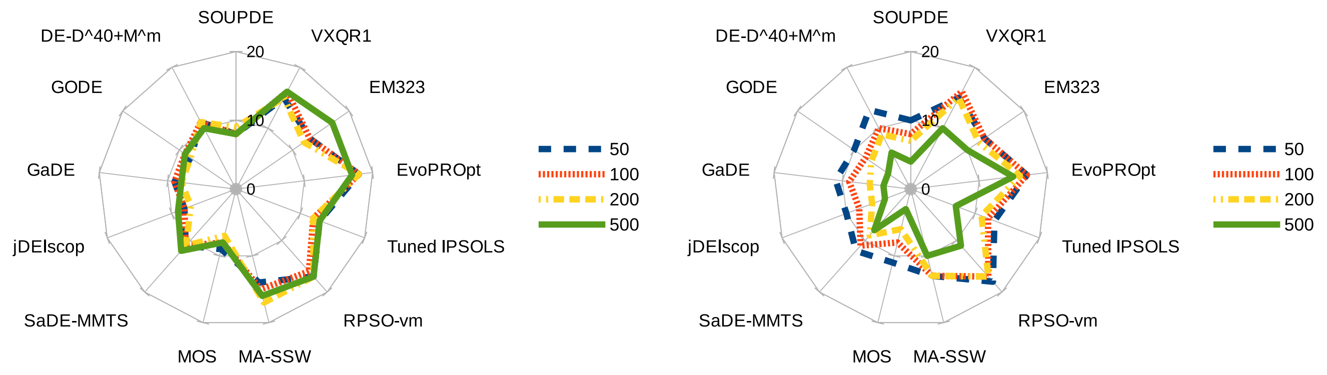


Fig. 4 Number of test problems where DEGPA (left) and eDEGPA (right) achieved equal or better average error than the other algorithms, for dimensions $n = 50$, 100, 200, and 500

significance was counted as a *win*. The corresponding negative comparisons were counted as *loses*. The lack of statistical significance was considered a *tie*. The results for all comparisons are given in Table 3, where wins, loses, and ties are denoted as "+", "−", and "=", respectively.

The reported results verify our previous findings. Specifically, both DEGPA and eDEGPA had statistically better or equivalent performance with the rest of the base algorithms in most of the test problems. Moreover, their numbers of wins exhibited increasing trend with dimension. Another interesting observation is that the strongest competitor, namely DE_{exp}, had always lower number of wins than both DEGPA and eDEGPA.

For all but the DE_{exp} case, eDEGPA had identical number of wins, loses, and ties with DEGPA against the other algorithms. This is remarkable, since eDEGPA has additional self-adaptation capabilities. Finally, the last line per dimension block in Table 3 reports the wins, loses, and ties of eDEGPA against DEGPA. These comparisons verify our previous observations on the superior performance of eDEGPA in lower dimension and its decline in higher dimension as a result of the extra effort imposed by the dynamic adaptation of the crossover operator type.

Finally, Fig. 3 illustrates trajectories of the parameter pairs in the grid, for four indicatively selected test problems (corresponding to lines of different colors). In the case of eDEGPA, solid lines correspond to exponential crossover operators while dashed lines correspond to binomial operators.

In the second phase of experimentation, DEGPA and eDEGPA were compared with a number of different algorithms, following the requirements of the employed test suite Lozano et al. (2010, 2011). The comparisons were based on the average error values per algorithm and test function, which are reported in Tables 5 and 6. The data for the rest of the algorithms are reproduced from the original sources Lozano et al. (2010) in both Tables. Also, the results

Table 4 Number of problems where DEGPA/eDEGPA exhibited inferior and non-inferior average solution values against the rest of the algorithms

Algorithm	Non-Inferior Dim.				Inferior Dim.			
	50	100	200	500	50	100	200	500
DEGPA								
EvoPROpt	18	18	18	17	1	1	1	2
EM323	13	13	12	17	6	6	7	2
SOUPDE	8	8	9	8	11	11	10	11
DE-D^{40}+M^m	11	11	11	10	8	8	8	9
GODE	8	9	8	9	11	10	11	10
MA-SSW-Chains	14	15	17	16	5	4	2	3
GaDE	9	9	8	8	10	10	11	11
RPSO-vm	17	16	17	17	2	3	2	2
jDElscop	8	8	7	9	11	11	12	10
SaDE-MMTS	11	11	11	12	8	8	7	5
MOS	9	8	7	8	10	11	12	11
Tuned IPSOLS	13	12	12	13	6	7	7	6
VXQR1	15	16	15	16	4	3	4	3
eDEGPA								
EvoPROpt	17	17	16	15	2	2	3	4
EM323	13	13	12	10	6	6	7	9
SOUPDE	10	8	7	4	9	11	12	15
DE-D^{40}+M^m	13	10	9	6	6	9	10	13
GODE	10	8	7	4	9	11	12	15
MA-SSW-Chains	13	13	13	10	6	6	6	9
GaDE	11	9	6	4	8	10	13	15
RPSO-vm	18	17	17	11	1	2	2	8
jDElscop	10	8	6	4	9	11	13	15
SaDE-MMTS	12	11	9	8	7	8	10	11
MOS	11	8	6	3	8	11	13	16
Tuned IPSOLS	13	12	11	7	6	7	8	12
VXQR1	15	16	15	10	4	3	4	9

Table 5 Mean errors of DEGPA, eDEGPA, and other algorithms provided in the test suite

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
DEGPA										
50	$0.00e+00$	$4.48e+00$	$4.59e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$5.94e+01$	$1.30e-04$	$0.00e+00$
100	$0.00e+00$	$1.60e+01$	$1.10e+02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$9.82e+02$	$3.85e-04$	$0.00e+00$
200	$0.00e+00$	$3.67e+01$	$2.17e+02$	$8.00e-04$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.10e+04$	$9.47e-05$	$0.00e+00$
500	$0.00e+00$	$7.97e+01$	$4.96e+02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$9.55e+04$	$2.32e-03$	$0.00e+00$
eDEGPA										
50	$0.00e+00$	$3.19e+00$	$5.56e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.86e+02$	$6.58e-05$	$0.00e+00$
100	$0.00e+00$	$2.90e+01$	$1.36e+02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$2.55e+03$	$1.07e-03$	$0.00e+00$
200	$0.00e+00$	$6.22e+01$	$2.56e+02$	$0.00e+00$	$7.21e-07$	$0.00e+00$	$0.00e+00$	$2.79e+04$	$6.28e-03$	$0.00e+00$
500	$0.00e+00$	$7.31e+01$	$6.22e+02$	$3.98e-02$	$0.00e+00$	$1.24e-01$	$3.64e-03$	$1.77e+05$	$2.10e-02$	$0.00e+00$
SOUPDE										
50	$0.00e+000$	$1.18e+000$	$3.10e+001$	$3.98e-002$	$0.00e+000$	$1.47e-014$	$2.28e-014$	$9.69e-002$	$3.75e-006$	$0.00e+000$
100	$0.00e+000$	$7.47e+000$	$7.92e+001$	$3.98e-002$	$0.00e+000$	$3.03e-014$	$3.88e-014$	$6.55e+001$	$7.82e-006$	$0.00e+000$
200	$0.00e+000$	$2.38e+001$	$1.80e+002$	$1.19e-001$	$0.00e+000$	$6.40e-014$	$7.46e-014$	$2.46e+003$	$1.51e-005$	$0.00e+000$
500	$0.00e+000$	$6.50e+001$	$4.71e+002$	$7.96e-002$	$0.00e+000$	$1.67e-013$	$1.78e-013$	$4.36e+004$	$3.59e-005$	$0.00e+000$
DE-$D^{40}+M^m$										
50	$3.33e-018$	$1.67e-001$	$1.34e+001$	$1.99e-001$	$0.00e+000$	$4.55e-014$	$0.00e+000$	$6.11e-001$	$0.00e+000$	$1.89e-031$
100	$2.78e-017$	$2.24e+000$	$7.61e+001$	$1.99e-001$	$1.39e-017$	$1.01e-013$	$5.33e-017$	$4.75e+005$	$4.29e-004$	$0.00e+000$
200	$6.66e-017$	$9.58e+000$	$1.69e+002$	$2.39e-001$	$2.78e-017$	$2.51e-013$	$0.00e+000$	$2.19e+008$	$0.00e+000$	$3.51e+001$
500	$2.23e-016$	$3.72e+001$	$4.54e+002$	$9.15e-001$	$1.03e-016$	$7.14e-013$	$0.00e+000$	$1.41e+010$	$6.78e-009$	$2.43e-031$
GaDE										
50	$0.00e+000$	$1.46e+001$	$1.18e+001$	$0.00e+000$	$0.00e+000$	$0.00e+000$	$0.00e+000$	$1.08e-008$	$6.24e-007$	$0.00e+000$
100	$0.00e+000$	$3.88e+001$	$5.89e+001$	$0.00e+000$	$0.00e+000$	$0.00e+000$	$0.00e+000$	$1.23e-003$	$3.87e-007$	$0.00e+000$
200	$0.00e+000$	$5.76e+001$	$1.61e+001$	$0.00e+000$	$0.00e+000$	$0.00e+000$	$0.00e+000$	$3.02e+000$	$4.53e-009$	$4.20e-002$
500	$0.00e+000$	$7.42e+001$	$4.40e+002$	$0.00e+000$	$0.00e+000$	$1.46e-014$	$0.00e+000$	$1.33e+003$	$0.00e+000$	$3.78e-001$

Table 5 continued

	F11	F12	F13	F14	F15	F16	F17	F18	F19
DEGPA									
50	6.98e − 05	1.08e − 08	2.97e + 01	3.15e − 08	0.00e + 00	1.03e − 06	2.32e + 00	9.50e − 07	0.00e + 00
100	6.60e − 04	2.31e − 07	7.23e + 01	3.58e − 08	0.00e + 00	3.05e − 06	2.27e + 01	2.56e − 06	0.00e + 00
200	7.82e − 04	1.49e − 06	1.40e + 02	1.55e − 08	0.00e + 00	1.71e − 06	6.24e + 01	1.06e − 05	0.00e + 00
500	2.16e − 03	4.24e − 06	3.71e + 02	1.60e − 08	0.00e + 00	1.71e − 05	1.41e + 02	7.67e − 06	0.00e + 00
eDEGPA									
50	7.25e − 05	0.00e + 00	5.02e + 01	3.34e − 06	0.00e + 00	0.00e + 00	7.44e + 00	4.67e − 06	0.00e + 00
100	1.01e − 03	1.43e − 02	9.07e + 01	1.78e − 06	0.00e + 00	1.48e − 08	3.43e + 01	8.40e − 06	0.00e + 00
200	5.31e − 03	3.72e − 03	1.95e + 02	1.38e − 07	0.00e + 00	1.33e − 03	6.29e + 01	1.67e − 04	4.20e − 02
500	2.78e − 03	1.66e − 04	5.77e + 02	3.30e − 03	5.50e − 01	1.46e + 01	2.67e + 02	1.19e − 01	2.16e + 00
SOUPDE									
50	3.09e − 006	0.00e + 000	2.06e + 001	0.00e + 000	1.38e − 014	0.00e + 000	2.53e − 001	0.00e + 000	0.00e + 000
100	6.75e − 006	0.00e + 000	5.85e + 001	9.09e − 015	2.79e − 014	0.00e + 000	8.55e + 000	0.00e + 000	0.00e + 000
200	1.43e − 005	0.00e + 000	1.35e + 002	3.98e − 002	5.79e − 014	0.00e + 000	3.31e + 001	0.00e + 000	1.91e − 014
500	4.66e − 004	0.00e + 000	3.58e + 002	1.31e − 012	1.39e − 013	0.00e + 000	1.09e + 002	2.82e − 013	4.95e − 014
DE-$D^{40} + M^{m}$									
50	6.06e − 004	1.58e − 021	1.39e + 001	1.19e − 001	0.00e + 000	1.76e − 016	4.31e − 002	3.98e − 002	0.00e + 000
100	0.00e + 000	4.62e − 017	5.33e + 001	1.19e − 001	0.00e + 000	8.99e − 015	3.22e + 000	8.11e − 010	0.00e + 000
200	0.00e + 000	5.45e − 015	1.23e + 002	3.98e − 002	0.00e + 000	2.26e − 013	2.72e + 001	3.98e − 002	9.47e − 032
500	0.00e + 000	5.05e − 013	3.48e + 002	2.39e − 001	0.00e + 000	4.17e − 012	1.02e + 002	9.97e − 008	0.00e + 000
GaDE									
50	1.31e − 006	0.00e + 000	1.19e + 001	9.78e − 013	0.00e + 000	4.78e − 012	4.97e − 001	4.82e − 008	0.00e + 000
100	4.34e − 007	0.00e + 000	4.99e + 001	7.90e − 013	0.00e + 000	2.45e − 012	3.28e + 000	1.96e − 008	0.00e + 000
200	1.85e − 007	4.92e − 014	1.24e + 002	2.87e − 012	0.00e + 000	1.58e − 012	2.45e + 001	2.53e − 008	0.00e + 000
500	0.00e + 000	1.07e − 012	3.34e + 002	2.79e − 011	0.00e + 000	1.67e − 012	9.26e + 001	5.59e − 008	4.20e − 002

Table 5 continued

	F1	F2	F3	F4	F5	F6	F7	F8	F9
jDElscop									
50	$0.00e+00$	$3.15e-02$	$2.28e+01$	$0.00e+00$	$0.00e+00$	$9.55e-14$	$0.00e+00$	$9.97e-03$	$0.00e+00$
100	$0.00e+00$	$1.21e+00$	$6.13e+01$	$0.00e+00$	$0.00e+00$	$2.00e-13$	$0.00e+00$	$5.57e+00$	$7.18e-09$
200	$0.00e+00$	$7.54e+00$	$1.40e+02$	$0.00e+00$	$0.00e+00$	$4.52e-13$	$0.00e+00$	$2.52e+02$	$4.30e-08$
500	$0.00e+00$	$3.06e+01$	$4.06e+02$	$1.59e-01$	$0.00e+00$	$1.18e-12$	$0.00e+00$	$5.66e+03$	$6.10e-08$
SaDE-MMTS									
50	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$4.13e-09$	$1.35e-01$
100	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$3.05e-04$	$3.18e-01$
200	$0.00e+00$	$1.34e+00$	$0.00e+00$	$8.08e-02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$2.67e+01$	$1.24e+00$
500	$0.00e+00$	$1.25e+01$	$0.00e+00$	$3.85e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$3.01e+02$	$2.81e+01$
MOS									
50	$0.00e+00$	$4.64e-13$	$9.61e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.54e-08$	$0.00e+00$
100	$0.00e+00$	$2.94e-12$	$2.03e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$9.17e-02$	$0.00e+00$
200	$0.00e+00$	$1.24e-11$	$4.01e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.16e+02$	$0.00e+00$
500	$0.00e+00$	$5.51e-04$	$4.57e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.28e+04$	$0.00e+00$
MA-SSW-Chains									
50	$1.67e-17$	$7.61e-02$	$4.79e+01$	$1.19e-01$	$0.00e+00$	$4.89e-14$	$9.33e-17$	$3.06e-01$	$2.94e+02$
100	$2.78e-17$	$7.01e+00$	$1.38e+02$	$1.19e-01$	$1.39e-17$	$6.03e-14$	$8.17e-16$	$3.48e+01$	$5.63e+02$
200	$5.33e-17$	$3.36e+01$	$2.50e+02$	$4.43e+00$	$2.72e-17$	$1.19e-13$	$6.96e-15$	$7.23e+02$	$1.17e+03$
500	$1.01e-16$	$7.86e+01$	$6.07e+02$	$1.78e+02$	$7.70e-17$	$2.63e-13$	$4.69e-14$	$1.32e+04$	$2.53e+03$

Table 5 continued

	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19
jDElscop										
50	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.36e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$7.43e-03$	$2.41e-14$	$0.00e+00$
100	$0.00e+00$	$8.17e-09$	$0.00e+00$	$5.11e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$3.21e-01$	$6.33e-14$	$0.00e+00$
200	$0.00e+00$	$9.58e-09$	$0.00e+00$	$1.10e+02$	$4.11e-16$	$0.00e+00$	$0.00e+00$	$2.39e+01$	$2.04e-13$	$0.00e+00$
500	$0.00e+00$	$4.40e-08$	$0.00e+00$	$3.14e+02$	$8.00e-02$	$0.00e+00$	$0.00e+00$	$7.65e+01$	$1.11e-12$	$0.00e+00$
SaDE-MMTS										
50	$0.00e+00$	$5.19e-05$	$0.00e+00$	$4.23e+00$	$3.93e-08$	$0.00e+00$	$0.00e+00$	$4.78e-01$	$9.38e-03$	$0.00e+00$
100	$0.00e+00$	$2.00e-04$	$0.00e+00$	$3.30e+01$	$1.02e-02$	$0.00e+00$	$0.00e+00$	$1.17e+01$	$4.70e-02$	$0.00e+00$
200	$0.00e+00$	$2.39e-04$	$0.00e+00$	$8.89e+01$	$1.57e-02$	$0.00e+00$	$0.00e+00$	$3.50e+01$	$3.35e-01$	$0.00e+00$
500	$0.00e+00$	$2.53e+01$	$0.00e+00$	$3.27e+02$	$4.01e-01$	$0.00e+00$	$0.00e+00$	$9.80e+01$	$1.18e+00$	$0.00e+00$
MOS										
50	$0.00e+00$	$0.00e+00$	$0.00e+00$	$4.55e-01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.40e+01$	$0.00e+00$	$0.00e+00$
100	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.75e+01$	$1.68e-11$	$0.00e+00$	$0.00e+00$	$1.43e+01$	$0.00e+00$	$0.00e+00$
200	$0.00e+00$	$0.00e+00$	$0.00e+00$	$9.03e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$5.03e+00$	$0.00e+00$	$0.00e+00$
500	$0.00e+00$	$0.00e+00$	$0.00e+00$	$3.78e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.21e+01$	$0.00e+00$	$0.00e+00$
MA-SSW-Chains										
50	$1.67e-30$	$4.49e-03$	$6.27e-41$	$3.02e+01$	$1.37e-17$	$3.91e-16$	$4.06e-03$	$2.60e+01$	$3.88e-19$	$4.02e-31$
100	$1.05e-29$	$1.09e-01$	$3.28e-03$	$8.35e+01$	$2.21e-16$	$1.59e-15$	$1.61e-02$	$9.92e+01$	$2.71e-18$	$3.15e-30$
200	$5.41e-29$	$3.50e-01$	$1.75e-02$	$1.68e+02$	$9.76e-01$	$5.32e-15$	$6.02e-02$	$7.55e+01$	$4.29e-04$	$1.51e-16$
500	$2.80e-01$	$4.21e+01$	$2.55e+01$	$4.00e+02$	$5.65e+01$	$5.53e+00$	$1.08e-01$	$1.38e+02$	$2.41e-03$	$7.84e-17$

Table 6 Mean errors of DEGPA, eDEGPA, and other algorithms provided in the test suite

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
DEGPA										
50	$0.00e+00$	$4.48e+00$	$4.59e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$5.94e+01$	$1.30e-04$	$0.00e+00$
100	$0.00e+00$	$1.60e+01$	$1.10e+02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$9.82e+02$	$3.85e-04$	$0.00e+00$
200	$0.00e+00$	$3.67e+01$	$2.17e+02$	$8.00e-04$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.10e+04$	$9.47e-05$	$0.00e+00$
500	$0.00e+00$	$7.97e+01$	$4.96e+02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$9.55e+04$	$2.32e-03$	$0.00e+00$
eDEGPA										
50	$0.00e+00$	$3.19e+00$	$5.56e+01$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$1.86e+02$	$6.58e-05$	$0.00e+00$
100	$0.00e+00$	$2.90e+01$	$1.36e+02$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$0.00e+00$	$2.55e+03$	$1.07e-03$	$0.00e+00$
200	$0.00e+00$	$6.22e+01$	$2.56e+02$	$0.00e+00$	$7.21e-07$	$0.00e+00$	$0.00e+00$	$2.79e+04$	$6.28e-03$	$0.00e+00$
500	$0.00e+00$	$7.31e+01$	$6.22e+02$	$3.98e-02$	$0.00e+00$	$1.24e-01$	$3.64e-03$	$1.77e+05$	$2.10e-02$	$0.00e+00$
RPSO-vm										
50	$2.62e-14$	$7.54e-03$	$1.75e+03$	$2.30e-14$	$9.53e-02$	$1.49e-12$	$1.14e-15$	$1.06e+03$	$2.94e-01$	$0.00e+00$
100	$2.66e-14$	$1.98e-01$	$1.42e+03$	$2.61e-14$	$1.07e-01$	$4.76e-12$	$0.00e+00$	$1.09e+04$	$1.85e-01$	$0.00e+00$
200	$2.53e-14$	$2.00e+00$	$1.03e+03$	$2.43e-14$	$1.73e-01$	$2.95e-12$	$0.00e+00$	$5.23e+04$	$1.67e+00$	$0.00e+00$
500	$2.65e-14$	$1.67e+01$	$1.13e+03$	$2.44e-14$	$2.54e-01$	$3.14e-12$	$8.90e-16$	$3.00e+05$	$4.85e+00$	$0.00e+00$
Tuned IPSOLS										
50	$0.00e+000$	$2.56e-014$	$0.00e+000$	$0.00e+000$	$6.72e-003$	$0.00e+000$	$4.98e-012$	$4.78e-009$	$4.95e-006$	$0.00e+000$
100	$0.00e+000$	$3.42e-014$	$0.00e+000$	$0.00e+000$	$1.31e-002$	$0.00e+000$	$3.00e-012$	$7.18e-003$	$1.09e+000$	$0.00e+000$
200	$0.00e+000$	$3.23e-014$	$0.00e+000$	$0.00e+000$	$0.00e+000$	$0.00e+000$	$1.14e-011$	$2.47e+001$	$4.09e+000$	$0.00e+000$
500	$0.00e+000$	$8.07e-014$	$0.00e+000$	$2.82e-003$	$0.00e+000$	$0.00e+000$	$1.06e-011$	$8.64e+003$	$1.13e+001$	$0.00e+000$

Table 6 continued

	F11	F12	F13	F14	F15	F16	F17	F18	F19
DEGPA									
50	6.98e − 05	1.08e − 08	2.97e + 01	3.15e − 08	0.00e + 00	1.03e − 06	2.32e + 00	9.50e − 07	0.00e + 00
100	6.60e − 04	2.31e − 07	7.23e + 01	3.58e − 08	0.00e + 00	3.05e − 06	2.27e + 01	2.56e − 06	0.00e + 00
200	7.82e − 04	1.49e − 06	1.40e + 02	1.55e − 08	0.00e + 00	1.71e − 06	6.24e + 01	1.06e − 05	0.00e + 00
500	2.16e − 03	4.24e − 06	3.71e + 02	1.60e − 08	0.00e + 00	1.71e − 05	1.41e + 02	7.67e − 06	0.00e + 00
eDEGPA									
50	7.25e − 05	0.00e + 00	5.02e + 01	3.34e − 06	0.00e + 00	0.00e + 00	7.44e + 00	4.67e − 06	0.00e + 00
100	1.01e − 03	1.43e − 02	9.07e + 01	1.78e − 06	0.00e + 00	1.48e − 08	3.43e + 01	8.40e − 06	0.00e + 00
200	5.31e − 03	3.72e − 03	1.95e + 02	1.38e − 07	0.00e + 00	1.33e − 03	6.29e + 01	1.67e − 04	4.20e − 02
500	2.78e − 03	1.66e − 04	5.77e + 02	3.30e − 03	5.50e − 01	1.46e + 01	2.67e + 02	1.19e − 01	2.16e + 00
RPSO-vm									
50	1.68e − 02	8.58e − 02	6.57e + 02	6.81e − 02	0.00e + 00	7.88e − 11	8.73e + 02	5.05e − 02	0.00e + 00
100	4.61e − 01	1.60e − 14	2.25e + 03	1.27e − 01	0.00e + 00	4.87e − 08	1.76e + 03	1.36e − 01	0.00e + 00
200	5.66e − 01	4.64e − 02	1.17e + 04	7.96e − 02	0.00e + 00	5.40e − 01	2.08e + 04	1.50e − 01	0.00e + 00
500	4.88e + 00	1.32e − 08	1.33e + 03	1.29e + 00	0.00e + 00	2.12e + 00	5.72e + 02	2.47e + 00	0.00e + 00
Tuned IPSOLS									
50	8.19e − 002	1.17e − 011	2.65e − 010	1.18e + 000	2.62e − 011	2.80e + 000	3.10e + 000	1.24e + 000	1.19e − 011
100	1.48e − 003	2.58e − 011	1.59e − 001	4.34e + 000	2.89e − 011	9.09e − 012	8.16e − 003	1.68e + 000	7.33e − 012
200	6.98e + 000	6.65e − 012	2.55e + 000	4.27e + 000	1.08e − 010	1.65e + 000	1.15e + 001	4.34e + 000	1.11e − 011
500	1.14e + 001	1.93e − 011	4.14e + 000	1.30e + 001	2.83e − 010	1.33e + 000	1.33e + 001	1.33e + 001	2.73e − 011

Table 6 continued

	F1	F2	F3	F4	F5	F6	F7	F8	F9
EvoPROpt									
50	$1.22e-02$	$3.71e-01$	$1.12e+02$	$4.96e-02$	$5.13e-02$	$6.85e-03$	$2.63e-02$	$2.08e+02$	$8.02e+00$
100	$4.34e-02$	$3.30e+00$	$3.98e+02$	$1.07e-01$	$3.92e-02$	$2.50e-04$	$9.17e-02$	$2.27e+03$	$2.91e+01$
200	$8.03e-02$	$8.03e+00$	$2.91e+02$	$3.52e-01$	$2.68e-02$	$6.22e-01$	$3.82e-02$	$1.34e+04$	$6.22e+01$
500	$0.00e+00$	$2.04e+01$	$5.97e+02$	$1.45e+00$	$3.03e-02$	$1.21e+00$	$8.06e-03$	$7.05e+04$	$1.75e+02$
EM323									
50	$4.80e-13$	$4.08e-09$	$6.12e+01$	$5.34e-13$	$3.05e-13$	$5.66e-13$	$0.00e+00$	$2.08e+02$	$0.00e+00$
100	$9.91e-13$	$3.42e-04$	$2.10e+02$	$1.15e-12$	$5.91e-13$	$1.14e-12$	$0.00e+00$	$6.31e+02$	$3.29e-08$
200	$2.15e-12$	$1.92e-01$	$4.47e+02$	$2.23e-12$	$3.95e-04$	$2.42e-12$	$0.00e+00$	$3.37e+04$	$1.31e-08$
500	$5.80e-12$	$2.04e+01$	$1.25e+03$	$7.08e-12$	$1.77e-03$	$6.50e-12$	$0.00e+00$	$3.91e+05$	$2.21e-06$
VXQR1									
50	$0.00e+00$	$2.36e+00$	$3.69e-09$	$0.00e+00$	$4.55e-14$	$3.07e-13$	$0.00e+00$	$0.00e+00$	$9.62e+00$
100	$0.00e+00$	$5.12e+01$	$3.19e-01$	$0.00e+00$	$6.03e-14$	$9.44e-13$	$0.00e+00$	$0.00e+00$	$2.15e+01$
200	$0.00e+00$	$8.50e+01$	$3.60e+01$	$2.50e-14$	$1.18e-03$	$3.10e-12$	$0.00e+00$	$0.00e+00$	$4.54e+01$
500	$0.00e+00$	$9.38e+01$	$2.14e+02$	$1.86e-13$	$2.96e-04$	$9.10e-12$	$0.00e+00$	$0.00e+00$	$8.68e+01$
GODE									
50	$0.00e+000$	$2.57e-001$	$3.06e+001$	$1.05e-013$	$0.00e+000$	$1.24e-014$	$0.00e+000$	$1.67e-001$	$7.77e-006$
100	$0.00e+000$	$3.65e+000$	$8.14e+001$	$8.32e-014$	$0.00e+000$	$2.60e-014$	$0.00e+000$	$7.53e+001$	$1.46e-005$
200	$0.00e+000$	$1.53e+001$	$1.80e+002$	$4.17e-013$	$0.00e+000$	$5.45e-014$	$0.00e+000$	$2.10e+003$	$3.23e-005$
500	$0.00e+000$	$5.81e+001$	$4.76e+002$	$1.62e-003$	$0.00e+000$	$1.43e-013$	$0.00e+000$	$3.93e+004$	$7.84e-005$

Table 6 continued

	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19
EvoPROpt										
50	$4.80e-02$	$9.68e+00$	$2.27e+00$	$4.22e+01$	$9.97e-01$	$6.38e-02$	$5.63e+00$	$6.77e+01$	$1.62e+00$	$5.03e-02$
100	$2.05e-01$	$2.60e+01$	$5.01e+00$	$1.40e+02$	$1.24e+00$	$6.56e-02$	$8.29e+00$	$1.97e+02$	$3.34e+00$	$1.43e-01$
200	$1.04e+00$	$5.93e+01$	$1.00e+01$	$1.71e+02$	$3.75e+00$	$3.80e-01$	$1.74e+01$	$1.56e+02$	$8.85e+00$	$2.15e+00$
500	$3.29e+01$	$1.77e+02$	$1.73e+01$	$5.75e+02$	$9.00e+00$	$2.25e+00$	$4.87e+01$	$3.94e+02$	$3.28e+01$	$5.00e+01$
EM323										
50	$0.00e+00$	$2.16e-07$	$9.88e-08$	$1.96e+01$	$1.32e-07$	$0.00e+00$	$2.52e-07$	$8.58e+01$	$3.12e-07$	$0.00e+00$
100	$0.00e+00$	$9.99e-09$	$1.51e-02$	$5.97e+01$	$3.74e-07$	$0.00e+00$	$4.57e-07$	$9.60e+01$	$5.02e-02$	$0.00e+00$
200	$0.00e+00$	$1.92e-06$	$4.13e-07$	$2.97e+02$	$2.10e-01$	$0.00e+00$	$9.40e-07$	$5.73e+01$	$2.63e-03$	$0.00e+00$
500	$0.00e+00$	$1.10e-02$	$3.41e-01$	$1.19e+03$	$1.51e-01$	$0.00e+00$	$4.71e-03$	$2.14e+02$	$2.28e-01$	$0.00e+00$
VXQR1										
50	$0.00e+00$	$1.09e+01$	$1.49e+00$	$8.39e+00$	$2.58e-01$	$3.72e-05$	$4.48e+00$	$1.73e+01$	$6.94e-01$	$0.00e+00$
100	$0.00e+00$	$1.89e+01$	$5.24e+00$	$1.78e+01$	$6.41e-01$	$7.14e-04$	$1.18e+01$	$1.14e+02$	$1.26e+00$	$1.49e-06$
200	$0.00e+00$	$4.37e+01$	$4.00e+01$	$9.55e+01$	$1.27e+00$	$1.09e-01$	$3.00e+01$	$8.79e+01$	$8.48e+00$	$3.56e-05$
500	$0.00e+00$	$8.52e+01$	$1.21e+02$	$2.21e+02$	$4.31e+00$	$2.24e-01$	$6.32e+01$	$1.65e+02$	$5.52e+01$	$1.14e-03$
GODE										
50	$0.00e+000$	$6.44e-006$	$1.33e-013$	$2.55e+001$	$6.24e-009$	$0.00e+000$	$1.57e-010$	$1.17e+000$	$2.97e-007$	$0.00e+000$
100	$0.00e+000$	$1.58e-005$	$7.57e-012$	$6.32e+001$	$4.13e-008$	$0.00e+000$	$3.75e-010$	$1.11e+001$	$1.11e-006$	$0.00e+000$
200	$0.00e+000$	$3.12e-005$	$1.20e-010$	$1.38e+002$	$8.17e-002$	$0.00e+000$	$9.54e-010$	$3.74e+001$	$1.91e-006$	$0.00e+000$
500	$0.00e+000$	$8.25e-005$	$7.39e-010$	$3.59e+002$	$7.67e-002$	$0.00e+000$	$2.24e-009$	$1.12e+002$	$5.06e-006$	$0.00e+000$

of DEGPA and eDEGPA appear in both Tables to facilitate comparisons.

The number of test problems where DEGPA and eDEGPA achieved non-inferior (equal or better) or inferior (worse) average errors than the other algorithms is depicted in Fig. 4 and reported in Table 4, offering two interesting observations. On the one hand, we can see that DEGPA and eDEGPA have similarly-shaped lines, which shows that they retain contiguous behavior with respect to the rest of the algorithms. Nevertheless, we can patently observe that DEGPA retains its performance with dimension (overlapping lines), while eDEGPA shows declining behavior, with plot lines of higher dimension being enclosed by the lines of lower dimensions. This is a verification of the previous observations, interpreted as the outcome of its additional requirements in computational budget. Nevertheless, it worths noting that the proposed approaches achieved highly competitive performance also to non-DE-based algorithms, such as PSO-based approaches, MA-SSW-Chains, EvoPROpt, EM323, and VXQR1, in all dimensions.

5 Conclusion

Choosing the proper control parameters of metaheuristics is a difficult, problem-dependent task. This work introduced a technique for grid-based parameter adaptation of the DE algorithm. DE was selected due to its known sensitivity on parameter values that render their dynamic adaptation a challenging task. The efficiency of the resulting DEGPA and eDEGPA approaches was evaluated on a variety of test problems with dimensions ranging from 50 up to 500.

The results suggested that the proposed approaches can relieve the user from the burden of parameter setting without loss in average performance over the whole test suite. Additional, performance was significantly improved comparatively to the provided 17 algorithms. Naturally, dimension has a declining effect especially for the fully adaptive eDEGPA approach. However, one shall take into consideration that both DEGPA and eDEGPA do not spend additional effort in preliminary experimentation for their tuning. Also, the proposed grid-based adaptation can be straightforwardly parallelized with significant gain in running time.

The proposed approach opens directions for further inquiry. The inclusion of additional parameters, such as mutation operator and population size, can be the next step forward. Also, since the grid-based adaptation is not explicitly dependent from the DE algorithm, it would be interesting to incorporate it also in different (population-based or not) algorithms.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

References

Auger A, Hansen N (2005) A restart CMA evolution strategy with increasing population size. In: Proceedings of the 2005 IEEE congress on evolutionary computation, pp 769–1776

Bäck T (1996) Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press, New York

Brest J, Bošković B, Zamuda A (2012) Self-adaptive differential evolution algorithm with a small and varying population size. In: WCCI 2012 IEEE World congress on computational intelligence

Brest J, Greiner S, Bošković B, Mernik M, Žumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans Evol Comput 10(6):646–657

Brest J, Maucec MS (2011) Self-adaptive differential evolution algorithm using population size reduction and three strategies. Soft Comput 15:2157–2174

Das S, Suganthan PN (2011) Differential evolution: a survey of the state-of-the-art. IEEE Trans Evol Comput 15(1):4–31

de Oca MAM, Aydin D, Stützle T (2011) An incremental particle swarm for large-scale optimization problems: an example of tuning-in-the-loop (re)design of optimization algorithms. Soft Comput 15:2233–2255

Duarte A, Martí R, Gortazar F (2011) Path relinking for large scale global optimization. Soft Comput 15:2257–2273

Eiben AE, Hinterding R, Michalewicz Z (1999) Parameter control in evolutionary algorithms. IEEE Trans Evol Comput 3(2):124–141

Eiben AE, Smit SK (2011) Evolutionary algorithm parameters and methods to tune them. In: Hamadi Y, Monfroy E, Saubion F (eds) Autonomous search, chap. 2. Springer, Berlin, pp 15–36

Eshelman LJ, Schaffer JD (1993) Real-coded genetic algorithms and interval-schemata. Found Genet Algorithms 2:187–202

García-Martínez C, Rodríguez FJ, Lozano M (2011) Role differentiation and malleable mating for differential evolution: an analysis on large scale optimisation. Soft Comput 15:2109–2126

García-Nieto J, Alba E (2011) Restart particle swarm optimization with velocity modulation: a scalability test. Soft Comput 15:2221–2232

Gardeux V, Chelouah R, Siarry P, Glover F (2011) EM323: a line search based algorithm for solving high-dimensional continuous non-linear optimization problems. Soft Comput 15:2275–2285

Hoos HH (2011) Automated algorithm configuration and parameter tuning. In: Hamadi Y, Monfroy E, Saubion F (eds) Autonomous search, chap. 3. Springer, Berlin, pp 37–72

LaTorre A, Muelas S, Peña J (2011) A MOS-based dynamic memetic differential evolution algorithm for continuous optimization a scalability test. Soft Comput 15:2187–2199

LaTorre A, Muelas S, Peña J (2012) Multiple offspring sampling in large scale global optimization. In: 2012 IEEE congress on evolutionary computation (CEC). IEEE, pp 1–8

Lozano M, Herrera F, Molina D (2010) Evolutionary algorithms and other metaheuristics for continuous optimization problems. http://sci2s.ugr.es/eamhco/

Lozano M, Herrera F, Molina D (2011) Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems. Soft Comput 15:2085–2087

Molina D, Lozano M, Sánchez AM, Herrera F (2011) Memetic algorithms based on local search chains for large scale continuous optimisation problems: MA-SSW-Chains. Soft Comput 15:2201–2220

Neumaier A, Fendl H, Schilly H, Leitner T (2011) VXQR: derivative-free unconstrained optimization based on QR factorizations. Soft Comput 15:2287–2298

Parsopoulos K, Vrahatis M (2010) Particle swarm optimization and intelligence: advances and applications. Information Science Publishing (IGI Global)

Piotrowski AP (2013) Adaptive memetic differential evolution with global and local neighborhood-based mutation operators. Inf Sci 241:164–194

Poláková R, Tvrdík J, Bujok P (2014) Controlled restart in differential evolution applied to CEC2014 benchmark functions. In: IEEE congress on evolutionary computation

Price K, Storn R (2009) Differential evolution (DE) for continuous function optimization (an algorithm by Kenneth Price and Rainer Storn). http://www1.icsi.berkeley.edu/~storn/code.html

Price KV, Storn RM, Lampinen JA (2005) Differential evolution: a practical approach to global optimization. Springer, Berlin

Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evol Comput 13(2):398–417

Qing A (2009) Differential evolution: fundamentals and applications in electrical engineering. Wiley-IEEE Press, New York

Segura C, Coello CAC, Segredo E, León C (2015) On the adaptation of the mutation scale factor in differential evolution. Optim Lett 9(1):189–198

Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359

Takahama T (1997) Sample source code of differential evolution (coded by T. Takahama). http://www.ints.info.hiroshima-cu.ac.jp/~takahama/download/DE.html

Tanabe R, Fukunaga A (2013) Success-history based parameter adaptation for differential evolution. In: IEEE congress on evolutionary computation

Tanabe R, Fukunaga A (2014) Improving the search performance of SHADE using linear population size reduction. In: IEEE congress on evolutionary computation

Tang K, Yao X, Suganthan PN, MacNish C, Chen YP, Chen CM, Yang Z (2007) Benchmark functions for the cec2008 special session and competition on large scale global optimization. Nature Inspired Computation and Applications Laboratory, USTC, China, pp 153–177

Tvrdík J (2006) Competitive differential evolution. In: 12th international coference on soft computing

Tvrdík J, Poláková R (2013) Competitive differential evolution applied to CEC 2013 problems. In: 2013 IEEE Congress on evolutionary computation (CEC). IEEE, pp 1651–1657

Wang H, Wu Z, Rahnamayan S (2011) Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems. Soft Comput 15:2127–2140

Weber M, Neri F, Tirronen V (2011) Shuffle or update parallel differential evolution for large scale optimization. Soft Comput 15:2089–2107

Weber M, Tirronen V, Neri F (2010) Scale factor inheritance mechanism in distributed differential evolution. Soft Comput 14:1187–1207

Yang Z, Tang K, Yao X (2011) Scalability of generalized adaptive differential evolution for large-scale continuous optimization. Soft Comput 15:2141–2155

Zaharie D (2007) A comparative analysis of crossover variants in differential evolution. In: Proceedings of IMCSIT, pp 171–181

Zaharie D (2009) Influence of crossover on the behavior of differential evolution algorithms. Appl Soft Comput 9(3):1126–1138

Zaharie D, Petcu D (2005) Parallel implementation of multi-population differential evolution. In: Concurrent information processing and computing, pp 223–232

Zhang J, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. IEEE Trans Evol Comput 13:945–958

Zhao S, Suganthan P, Das S (2011) Self-adaptive differential evolution with multi-trajectory search for large-scale optimization. Soft Comput 15(11):2175–2185