

Parameter control and hybridization techniques in differential evolution: a survey

Elena-Niculina Dragoi¹ · Vlad Dafinescu²

Published online: 18 December 2015
© Springer Science+Business Media Dordrecht 2015

Abstract Improving the performance of optimization algorithms is a trend with a continuous growth, powerful and stable algorithms being always in demand, especially nowadays when in the majority of cases, the computational power is not an issue. In this context, differential evolution (DE) is optimized by employing different approaches belonging to different research directions. The focus of the current review is on two main directions: (a) the replacement of manual control parameter setting with adaptive and self-adaptive methods; and (b) hybridization with other algorithms. The control parameters have a big influence on the algorithms performance, their correct setting being a crucial aspect when striving to obtain optimal solutions. Since their values are problem dependent, setting them is not an easy task. The trial and error method initially used is time and resource consuming, and in the same time, does not guarantee optimal results. Therefore, new approaches were proposed, the automatic control being one of the best solution developed by researchers. Concerning hybridization, the scope was to combine two or more algorithms in order to eliminate or to reduce the drawbacks of each individual algorithm. In this manner, different combinations at different levels were proposed. This work presents the main approaches mixing DE with global algorithms, DE with local algorithms and DE with global and local algorithms. In addition, a special attention was given to the situations in which DE is employed as a local search procedure or DE principles are included in other global search methods.

Keywords Self-adaptation · Hybridization · Differential evolution · Global optimization · Memetic algorithms

✉ Vlad Dafinescu
vdafinescu@gmail.com

Elena-Niculina Dragoi
elenan.dragoi@gmail.com

¹ Faculty of Chemical Engineering and Environmental Protection, “Gheorghe Asachi” Technical University of Iasi, 73 Prof. Dr. Doc. D. Mangeron St., 700050 Iasi, Romania

² Clinical Emergency Hospital “Prof. Dr. Nicolae Oblu”, 2 Ateneului St., 700309 Iasi, Romania

1 Introduction

In the context of maximizing efficiency and reducing the resources needed, optimization plays a key role, all the manufacturing and engineering processes being influenced in a positive manner. Optimization is a dynamic process that implies finding the best-suited solution to a problem and maintaining the given constraints (Das and Suganthan 2011; Fister et al. 2011). The characteristics of the problem being solved (such as: types and complexity of the relations between objectives, constraints and decision variables) influence the optimization difficulty. Therefore, the classification of the methodologies used are organized into different classes (based on the characteristics taken into consideration). For example, when the criterion employed is the type of solution, two classes are encountered: global and local (Nocedal and Wright 2006). If the criterion applied is the type of model, then optimization can be deterministic or stochastic (Nocedal and Wright 2006). In this work, a specific optimization algorithm belonging to the global and stochastic classes is studied. It is represented by differential evolution (DE), a population based stochastic metaheuristic (Zaharie 2009) developed by Storn and Price (1995) in order to solve the Chebychev Polynomial fitting problem.

All the developments performed in the area of DE have the role of improving it and making it more flexible for theoretical and real-life applications. Although it performs well on a wide variety of problems, DE has a series of problems related to: stagnation, premature convergence, sensitivity/insensitivity to control parameters (Das et al. 2007; Das and Suganthan 2011; Lu et al. 2010a; Mohamed et al. 2013).

Stagnation is the undesirable situation in which a population-based algorithm does not converge even to a suboptimal solution, while the population diversity is still high (Neri and Tirronen 2010). As the population does not improve over a period of generations, the algorithm is not able to determine a new search space for finding the optimal solution (Davendra and Onwubolu 2009). The persistence of a fit individual for a number of generations does not necessarily imply poor performance, but it may indicate a natural stage of the algorithm in which the other individuals are still updated (Neri and Tirronen 2008). Various factors can induce stagnation, the most influential being represented by the use of bad choices for the control parameters (CPs) and the dimensionality of the decision space (Neri and Tirronen 2010; Salman et al. 2007). For example, in the endeavor to obtain fast convergence, low values for population dimension (Np) are used, but this leads to smaller perturbation possibilities and therefore to limited power to find new regions for improvement (Storn 2008).

Premature convergence is the situation when the characteristics of some highly rated individuals dominate the population, determining a convergence to a local optimum where no more descendants better than the parents can be produced (Nicoara 2009). In DE, three types of convergence can be encountered: (a) good (the global optimum is reached in a reasonable amount of generations, obtaining a good trade-off between exploration and exploitation); (b) premature; and (c) slow (the optimum is not reached in a reasonable amount of generation, the perturbation overwhelming the selection process) (Zaharie 2002b). Related to convergence, two imperatives are considered: (a) identification of occurrence and (b) evaluation of its extent (Nicoara 2009). Concerning identification of different measures (that are in fact measures for the level of population degeneration), the difference between the best and average fitness, or Hamming distance between individuals and the variance of the Hamming distances, can be employed (Nicoara 2009). Other measures of convergence are the Q-measure (combines convergence with the probability to converge and it serves to compare the objective function convergence of different evolutionary algorithms) and the P-measure (analyses convergence from the population point of view) (Feoktistov 2006).

Sensitivity or insensitivity to *CPs* is another DE drawback. Empirical studies showed that the more sensitive a strategy is, the better the solution that can be achieved (Feoktistov 2006). This is because the effectiveness, efficiency, and robustness of the algorithm are dependent on the *CPs* values, their best settings being related to the function and requirements for time and accuracy (Brest 2009).

Along with the drawbacks mentioned earlier, another aspect that must be taken into consideration when using DE is the lack of mechanisms for handling constraint problems. This is often encountered in real-life optimization procedures, and therefore, various researchers focused on this aspect.

In order to overcome some of these problems, new variants (based on twisting and turning of the various DE constituents) were proposed (Peng and Wang 2010; Storn 2008). Although the No Free Lunch Theorem suggested that no panacea could exist, the scope was to make DE a fool proof and fast optimization method for any kind of objective function (Storn 2008). All these modifications followed three main directions (Brest et al. 2011): (a) replacing the hand tuning of control parameters with adaptive or self-adaptive mechanisms; (b) hybridizing DE by combining it with other optimization techniques; and (c) Introducing more mutation strategies during the optimization process.

In this work, the problems of hand-tuning replacement and hybridization are tackled in detail. The introduction of new mutation strategies is not discussed in this review because the authors considered that it would be better presented in the context of all the steps of DE algorithm, fact that will be the subject of a future work.

2 Parameter control

The role of *CPs* (F = mutation factor, Cr = crossover probability and Np = population dimension) is to keep the exploration/exploitation balance (Feoktistov 2006). Exploration is related to the discovery of new solutions and the exploitation is related to the search near new good solutions, both interweaving each other in the evolutionary search (Fister et al. 2011).

Each parameter influences specific aspects of the algorithm, the DE effectiveness, efficiency, and robustness being dependent on their correct values (Brest 2009). The determination of the *CPs* (as their optimal values are problem specific, varying for different functions or for functions with different requirements) is a difficult task, especially when a balance between reliability and efficiency is desired (Hu and Yan 2009b). In the early days, when DE was still in its infancy, empirical rules were laid down (Gamperle et al. 2002; Storn 1996; Storn and Price 1997). Unfortunately, these were sometimes contradictory and lead to confusion (Das and Suganthan 2011). In addition to these rules, the standard attempt to set up the *CPs* was represented by the trial-and-error approach that was not only time consuming, but it lacked efficiency and reliability (Tvrdik 2009). As time passed, researchers focused on the diversity of population estimation, taking into consideration the fact that the ability of an evolutionary algorithm (EA) to find optimal solutions is dependent on the exploration–exploitation relation (Feoktistov 2006).

Cr and F affect the convergence speed and robustness of the search space, their optimal values depending on the characteristics of the objective function and on Np (Ilonen et al. 2003). Cr controls the number of characteristics inherited from the mutant vector and thus, it can be interpreted as a mutation probability, providing the means to exploit decomposability (Price et al. 2005). Compared to F , Cr is more sensitive to the problems characteristics (complexity, multi-modality and so on) (Qin and Suganthan 2005).

On the other hand, F is more related to the convergence speed, influencing the size of perturbation and ensuring the population diversity (Price 2008). Larger values of F imply a larger exploration ability, but it was determined that smaller values than 1 are usually more reliable (Ronkkonen et al. 2005). Some authors go even further and prove that F should be larger in the first generations and smaller in the last ones, thus focusing on the local search as a mean to ensure convergence (Li and Liu 2010).

In the context of DE scaling factor randomization, two new terms (jitter and dither) are defined by Price et al. (2005). Jitter represents the procedure in which, for each parameter of the individual, a different F value is generated and subscribed with the corresponding index. Although jitter is not rotationally invariant, this approach seems to be effective for non-deceiving objective functions (which possess a strong global gradient information) (Price et al. 2005). Distinctively from the jitter case, dither represents the situation in which F is generated for each individual and assigned to its corresponding index. In this case, each characteristic of the same individual is evolved using the same scaling factor (Das and Suganthan 2011). Although dither is rotationally invariant, when the level of variation is very small, the rotation has small influence (Price et al. 2005). The application of these principles (dither and jitter) is encountered in multiple studies. For example, in Kaelo and Ali (2007), F is generated for each individual in the $[0.4, 1]$ range, while Cr is chosen from the interval $[0.5, 0.7]$ and is fixed per iteration.

Concerning Np , when its value is too small, stagnation appears (as there is no sufficient exploration), and when it is too big, the number of function evaluations rises, retarding the convergence (Feoktistov 2006). Different researchers recommend different ranges included in the interval $[2D-40D]$, where D represents the problem dimensionality. In the case of high D problems, using an Np value respecting this rule leads to a high computational time and therefore the recommended interval is not always used by researchers. Depending on the problem characteristics, different Np values are optimal. For example, separable and unimodal functions require low values, while parameter dependent and multimodal functions require high values (Mallipeddi et al. 2011). In addition, a correlation between population and F exists, a larger Np requiring a smaller F (Feoktistov 2006).

As it can be observed, due to different factors, setting the CPs is not a straightforward process. When taking into consideration the ‘how’ aspect of the methods used for parameter determination, two classes are distinguished: parameter tuning and parameter control (Eiben and Schut 2008). Parameter tuning consists in finding good values before running the algorithm. The drawbacks of this approach are related to: (a) the impossibility of trying all possible combinations; (b) the tuning process is time consuming; (c) even if the effort made for setting the parameters is significant, the selected values for a given problem are not necessarily optimal; (d) EAs are dynamic, adaptive processes and the use of rigid parameters is in contrast to this idea (Eiben et al. 1999).

In case of parameter control, the values are changed dynamically during the run (Brest et al. 2007), based on a set of defined rules. Based on the ‘how’ criterion of Eiben and Schut (2008), four sub-classes are encountered: (a) deterministic control; (b) adaptive control; (c) self-adaptive control and; (d) hybrid. On the other hand, in Takahama and Sakai (2012) the methods for CPs control are classified into: (a) observation based (the proper parameter values are inferred according to the observations); and (b) success based (the adjustments are performed so that the success cases are frequently used). In Chiang et al. (2013), a new taxonomy for classifying the algorithms according to the number of candidate parameter values (continuous or discrete), number of parameters used in a single generation (one, multiple, individual, variable) and source of considered information (random, population, parent, individual) is proposed. This approach is applied only for F and Cr parameters.

In case of deterministic control, the *CPs* are adapted using a deterministic law, without any feedback information from the system (Feoktistov 2006). For example, in Michalski (2001), the population is set to a higher value (50) in the first nine generations and then is reduced to half in order to minimize the computational cost. In Zaharie (2002a) F is randomized, the pool of potential trial vectors being enlarged without increasing the population size. Das et al. (2005) proposed two DE variants in which F is modified randomly [DE with random scale Factor (DERSF)] or linearly decreased with time [DE with time varying scale factor (DETVSF)].

2.1 Adaptive control

In case of adaptive control, the direction and/or magnitude of the parameter change is determined using feedback information (Brest et al. 2007). Feoktistov (2006) identifies two classes belonging to this group: refresh of population and parameter adaptation. When applying refresh of population, the mechanisms consist in either replacing the bad individuals or injecting new individuals in the population (Feoktistov 2006). In Zhang et al. (2011), two approaches for changing the population size (lifetime mechanism and extinction mechanism) and two for inserting new individuals (clone good individuals and create a new population) were employed for adapting the population size, according to the online progress of fitness improvement. When adapting the parameters, the methods applied obey the state of population.

Zaharie (2003) proposed a parameter adaptation based on the idea of controlling the parameter diversity through population variance evolution. The algorithm was called ADE and the feedback rule for adapting F was dependent on another parameter (Υ) which must be tuned. As the author points out, the initial problem of choosing suitable parameter values seems to be replaced with the problem of choosing Υ , but the replacement is simpler because there are no inter-related parameters.

Zhang and Sanderson (2009b) give a special attention to parameter adaptation, an entire study being dedicated to the discussion of this problem. In addition, a new adaptive DE version (JADE) based on a new mutation strategy (DE/current-to-pbest/1) was proposed. For each individual, F and Cr are generated based on two additional parameters (μF and μCr) which are adapted using the average value of the parameters that generate successful individuals. Another adaptive DE variant is ADE, proposed by Hu and Yan (2009a). The *CPs* were modified for each generation, using the current generation and fitness value. After that, the individual's F and Cr were selected based on the fitness values of current, worst, and best individuals.

In Pant et al. (2009), F is randomly modified using a Laplace distribution. Laplace distribution is similar to the normal distribution, the difference consisting in its expressions: absolute difference from the mean (Laplace) and squared difference from the mean (normal distribution). Therefore, Laplace distribution has a fatter tail, which implies that it is able to control more effectively the differential term, and thus prevent premature convergence. The empirical results showed that the modified DE with Laplace distribution (MDE) has an improved performance compared to the classical approach.

Thangaraj et al. (2009a) changed F and Cr used in each generation by applying simple rules. Although the authors called the method adaptive in the sense that the *CPs* are changed every generation, the rules used do not depend on a feedback information from the system (Eqs. 1, 2):

$$F_{g+1} = \begin{cases} F_l + rand_1 \cdot \sqrt{Grand_1^2 + Grand_2^2}, & \text{if } P_F < rand_2 \\ F_0, & \text{otherwise} \end{cases} \tag{1}$$

$$Cr_{g+1} = \begin{cases} Cr_l \cdot rand_3, & \text{if } P_{Cr} < rand_4 \\ Cr_0 & \text{otherwise} \end{cases} \tag{2}$$

where $Grand_1$ and $Grand_2$ are Gaussian distributed random numbers with standard deviation 1 and mean 0, $rand_i, i = \{1, 2, 3\}$ is a uniform random number, P_F and P_{Cr} are probabilities to adjust the F and Cr parameters (fixed and equal to 0.5), F_l and Cr_l are the lowest boundaries of the F and Cr , respectively, and $F_0 = Cr_0 = 0.5$ are constant values.

In order to balance the local and the global search, [Lu et al. \(2010a, b\)](#) proposed a rule of adapting Cr based on the current generation:

$$Cr = Cr_0 \cdot 2^{e^{(1 - \frac{G}{\sigma_{current} + 1})}} \tag{3}$$

where Cr_0 is a user chosen value.

[Bhowmik et al. \(2010\)](#) proposed an adaptive selection of F , the main idea consisting in generating a population of F parameters around a F_{mean} value, one for each individual. At the end of generation, F_{mean} is updated based on the individual F .

Taking into consideration the optimization state (computed based on the population distribution), [Yu and Zhang \(2012\)](#) changed the strategy of adjusting F and Cr in the following way: when the system is in exploration state F is increased and Cr is decreased; and when the system is in exploitation state F is decreased and Cr increased.

In [Islam et al. \(2012\)](#), a similar adaptive approach to the one used in JADE ([Zhang and Sanderson 2009b](#)) called MDE_pBX was proposed. F and Cr were generated using a Cauchy distribution with a location parameter, which is adapted, based on the power mean of all F/Cr generating successful individuals.

[Alguliev et al. \(2012\)](#) adapted F using an affinity index (Af_i), computed using the fitness information of the individual and of the system. A small Af_i indicates that the individual is far away from the global best solution and therefore a strong global exploration is required. The adaptation formula is the following:

$$F_i(g) = \frac{1}{1 + \tanh(2Af_i(g))} \tag{4}$$

where \tanh represents the hyperbolic tangent function.

Another approach used for adapting the CPs is represented by the Levy distribution. [He and Yang \(2012\)](#) proposed a DE version in which, for every mutation in each generation, F and Cr are adapted using one of the four pre-defined Levy distributions. In order to determine which distribution to employ, probability parameters (adaptively updated based on the historical performance) were introduced. The historical performance is retained by using fitness improvement memories that store the difference between the fitness of an individual and its offspring for a fixed number of generations, named learning period. The larger the fitness improvement, the larger the probability of applying the strategy for determining F corresponding to the current generation.

In [Asafuddoula et al. \(2014\)](#), a roulette wheel based Cr selection scheme was employed. Initially the Cr values were mapped to continuous segments. From these sets, a Cr value was considered using a selection value and then updated based on the success or failure of the individual generated.

In the case of multi-objective problems, two different directions related to adaptation and self-adaptation can be encountered: (a) adaptation of the strategies developed for single-

objective cases; and (b) development of new specific approaches. For example, in order to extend the application of ADE to multi-objective problems, Zaharie and Petcu (2004) designed an adaptive Pareto DE (APDE).

As it is pointed out by numerous studies, the adaptive approaches are more effective than the classical versions. The added complexity and computational costs translate into performance improvement, fact that encouraged researchers to continue their work and to test the effectiveness of these approaches on different synthetic and real-life problems.

2.2 Self-adaptive control

In the case of self-adaptive control, the parameters are encoded into the algorithm itself (Feoktistov 2006). The concept of co-evolution (which is an effective approach to decompose complex structures and to achieve better performance) can be used to select the *CPs*, the user being relieved from the trouble of performing this task (Hu and Yan 2009b; Thangaraj et al. 2009a). By reconfiguring itself, the evolutionary strategy is fit to any general class of problems, the generality of the algorithm being extended (Brest et al. 2007). In addition, the convergence rate can be improved (Zhang and Sanderson 2009a). On the other hand, due to the randomness involved, the proof of convergence in self-adaptive EAs is difficult to determine (Brest et al. 2006).

An alternative to modifying the *CPs* at each generation is to gradually self-adapt, based on the success rate. Qin and Suganthan (2005), in SaDE, applied this principle to evolve the *CPs*, the strategies for generating the trial vectors and their associated parameters. In Yang et al. (2008a) SaDE is improved, a new algorithm (SaNSDE) being proposed. Three self-adaptive mechanisms were employed: (a) the candidate mutation adaptation was adapted with the strategy found in SaDE; (b) *F* was adjusted separately; and (c) the *Cr* self-adaptation of SaDE was enhanced with weighting.

Brest et al. (2006) proposed a self-adaptive algorithm (jDE) in which, for each individual in the new generation, the *F* and *Cr* parameters were computed as:

$$F_{i,G+1} = \begin{cases} F_l + rand_1 \cdot F_u, & \text{if } rand_2 < \tau_1 \\ F_{i,G}, & \text{otherwise} \end{cases} \quad (5)$$

$$Cr_{i,G+1} = \begin{cases} rand_3, & \text{if } rand_4 < \tau_2 \\ Cr_{i,G} & \text{otherwise} \end{cases} \quad (6)$$

where F_l , F_u are the lower and upper limits of the *F* parameter, τ_1 and τ_2 are the probabilities to adjust *F* and *Cr*.

In addition to *F* and *Cr*, Teo (2006) included the population size into the self-adaptive procedure. The algorithm, called DESAP, had two versions, one using an absolute encoding methodology for the population size (DESAP-Abs) and one a relative encoding (DESAP-Rel). Another difference between the two versions consists in the manner in which N_p is initialized.

The same principle of self-adaption encountered in Brest et al. (2006) is also employed by Neru and Tirronen in their hybrid version called scale factor local search differential evolution (SFLSDE) (Neru and Tirronen 2009). In addition, the evolution of *F* is improved by including a local search based on Golden selection search or hill-climb.

The self-adapting control parameter modified DE (SAPMDE) algorithm contains a modified mutation and a self-adaptive procedure in which *F* and *Cr* are changed using the information fitness of some of the individuals participating in the mutation phase (Wu et al. 2007). In Nobakhti and Wang (2008), a randomized approach is applied to self-adapt *F* based

on two new parameters (adaptation update interval and diversity value) and a set of upper and lower limits.

Zhang et al. (2010) proposed a novel self-adaptive differential evolution algorithm (DMSDE) in which the population was divided into multi-groups individuals. The difference between the objective function of individuals from the current group influences F and Cr , the strategy being constructed based on Eqs. 7 and 8.

$$F_{gi}^t = F_l + (F_u - F_l) \cdot \frac{f_{g\ middle}^t - f_{g\ best}^t}{f_{g\ worst}^t - f_{g\ best}^t} \tag{7}$$

where F_{gi}^t is the scaling factor of the i th vector of g th group from the current generation t , F_l and F_u are the lower and upper limits of the F parameter, $f_{g\ best}^t$, $f_{g\ middle}^t$, $f_{g\ worst}^t$ are the best, middle and worst fitness functions of the three randomly selected vectors from the g group, in the generation t .

$$Cr_{gi}^t = \begin{cases} Cr_{gi}^t, & f_{gi}^t < \overline{f_g^t} \\ Cr_l + (Cr_u - Cr_l) \cdot \frac{f_{gi}^t - f_{g\ min}^t}{f_{g\ max}^t - f_{g\ min}^t}, & f_{gi}^t \geq \overline{f_g^t} \end{cases} \tag{8}$$

where Cr_{gi}^t is the crossover of the individual i from the g group in the t generation, Cr_u is the upper limit and Cr_l is the lower limit of the Cr parameter; $f_{g\ max}^t$, $f_{g\ min}^t$ are the maximum and minimum values of the fitness functions of all the individuals in the g group at t generation, f_{gi}^t is the fitness of the i individual from the g group, and $\overline{f_g^t}$ is the average value of the fitness of all individuals in the g group.

Recently, Pan et al. (2011) created a new DE algorithm (SspDE) with self-adaptive trial vector generation strategy and CPs . Three lists were used: strategy list (SL), mutation scaling factor list (FL), and crossover list (CRL). Trial individuals were created during each generation by applying the standard mutation and crossover steps, which use the parameters in the target-associated lists. If the trial was better than the target, the parameters were then inserted in the winning strategy list (wSL), winning F list (wFL), and winning Cr list (wCRL). After a predefined number of iterations, SL, FL and CRL were refilled with a great probability from the wining lists or with randomly generated values. In this manner, the self-adaptation of the parameters followed the different phases of the evolution.

In the improved self-adaptive differential evolution with multiple strategies (ISDEMS) algorithm, Deng et al. (2013) mentions that it employs for F (Eq. 9) the same adapting rule as in SACPMDE (Eq. 10) (Wu et al. 2007). However, when comparing the relations, it is clear that there is a big difference between the two.

$$F(g) = \frac{F_{min}}{1 + \left(\frac{F_{min}}{F_{max}} - 1\right) e^{-\alpha g}} \tag{9}$$

where α is the initial decay rate, g is the current generation and F_{min} , F_{max} are the minimum and the maximum values of F .

$$F_i = F_l + (F_u - F_l) \frac{f_{im} - f_{ib}}{f_{iw} - f_{ib}} \tag{10}$$

where f_{ib} , f_{im} and f_{iw} are the fitness functions of the base vector applied for generating the mutation vector corresponding to the i th individual, of the best and worst individuals in the current generation.

In another work (Wang and Gao 2014), the adaptation of F and Cr based on the principles used in jDE (Brest et al. 2006) is extended using a dynamic population size. The main

characteristics of the algorithm are: (a) it follows the inspiration of the original DE selection operator; (b) it requires few additional operations; and (c) it can be efficiently implemented. In addition, the new algorithm (called jDEdynNP-F), uses a changing sign mechanism for F .

In [Huang et al. \(2007\)](#) the SaDE algorithm was extended to solve numerical optimization problems with multiple conflicting objectives. The difference between SaDE and the new algorithm (called MOSaDE) consists in the evaluation criteria of promising or inferior individuals. MOSaDE is further improved, resulting in a multiobjective self-adaptive differential evolution with objective-wise learning strategies (OW-MOSaDE) where Cr and mutation strategies specific for each objective are separately evolved ([Huang et al. 2009](#)).

[Jingqiao and Sanderson \(2008\)](#) proposed a self-adaptive multi-objective DE called JADE2, in which an archive was used to store the recently explored inferior solutions. The difference between the individuals from the archive and the current population is utilized as a directional information about the optimum. A similar idea was employed by [Wang et al. \(2010c\)](#), an external elitist archive being used to retain the non-dominated solution. In addition, a crowding entropy diversity measure is used to preserve the Pareto optimality. If the CPs do not produce better trial vectors over a pre-specified number of generations, then they are replaced by adding to the lower limit a randomly scaled difference between the upper and the lower limit. The results showed that the algorithm, called MOSADE, was able to find better spread solutions with better convergence.

In recent years, some researchers claimed that no significant advantages are obtained when using self-adaptation to guide the CPs , but it was shown that there is a relationship between the schemes effectiveness and the balance between exploration and exploitation ([Segura et al. 2015](#)). In the majority of cases, the self-adaptive procedures tend to be more efficient than adaptive or deterministic approaches, a high number of works encountered in literature employing self-adaptation as an improvement technique.

2.3 Hybrid control

In this case, the parameters are modified using combined techniques from deterministic, adaptive, and self-adaptive control groups or other algorithms and principles. For example, [Mezura-Montes and Palomeque-Ortiz \(2009\)](#) proposed a modified version of DE in which a mechanism of deterministic and self-adaptive parameter control is used. In [Hu and Yan \(2009b\)](#) the Immune System algorithm was employed to perform the search in the control parameter space, while DE searched the solution space.

One approach for evolving the CPs is represented by fuzzy logic. One of the first works in which this direction was applied is represented by [Liu and Lampinen \(2005\)](#) where F and Cr are adapted using fuzzy logic. The parameters of the new DE variant (called FADE) responded to the population information. The algorithm convergence was much better than the classical variant, especially when high dimensionality problems were solved. In [Xue et al. \(2005\)](#), fuzzy logic was applied to dynamically adapt the perturbation factor of the reproduction operator and the greediness (a specific parameter of the multi-objective DE version employed). Two state variables (population diversity and generation percentage) were considered as inputs for the fuzzy logic controller. [Zade et al. \(2011\)](#) applied the fuzzy control logic to adapt F of a DE version used for a series of economic load dispatch problems.

Another mechanism for modifying the CPs is to employ chaotic systems. For example, [dos Santos Coelho et al. \(2009\)](#) and [dos Santos Coelho and Mariani \(2006\)](#) adapted the F parameter using three chaotic sequences based on a logistic equation. An approach using Lorzi's map was employed in [dos Santos Coelho \(2009\)](#), the algorithm (called LDE) having

a substantial potential for application in constrained optimization problems. Another work in which, among other DE alterations (mutation operation considering equality constraints and selection operation based on handling inequality constraints), the chaotic theory was applied to adapt the *CPs*, is represented by Yuan et al. (2008). In dos Santos Coelho et al. (2014), *F* and *Cr* are adapted using a Gaussian probability distribution and chaotic sequence based on a logistic map, the new algorithm, called DECG, having superior features. Another chaotic map used for setting the *CPs* is represented by the Ikeda map (dos Santos et al. 2012).

Lu et al. (2011) introduced the chaotic principle not only at parameter level, but also at general level, a chaotic local search procedure being applied. In their work, three different approaches combining DE with chaos principles were proposed. In the first version, called “CDE method 1”, *F* and *Cr* are adapted based on the Tent equations. In the second version (CDE method 2), a chaotic search procedure based on the same Tent equations is applied to locally determine the global optimal solution. The third version (CDE method 3) combines the two previous versions, resulting in a hybrid chaotic DE approach.

Although more complex and more difficult to apply than the other approaches it is based on, the hybrid set-up of control parameters can lead to good results. In this case, the predominantly used methods are represented by fuzzy logic and chaotic system, their efficiency when mixed with DE proving that this research direction can generate useful approaches.

3 Hybridization

Hybridization is the process of combining the best features of two or more algorithms in order to create a new algorithm that is expected to outperform the parents (Das and Suganthan 2011). It is believed that hybrids benefit from synergy, choosing adequate combinations of algorithms being one of the keys for top performance (Blum et al. 2011).

In the field of combinatorial optimization, the algorithms undergoing this process are also encountered under the name of hybrid metaheuristics (Xin et al. 2012). In literature, various optimization methods can be encountered, their performance depending on different aspects belonging to the problem domain characteristics (time variance, parameter dependence, dimensionality, objectives, constraints, etc.) and to the algorithm characteristics (convergence speed, type of search, etc.). In this context, each methodology has its strong points and weaknesses and, by combining different features from different algorithms, a new and improved methodology (that avoids if not all the problems but a majority of them) is created. By incorporating problem specific knowledge into an EAs, the No Free Lunch Theorem can be circumvented (Fister et al. 2011).

Depending on the type of algorithm the DE can be hybridized with, three situations are encountered: (a) DE and other global optimization algorithms; (b) DE with local search (LS) methods; and (c) DE with global optimization and LS methods. However, this is just one aspect in which hybridization can be organized. In Raidl (2006) an overview of hybrid metaheuristics is performed and a new categorization tree emerged. If the aspect of what is hybridized is considered, then three classes are encountered: (a) metaheuristics with metaheuristics; (b) metaheuristics with problem-specific algorithms; (c) metaheuristics with other operations research or artificial intelligence (which, in their turn can be: exact techniques, or other heuristics or soft computing methods). Concerning the level of hybridization, two cases are encountered: (a) high-level, weak coupling (the algorithms retain their own identities) and (b) low-level, strong coupling (individual components are exchanged). When order of execution is taken into account, the hybridization can be: (a) batch (sequential); (b) interleaved; and (c) parallel (from architecture, granularity, hardware, memory, task, and data allocation

or synchronization points of view). Concerning control strategy, two cases of hybridization are encountered: integrative and collaborative.

In an effort to characterize the hybridization based on the level of interaction at which the hybridization can be settled, Feoktistov (2006) distinguished four distinct situations: (a) *Individual level or search exploration level* For example, in Zhang and Xie (2003) the individuals of the PSO algorithm are mutated using alternatively the DE and PSO operator, resulting in a methodology called DEPSO. In Bandurski and Kwedlo (2010), the conjugate gradients (CG) algorithm was used to improve the individuals using two different strategies. In the first one, the candidates were fine-tuned, while in the second approach, the main population was improved after the selection step; (b) *Population level* This level represents the dynamic of a population or subpopulation. In order to eliminate the bad individuals from the population, Feoktistov and Janaqi (2006) proposed adding an energetic filter through which only individuals with lower fitness can pass; (c) *External level* that provides interaction with other methods. Keeping DE unchanged, Feoktistov and Janaqi (2004) add the least-square support vector machine approximation at the end of each cycle in order to improve the convergence algorithm. The method proposed in Ali and Torn (2002) and described below in Sect. 3.2 is also included in this class; (d) *Meta level* At this level, a superior meta-heuristic includes the algorithm as one of its strategy (Feoktistov 2006). DE algorithm was integrated into a set of competing heuristics where each heuristic was used based on a certain probability, depending on the success of heuristic in the previous step (Islam and Yao 2008).

3.1 DE with other global optimization techniques (global–global)

In this class, different global approaches are combined with DE in order to create a “super-algorithm” with extensive search capabilities. Depending on the situation, the combination can be performed in parallel or sequential, it can share the same population or work with separate populations.

3.1.1 DE and swarm intelligence

For hybridizing DE algorithm, the most commonly used global optimization technique is particle swarm optimization (PSO). PSO is inspired from swarm theory (bird flocking, fish schooling) and it is related to EAs, the adjustment to the best solution being conceptually similar to the crossover operation. Hendtlass (2001) realized the first combination of DE and PSO. In his algorithm (SDEA), the individuals obey the swarm principles. At random generations, DE is applied to move the individuals from the poorer area to a better one. Zhang and Xie (2003) use the same principle of updating the PSO individuals with DE in their DEPSO methodology.

Liu et al. (2010) proposed an integration of PSO with DE in a two-population method, applied for solving constraint numerical optimization problems. At each generation, three mutation strategies of DE (DE/rand/1, DE/current_to_best/1, and DE/rand/2) were used to update the previous best particles. Dulikravich et al. (2005) created a hybrid multi-objective, multi variable optimizer by combining the non-dominated sorting differential evolution (NSDE) with strength pareto evolutionary algorithm (SPEA) and multi-objective particle swarm (MOPSO). The methodology uses these algorithms alternatively based on a switch criterion, considering five different aspects of successive Pareto approximations and population generation.

In a review related to the combination of DE with PSO, Xin et al. (2012) introduced a new term to describe the combination of two global optimizers: collaboration. In this context, the

authors pointed out that it is difficult to separate the influence of each algorithm on the fitness function. In addition, an impressive list of DE–PSO combinations was presented. Another example of collaboration between PSO and DE is presented in [Xu et al. \(2012\)](#) where a new PSO-DE-least squares support vector regression combination is proposed for modelling the ammonia conversion rate in ammonia synthesis production. In [Epitropakis et al. \(2012\)](#) after each PSO step, the social and cognitive experience of the swarm is evolved using DE. The proposed framework is very flexible, different variants of PSO [bare bones PSO (BBPSO), dynamic multi swarm PSO (DMPSO), fully informed PSO (FIPS), unified PSO (UPSO) and comprehensive learning PSO (CLPSO)] incorporating different DE mutation strategies (DE/rand/1, DE/rand/2 and trigonometric) or variants (jDE [Brest et al. 2006](#), JADE [Zhang and Sanderson 2009b](#), SaDE [Qin and Suganthan 2005](#) and DEGL [Das et al. 2009](#)).

The latest DE-PSO combination proposed are: HPSO-DE (in which DE or PSO is randomly chosen to be applied to a common population) ([Yu et al. 2014](#)) and PSODE (where a serial approach is used together with a set of improvements, including hybrid inertia weight strategy, time-varying acceleration coefficients and random scaling factor strategy) ([Pandiarajan and Babulal 2014](#)).

[Ji-Pyng et al. \(2004\)](#) used the concept of ant colony optimizer (ACO) to search for the proper mutation operator, in order to accelerate the search for the global solution. [Vaisakh and Srinivas \(2011\)](#) proposed an evolving ant direction differential evolution (EADDE) algorithm in which the ant colony search systems are used to find the proper mutation operator according to heuristic information and pheromone information. In order to properly set the parameters of ant search, a GA version that includes reproduction by Roulette-wheel selection and single point crossover is applied. Other algorithms in which DE is combined with ACO include: DEACO ([Xiangyin et al. 2008](#); [Yulin et al. 2010](#)), ACDE ([Ali et al. 2009](#)), MACO ([dos Santos Coelho and de Andrade Bernert 2010](#)).

In the work of [Chang et al. \(2012\)](#) a serial combination is proposed, a dynamic DE version being hybridized with a continuous ACO and then applied for wideband antenna design. In another study, ACO is improved with DE and with the cloning principle of Artificial Immune System. In the new algorithm (DEIANT), DE was applied to a duplication of the original pheromone matrix. DEIANT was then used to solve the economic load dispatch problem ([Rahmat and Musirin 2013](#)) and the weighted economic load dispatch problem ([Rahmat et al. 2014](#)).

In combination with group search optimization (GSO), DE was applied to find the optimal operating conditions of a cracking furnace when variable feedstock properties are available ([Nian et al. 2013](#)). In the algorithm (called DEGSO), DE is first applied to find the local solution space and, when the change in fitness reaches a predefined value, the DE is stopped and GSO is started.

3.1.2 DE and evolutionary algorithms

[Yang et al. \(2008b\)](#) generalized the common features of DE and evolutionary programming (EP) into a unified framework and proposed NSDE (DE with neighborhood search) by introducing Gaussian and Cauchy Neighborhood Search operators. [Thangaraj et al. \(2009b\)](#) proposed two modifications of DE algorithms. First, DE was hybridized with EP and second, different initialization techniques for generating random number (such as uniformly distributed random numbers, Gaussian distributed random number, and quasi-random Sobol sequence) were applied. The EP based mutation was used only when the DE mutation does not generate a trial vector better than the current individual.

A new algorithm called DE/BBO combining the exploration capabilities of DE with the exploitation of biogeography-based optimization (BBO) is proposed in [Gong et al. \(2010\)](#). In addition, a new hybrid migration operator based on two considerations was developed. First, the good solutions are less often destroyed and the poor solution can accept a higher number of features from the good solutions. Secondly, the DE mutation is capable of efficiently exploring new search spaces and therefore, makes the hybrid more powerful.

An approach combining DE and GA was proposed in [da Silva and Barbarosa \(2010\)](#). In every generation, one of the two algorithms is chosen based on productivity (an algorithm being considered productive if it produces a new best). The probability of being selected is redefined at every run using a reward index. In case of constraint problems, the reward is obtained similar to the method of constraint handling. If both individuals are feasible, the reward is equal to the fitness difference. In case just one individual is not feasible, then the reward is represented by the sum of that individual's constraints.

Another study in which DE and GA are mixed is represented by [Meena et al. \(2012\)](#). Features from a discrete DE (DE) and GA (both with fixed *CPs*) were alternated based on the current generation parity and then used to solve the text documents clustering problem.

3.1.3 Other algorithms

A combination involving coevolution of DE and harmony search (HS) is applied in the work of [Wang and Li \(2012\)](#) where two populations evolve simultaneously. The algorithm, called CDEHS, is applied to different engineering problems and, in order for the HS to handle integer optimization, the search operator is modified to find only integers. In addition, the pitch adjustment operation is modified to directly generate integer variables.

In most cases, the authors start from simple DE variants and combine them until a complex, hybrid approach with improved performance is obtained. In [Guo et al. \(2013\)](#) another idea is employed, in the SCE-UA (shuffled complex evolution), a replacement of the simple search method with DE being realized. SCE-UA is a simple algorithm consisting of two operations: multiple complex shuffling and competitive evolution. The new algorithm, called MOSCDE has four main extensions: (a) strategy for sorting individuals of the population; (b) achievement of sets and updating strategy (specific to a multi-objective problem); (c) replacement of the simplex search with DE; and (d) extension of DE into a multi-objective framework.

3.2 DE with local search methods (global–local)

The evolutionary algorithms that apply LS processes in order to improve performance and to refine individuals are also called memetic algorithms (MA) ([Liao 2010](#)). This type of hybridization is an integrative (coercive) approach as one algorithm is considered a subordinate, being embedded in another algorithm ([Raidl 2006](#)).

Various studies showed that for some problems, MAs are more efficient and more effective than the traditional EAs ([Krasnogor and Smith 2005](#)). Since DE can suffer from stagnation problems, the role of the LS strategy is to compensate for this deficiency by refining the individuals ([Jia et al. 2011](#)).

[Rogalsky and Derksen \(2000\)](#) combined downhill simplex (DS) with DE in order to accelerate convergence, without getting trapped into local minima. The DE based mechanisms rely on mutation and crossover, DS relies on reflection, and the hybrid version denoted HDE uses all three mechanisms. From the trial vectors generated by DE, $n + 1$ are chosen to form a simplex, which is modified through reflection, until one or more individuals are improved.

In combination with rough sets, a modified DE was applied for multi-objective problems (Hernandez-Diaz et al. 2006). The optimization process is split in two phases, in Phase I, DE being used for 2000 fitness function evaluations. In the second phase, for 1000 fitness evaluations, the population of non-dominated and dominated sets generated in the first step are used to perform a series of rough sets iterations. A similar approach is also used in Santana-Quintero et al. (2010) where a new algorithm called DE for multi-objective optimization with local search based on rough set theory (DEMORS) is proposed.

Inspired from the evolutionary programming neighborhood search strategy, Yang et al. (2008b) introduced the concept of neighborhood into the DE algorithm. Experimental results indicated that the evolutionary behavior of the DE algorithm is affected. For the 48 widely used benchmarks, the performance of the improved version had significant advantages over classical DE. After that, further improvements were added to the strategy by utilizing a self-adaptive mechanism (Zhenyu et al. 2008). By incorporating the topological information into DE and by using pre-calculated differentials, Ali and Torn (2002) created a very fast and robust methodology.

Noman and Iba (2008) incorporated an adaptive hill climbing (HC) local search heuristic (AHCXLS) resulting in a new DEahcSPX algorithm. In order to solve the middle-size travelling salesman problem, a DE version with a position-ordering encoding (PODE) was improved by including HC (Wang and Xu 2011). In the HC operator, the neighborhood of the current solution is determined (by using swap, reverse edge, and insert operators) and the best solution is preserved. Another work in which the best DE individuals are improved by HC is (Hernandez et al. 2013). The HC implementation is a non-classical version, which operates on more than one dimension at a time.

For solving single objective and multi-objective permutation flow shop scheduling problems, a series of alterations to DE (including a largest-order value rule for converting continuous values to job permutations and a local search procedure designed according to the problems landscape) were performed (Qian et al. 2008). Specific to the problems characteristics, an insert-based LS (in which the parameters are randomly chosen, cycling is avoided and the new solution is accepted only if it is better than the existing one) was chosen. Another specific element is represented by the fact that LS is not applied to the individual, but to the job permutation it represents. A similar LS variant is used in Wang et al. (2010b). Another approach employed for solving the job-shop scheduling problem consisted in combining DE with a tree search algorithm (Zhang and Wu 2011). A set of ideas from the filter and fan algorithm were borrowed for a tree based local search that was carried out (immediately after the selection phase) for the $e\%$ individuals from the population. In order to deal with the parameters of the zero-wait scheduling of multiproduct batch plant with setup time (which is formulated as an asymmetrical traveling salesman problem), a permutation based DE, in combination with a fast complex heuristic local search scheme, is proposed in Dong and Wang (2012).

In order to improve the performance of the algorithm when applied to the problem of worst-case analysis of nonlinear control laws for hypersonic re-entry vehicles, a gradient-based local optimization procedure is introduced into DE (Menon et al. 2008). Distinctively from the majority of works in which the LS procedure is applied to the best individual, in this work, when no improvement is obtained, LS is applied to a random individual, the aim being to obtain local improvements in the search space. Trigonometric local search (TLS) and interpolated local search (ILS) are other two local procedures that were combined with DE in order to increase its efficiency (Ali et al. 2010). TSL is based on the trigonometric mutation operator (Fan and Lampinen 2003), while ISL is based on Quadratic Interpolation, being one of the oldest gradient-based methods used for optimization. In both cases, the

combination with DE (called DETLS and DEILS respectively) implies the selection of the best solution of two other random points for biasing the search in the neighborhood of the best fitness individual. The LS procedure is applied until there is no improvement. In another work, in two combinations [DE+LS(1), DE+LS(2)], ILS is applied to improve the DE performance (Tardivo et al. 2012). The difference between the two variants consists in the selection scheme applied to the LS procedure, in case of DE+LS(1) two individuals being randomly selected and the third representing the best individual found. On the other hand, in case of DE+LS(2), just one individual is randomly selected, the other two being represented by the two best solutions. In Asafuddoula et al. (2014), another gradient based search approach [sequential quadratic programming (SQP)] is applied to locally improve the best solution of DE. A 10% of the total function evaluations are allocated to the search procedure. When the LS fails to determine a better solution consuming a set of predefined number of functions of evaluations, then it is initialized for the next best solution.

In the memetic differential evolution (MDE) (Neri and Tirronen 2008), Hooke–Jeeves algorithm (HJA) and stochastic local search (SLS) are used to locally improve the initial solution by exploring its neighborhood. Liao (2010) applied to a modified DE version a random walk with direction exploitation (RWDE) in order to improve randomly selected trial vectors. In this manner, the creation mechanism has more chances of generating better individuals.

Wang et al. (2011) fused the search performed by DE with Nedler–Mead (NM) in a NMDE algorithm applied to parameter identification of several chaotic systems. The current population is improved by NM and, after that, it is taken over by the DE algorithm which creates the next generation.

In the integrated strategies differential evolution with local search (ISDE-L) algorithm, along with a series of improvements related to the mutation strategies, a local search procedure is applied to improve the performance (Elsayed et al. 2011). At each K generation, the 25% best individuals are selected and, for each individual, a random variable is selected and then modified by adding or subtracting a random Gaussian number.

Another approach often used as a local search method is represented by chaos theory. In combination with DE, chaos was not only applied for improving good solutions but also for adaptation of CPs. In dos Santos Coelho and Mariani (2008), the best solution generated with DE is considered a starting point for the chaotic local search (CLS) approach. A similar methodology is used in Lu et al. (2010b), the CLS procedure designed to solve the short-term hydrothermal generation scheduling being based on the logistic equation. Since chaotic search suffers from performance deterioration when exploring large search spaces, Jia et al. (2011) introduced a shrinking strategy for the search space and, after that, applied the new CLS to DE, the new algorithm (DECLS) being a promising tool for solving high dimensional optimization problems. Deng et al. (2013) proposed a DE variant with multiple populations (for solving high dimensional problems) in which a LS strategy in combination with the Chaos search approach (one dimensional Logistic mapping), is applied to improve the best solution obtained so far.

Another approach used to improve the performance of the algorithm is to apply the previous steps for generating better individuals. For example, Thangraj et al. (2010) used a Cauchy mutation operator as LS procedure. At the end of each DE generation, the best solution is mutated using the “best/1” strategy until no more improvement is obtained. The same Cauchy mutation operator is used in Ali and Pant 2011 to force the best individual to jump to a new position when the new concept of failure counter (FC) reaches a predefined value. The role of FC is scanning the individuals from each generation and keeping an account on the times an individual failed to show improvement.

In order to improve the MOEA/D-DE algorithm proposed in [Li and Zhang \(2009\)](#) for solving multi-objective problems with complex Pareto sets, [Tan et al. \(2012\)](#) introduced: (a) a uniform design method for generating aggregation coefficient vectors and (b) a three points simplified quadratic approximation. The role of the design method is to uniformly distribute the scalar optimization sub-problems and to allow a uniform exploration of the region of interest. The quadratic approximation is applied to improve the local search ability of the aggregation function. The experiments showed that the new algorithm (called UMODE/D) outperforms MOEA/D-DE and NSGA-II, even when the number of generation is reduced to half, compared with the other two algorithms.

Another approach used as a LS procedure for the DE algorithms is represented by the Gauss–Newton method ([Zhao et al. 2013](#)). Similar to the other approaches presented in this review, DE carries out the global search and the Gauss–Newton further explores the promising regions. In addition, a collocation approach is used instead of the Renge–Kuta method usually applied to solve the initial value problem for Gauss–Newton.

When used in combination with artificial neural networks (ANNs) specific training approaches can be applied as LS procedures in order to improve specific solutions. This mix of algorithms is possible because each DE individual is a numerical representation of an ANN. One of the most used ANN training procedures employed in DE is represented by back propagation (BP) algorithm, which is a gradient descent method. Examples of algorithms containing DE-ANN-BK are: MPDENN (where a resilient BP with backtracking-iRprop+ variant is used as a LS) ([Cruz-Ramirez et al. 2010](#)), SADE-NN-2 ([Dragoi et al. 2012](#)), DE-BP ([Sarangi et al. 2013](#)), hSADE-NN(that has a local search procedure based on a random selection between BP and random search) ([Curteanu et al. 2014](#)). Another training approach (Levenberg–Marquardt) was employed as LS procedure for DE by [Subudhi and Jena \(2009a, b, 2011\)](#), all variants being applied for non-linear system identification.

3.3 DE with global optimization and local search methods (global–global–local)

In this case, multiple algorithms belonging to different classes and types of search are used alternatively or concurrently for DE improvement, by locally and globally modifying the individuals.

In [Neri and Tirronen \(2008\)](#), two versions of DE improved with both global and local algorithm are proposed: enhanced memetic differential evolution (EMDE) and super-fit memetic differential evolution (SFMDE). In EMDE [an improvement of Memetic DE ([Tirronen et al. 2007](#))], based on a probabilistic scheme, Hooke–Jeeves algorithm, stochastic local searcher and simulated annealing are applied to randomly selected individuals or to the best one. In SFMDE, in order to be improved and to be included into the next generation, a series of individuals of DE undergo a PSO procedure. Along with PSO, two other algorithms, Nelder Mead algorithm (NMA) and Rosenbrock algorithm (RA), are used for improving a randomly selected individual and the solution of the current population.

[Wang et al. \(2009\)](#) takes the concept of the ant direction hybrid differential evolution (AHDE) of [Ji-Pyng et al. \(2004\)](#) further and adds an accelerated phase for faster convergence and an operator for simultaneously handling integer and real variables. When the best solution is not improved by the new individual creation mechanism, a gradient descent method is used to push the individual to a better point.

[Liao \(2010\)](#) modified the methodology proposed in [Angira and Babu \(2006\)](#), in which the current generation and the next generation are condensed into a single population of potential solutions. The modifications include the Deb's constraint handling method and the generalized discrete variables handling methods. This new version is hybridized and two new

algorithms are created by combining it with a LS operator and with another meta-heuristic method. The meta-heuristics used is HS and has the role of cooperating with DE in order to produce a desirable synergetic effect.

In Wang et al. (2010a), DE was combined with a new technique called generalized opposition-based learning (GOBL). After the initialization was performed and the best solutions (from the united population of randomly generated solutions and GOBL solutions) are selected, at each iteration, based on a specific probability, GOBL or DE is applied to evolve the population. The role of GOBL is to transform the current search space into a new search space, providing more opportunities for finding the global optimum. In addition, when GOBL strategy is not executed, a chaotic operator is applied to improve a set of the best individuals in the population.

The strengths of EA, DE and sequential quadratic programming (SQP) are combined in order to create a new powerful memetic algorithm (Singh and Ray 2011). Initially, the population is generated using random samples of the solution space and then evolved by either EA (including simulated binary crossover and polynomial mutation) or DE (Rand/1/exp). For certain generations, or when the local procedure was unable to find an improvement in the previous generation, SQP performs a local search, starting from a randomly selected individual. Otherwise, the starting point is represented by the best solution. If for more than a specified number of generations, the algorithm is not able to improve the objective value, then the population is reinitialized, the best solution so far being preserved.

3.4 DE as a local search procedure

The idea of applying a global search approach that is further improved by a local procedure is not new and it was used by various researchers for solving different types of problems. Therefore, a special attention must be given to the local approach, as it is the one generating the best solutions.

If in the previous sections the focus was on DE as a global search procedure that is hybridized with other algorithms to improve its performance, in this section, DE is regarded as LS, being applied to solve specific problems of other global methods. By limiting the new solutions around a specific region, the global search approach of DE is turned into efficient LS.

In combination with SQP, DE was employed to fine-tune the solutions found by GA when applied for economic dispatch with valve-point effect problems (He et al. 2008). The initial population of DE is randomly generated from the existing GA population, while SQP starts from a single solution. In a study tacking the application of EAs for solving multi-objective problems with complex Pareto sets, Li and Zhang (2009) modified the NSGA-II-SBX algorithm by replacing the SBX operator with a DE operator, followed by a polynomial mutation. A similar approach was used by Arabas et al. (2011) that introduced the differential mutation into a simple EAs without crossover operator. Each reproduced chromosome is first mutated by the DE scheme and after that, by the conventional Gaussian mutation scheme. Results on the problems from the CEC2005 competition showed that the performance of this new algorithm (DMEA) is comparable or better than other similar algorithms.

Liu et al. (2010) combined PSO with DE in order to solve numerical and engineering optimization problems. As PSO is prone to stagnation, the role of DE is to update the identified best positions of particles. Three mutation strategies (rand/1, current-to-best/1 and rand/2) are employed to produce three offsprings and a rule for boundary violation is enforced. The individual is replaced only if its offspring has a better fitness value and a lower degree of constraint violation. A similar approach to the one from Zhenya et al. (1998) is proposed in

Zhang and Xie (2003), where a hybrid PSO (called DEPSO), with a bell-shaped mutation and consensus in the population, is applied to a set of benchmark functions. Das et al. (2008) performed a technical analysis of PSO and DE, and studied a PSO version proposed by Zhenya et al. (1998). The main characteristic of the algorithm (PSO-DV) is represented by the introduction of the differential operator of DE into the scheme used for velocity update.

In order to search for the optimal path, DE is merged with ACO, its role being to produce new individuals with a random deviance disturbance that translates into an appropriate disturbance of pheromone quantity left by ants (Zhao et al. 2011).

In an attempt to improve the HS algorithm, Chakraborty et al. (2009) borrowed the mutation principle of DE. A similar approach is used in Arul et al. (2013), where a chaotic self-adaptive mutation operator replaced the pitch adjustment in order to enhance the HS search performance.

Hybridizing DE is an effective technique to improve performance. However, the hybrids are usually more complicated (Wenyin and Zhihua 2013) and therefore more expensive in terms of consumed resources. When performance is taken into account, this aspect is not an issue, as it can be observed from the multitude of combinations developed.

4 Conclusions

With a good performance and flexibility, DE is an algorithm that can be applied to solve different types of problems from many areas. In an attempt to improve its characteristics, a series of approaches are applied, the results demonstrating that it is an algorithm with a lot of potential. In this work, two improvement techniques are studied and the main findings reported in literature are listed in a chronological order. These techniques are represented by: (a) replacement of manual parameter settings with adaptive or self-adaptive variants; and (b) hybridization of DE with other algorithms.

Although initially considered as being easy to set up, numerous studies regarding the *CPs* optimal identification demonstrated that this is not an easy task as their values are problem dependent. Depending on when the setting is performed, two classes are encountered: tuning (the *CPs* are set before the algorithm starts) and control (the *CPs* are set during the run). In this work, the emphasis was on control and especially on self-adaptation, as this approach was proven the most efficient.

On what concerns hybridization, over the years, different approaches were proposed, a review of all possible combinations (at different levels and with all algorithms) being almost impossible due to the high number of studies published or being published. In this review, the focus was set on the main important publications from the last 5 years. Depending on the problem being solved and on the aspect being improved, the DE combinations can be performed at global level or local level. At global level, all the algorithms have an equal influence on the hybrid performance, the main classes of algorithms combined with DE being represented by the swarm intelligence and evolutionary algorithms. At a local level (also known as local search), DE can be improved not only with other heuristics, but also with problem specific approaches. Hybridization can also be performed between multiple algorithms. In this case, a third category is encountered: global–global–local. Although DE is a global search approach, it can be used at a local level or its main principles can be borrowed and introduced into other algorithms.

As it can be observed, DE has a complex dynamic with all the existing algorithms, its study in the context of hybridization being an important aspect that can influence the decision of choosing the appropriate method when solving difficult benchmark or real-life problems.

References

- Alguliev RM, Aliguliyev RM, Isazade NR (2012) DESAMC + DocSum: differential evolution with self-adaptive mutation and crossover parameters for multi-document summarization. *Knowl Based Syst* 36:21–38
- Ali M, Torn A (2002) Topographical differential evolution using pre-calculated differentials. In: Dzemyda G, Saltenis V, Zilinskas A (eds) *Stochastic and global optimization*. Springer, New York, pp 1–17
- Ali M, Pant M (2011) Improving the performance of differential evolution algorithm using Cauchy mutation. *Soft Comput* 15:991–1007
- Ali M, Pant M, Abraham A (2009) A hybrid ant colony differential evolution and its application to water resources problems. In: *World congress on nature and biologically inspired computing (NaBIC 2009)*, pp 1133–1138
- Ali M, Pant M, Nagar A (2010) Two local search strategies for Differential Evolution. In: *2010 IEEE fifth international conference on bio-inspired computing: theories and applications (BIC-TA)*, pp 1429–1435
- Angira R, Babu BV (2006) Optimization of process synthesis and design problems: a modified differential evolution approach. *Chem Eng Sci* 61:4707–4721
- Arabas J, Bartnik L, Opara K (2011) DMEA—an algorithm that combines differential mutation with the fitness proportionate selection. In: *2011 IEEE symposium on differential evolution (SDE)*. IEEE, pp 1–8
- Arul R, Ravi G, Velusami S (2013) Chaotic self-adaptive differential harmony search algorithm based dynamic economic dispatch. *Int J Electr Power Energ Syst* 50:85–96
- Asafuddoula M, Ray T, Sarker R (2014) An adaptive hybrid differential evolution algorithm for single objective optimization. *Appl Math Comput* 231:601–618
- Bandurski K, Kwedlo W (2010) A lamarckian hybrid of differential evolution and conjugate gradients for neural network training. *Neural Process Lett* 32:31–44
- Bhowmik P, Das S, Konar A, Das S, Nagar AK (2010) A new differential evolution with improved mutation strategy. In: *IEEE congress on evolutionary computation (CEC '10)*. IEEE, pp 1–8
- Blum C, Puchinger J, Raidl GR, Roli A (2011) Hybrid metaheuristics in combinatorial optimization: a survey. *Appl Soft Comput* 11:4135–4151
- Brest J, Greiner S, Boskovic B, Mernik M, Zumer V (2006) Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE Trans Evol Comput* 10:646–657
- Brest J, Boskovic B, Greiner S, Zumer V, Maucec M (2007) Performance comparison of self-adaptive and adaptive differential evolution algorithms. *Soft Comput* 11:617–629
- Brest J (2009) Constrained real-parameter optimization with e-self-adaptive differential evolution. In: Mezura-Montes E (ed) *Constraint-handling in evolutionary optimization*. Springer, Berlin, pp 73–93
- Brest J, Zamuda A, Fister I, Boskovic B, Maucec MS (2011) Constrained real-parameter optimization using a Differential Evolution algorithm. In: *2011 IEEE symposium on differential evolution (SDE)*. IEEE, pp 1–8
- Chakraborty P, Roy GG, Das S, Jain D, Abraham A (2009) An improved harmony search algorithm with differential mutation operator. *Fundam Inform* 95:401–426
- Chang L, Liao C, Lin W, Chen LL, Zheng X (2012) A hybrid method based on differential evolution and continuous ant colony optimization and its application on wideband antenna design. *Progr Electromagn Res* 122:105–118
- Chiang TC, Chen CN, Lin YC (2013) Parameter control mechanisms in differential evolution: a tutorial review and taxonomy. In: *2013 IEEE symposium on differential evolution (SDE)*. IEEE, pp 1–8
- Cruz-Ramirez M, Sanchez-Monedero J, Fernandez-Navarro F, Fernandez JC, Hervas-Martinez C (2010) Memetic pareto differential evolutionary artificial neural networks to determine growth multi-classes in predictive microbiology. *Evol Intell* 3:187–199
- Curteanu S, Suditu G, Buburuzan AM, Dragoi EN (2014) Neural networks and differential evolution algorithm applied for modelling the depollution process of some gaseous streams. *Environ Sci Pollut Res* 21:12856–12867
- da Silva EK, Barbarosa HJC (2010) A study of the combined use of differential evolution and genetic algorithms. *Mec Comput* XXIX:9541–9562
- Das S, Konar A, Chakraborty U (2005) Two improved differential evolution schemes for faster global search. ACM, New York
- Das S, Konar A, Chakraborty U (2007) Annealed Differential Evolution. In: *IEEE congress on evolutionary computation (CEC '07)*. IEEE, pp 1926–1933
- Das S, Abraham A, Konar A (2008) Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives. In: Liu Y, Sun A, Loh H, Lu W, Lim EP (eds) *Advances of computational intelligence in industrial systems*. Springer, Berlin, pp 1–38

- Das S, Abraham A, Chakraborty UK, Konar A (2009) Differential evolution using a neighborhood-based mutation operator. *IEEE Trans Evol Comput* 13:526–553
- Das S, Suganthan PN (2011) Differential evolution a survey of the state-of-the-art. *IEEE Trans Evol Comput* 15:4–31
- Davendra D, Onwubolu G (2009) Forward backward transformation. In: Onwubolu G, Davendra D (eds) *Differential evolution: a handbook for global permutation-based combinatorial optimization*. Springer, Berlin, pp 35–80
- Deng W, Yang X, Zou L, Wang M, Liu Y, Li Y (2013) An improved self-adaptive differential evolution algorithm and its application. *Chemom Intell Lab Syst* 128:66–76
- Dong MG, Wang N (2012) A novel hybrid differential evolution approach to scheduling of large-scale zero-wait batch processes with setup times. *Comput Chem Eng* 45:72–83
- dos Santos Coelho L, Mariani VC (2006) Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. *IEEE Trans Power Syst* 21:989–996
- dos Santos Coelho L, Mariani V (2008) Self-adaptive differential evolution using chaotic local search for solving power economic dispatch with nonsmooth fuel cost function. In: Chakraborty U (ed) *Advances in differential evolution*. Springer, Berlin, pp 275–286
- dos Santos Coelho L (2009) Reliability-redundancy optimization by means of a chaotic differential evolution approach. *Chaos Soliton Fract* 41:594–602
- dos Santos Coelho L, Sauer JG, Rudek M (2009) Differential evolution optimization combined with chaotic sequences for image contrast enhancement. *Chaos Soliton Fract* 42:522–529
- dos Santos Coelho L, de Andrade Bernert DL (2010) A modified ant colony optimization algorithm based on differential evolution for chaotic synchronization. *Expert Syst Appl* 37:4198–4203
- dos Santos GS, Luvizotto LGJ, Mariani VC, dos Santos Coelho L (2012) Least squares support vector machines with tuning based on chaotic differential evolution approach applied to the identification of a thermal process. *Expert Syst Appl* 39:4805–4812
- dos Santos Coelho L, Ayala HVH, Mariani VC (2014) A self-adaptive chaotic differential evolution algorithm using gamma distribution for unconstrained global optimization. *Appl Math Comput* 234:452–459
- Dragoi EN, Curteanu S, Fissore D (2012) Freeze-drying modeling and monitoring using a new neuro-evolutionary technique. *Chem Eng Sci* 72:195–204
- Dulikravich G, Moral R, Sahoo D (2005) A multi-objective evolutionary hybrid optimizer. *Evolutionary and deterministic methods for design, optimization, and control with applications to industrial and societal problems (EUROGEN 2005)*. FLM, Munich, pp 1–13
- Eiben AE, Hinterding R, Michalewicz Z (1999) Parameter control in evolutionary algorithms. *IEEE Trans Evol Comput* 3:124–141
- Eiben G, Schut MC (2008) New ways to calibrate evolutionary algorithms. In: Siarry P, Michalewicz Z (eds) *Advances in metaheuristics for hard optimization*. Springer, Berlin, pp 153–177
- Elsayed SM, Sarker RA, Essam DL (2011) Integrated strategies differential evolution algorithm with a local search for constrained optimization. In: *IEEE congress on evolutionary computation (CEC '11)*. IEEE, pp 2618–2625
- Epitropakis MG, Plagianakos VP, Vrahatis MN (2012) Evolving cognitive and social experience in particle swarm optimization through differential evolution: a hybrid approach. *Inf Sci* 216:50–92
- Fan HY, Lampinen J (2003) A trigonometric mutation operation to differential evolution. *J Glob Optim* 27:105–129
- Feoktistov V (2006) *Differential evolution: in search of solutions*. Springer, Berlin
- Feoktistov V, Janaqi S (2004) Hybridization of differential evolution with least-square support vector machine. In: *Proceedings of the annual machine learning conference of Belgium and the Netherlands (BENER-LEARN)*, pp 26–31
- Feoktistov V, Janaqi S (2006) New energetic selection principle in differential evolution. In: Seruca I, Cordeiro J, Hammoudi S, Filipe J (eds) *Enterprise information systems VI*. Springer, Dordrecht, pp 151–157
- Fister I, Mernik M, Brest J (2011) Hibridization of evolutionary algorithms. In: Kita E (ed) *Evolutionary algorithms*. InTech, pp 3–26. <http://www.intechopen.com/articles/show/title/hybridization-of-evolutionary-algorithms>
- Gamperle R, Muller S, Koumoutsakos P (2002) A parameter study for differential evolution. In: *International conference on advances in intelligent systems, fuzzy systems, evolutionary computation (WSEAS)*, pp 292–298
- Gong W, Cai Z, Ling C (2010) DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization. *Soft Comput* 15:645–665
- Guo J, Zhou J, Zou Q, Liu Y, Song L (2013) A novel multi-objective shuffled complex differential evolution algorithm with application to hydrological model parameter optimization. *Water Resour Manag* 27:2923–2946

- He D, Wang F, Mao Z (2008) A hybrid genetic algorithm approach based on differential evolution for economic dispatch with valve-point effect. *Int J Electr Power Energy Syst* 30:31–38
- Hendtlass T (2001) A combined swarm differential evolution algorithm for optimization problems. In: Monostori L, Vancza J, Ali M (eds) *Engineering of intelligent systems*. Springer, Berlin, pp 11–18
- Hernandez SA, Leguizamón G, Mezura-Montes E (2013) Hybridization of differential evolution using hill climbing to solve constrained optimization problems. *Rev Iberoam Intell Artif* 16:3–15
- Hernandez-Diaz AG, Santana-Quintero LV, Coello Coello C, Caballero R, Molina J (2006) A new proposal for multi-objective optimization using differential evolution and rough sets theory. In: *Proceedings of the 8th annual conference on genetic and evolutionary computation*. ACM, pp 675–682
- He RJ, Yang ZY (2012) Differential evolution with adaptive mutation and parameter control using Lévy probability distribution. *J Comput Sci Technol* 27:1035–1055
- Huang V, Qin A, Suganthan P, Tasgetiren M (2007) Multi-objective optimization based on self-adaptive differential evolution algorithm. *Constraints* 1:3
- Huang VL, Zhao SZ, Mallipeddi R, Suganthan PN (2009) Multi-objective optimization using self-adaptive differential evolution algorithm. In: *IEEE congress on evolutionary computation (CEC '09)*. IEEE, pp 190–194
- Hu C, Yan X (2009a) A novel adaptive differential evolution algorithm with application to estimate kinetic parameters of oxidation in supercritical water. *Eng Optim* 41:1051–1062
- Hu C, Yan X (2009b) An immune self-adaptive differential evolution algorithm with application to estimate kinetic parameters for homogeneous mercury oxidation. *Chin J Chem Eng* 17:232–240
- Ilonen J, Kamarainen JK, Lampinen J (2003) Differential evolution training algorithm for feed-forward neural networks. *Neural Process Lett* 17:93–105
- Islam M, Yao X (2008) Evolving artificial neural network ensembles. In: Fulcher J, Jain L (eds) *Computational intelligence: a compendium*. Springer, Berlin, pp 851–880
- Islam SM, Das S, Ghosh S, Roy S, Suganthan PN (2012) An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. *IEEE Trans Syst Man Cybern Part B Cybern* 42:482–500
- Jia D, Zheng G, Khurram Khan M (2011) An effective memetic differential evolution algorithm based on chaotic local search. *Inf Sci* 181:3175–3187
- Jingqiao Z, Sanderson AC (2008) Self-adaptive multi-objective differential evolution with direction information provided by archived inferior solutions. In: *IEEE congress on evolutionary computation (CEC 2008)*. IEEE, pp 2801–2810
- Ji-Pyng C, Chung-Fu C, Ching-Tzong S (2004) Ant direction hybrid differential evolution for solving large capacitor placement problems. *IEEE Trans Power Syst* 19:1794–1800
- Kaelo P, Ali MM (2007) Differential evolution algorithms using hybrid mutation. *Comput Optim Appl* 37:231–246
- Krasnogor N, Smith J (2005) A tutorial for competent memetic algorithms: model, taxonomy, and design issues. *IEEE Trans Evol Comput* 9:474–488
- Liao TW (2010) Two hybrid differential evolution algorithms for engineering design optimization. *Appl Soft Comput* 10:1188–1199
- Li H, Zhang Q (2009) Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans Evol Comput* 13:284–302
- Li G, Liu M (2010) The summary of differential evolution algorithm and its improvements. In: *3rd international conference on advanced computer theory and engineering (ICACTE)*, p V3–153
- Liu J, Lampinen J (2005) A fuzzy adaptive differential evolution algorithm. *Soft Comput* 9:448–462
- Liu H, Cai Z, Wang Y (2010) Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Appl Soft Comput* 10:629–640
- Lu Y, Zhou J, Qin H, Li Y, Zhang Y (2010a) An adaptive hybrid differential evolution algorithm for dynamic economic dispatch with valve-point effects. *Expert Syst Appl* 37:4842–4849
- Lu Y, Zhou J, Qin H, Wang Y, Zhang Y (2010b) An adaptive chaotic differential evolution for the short-term hydrothermal generation scheduling problem. *Energy Convers Manag* 51:1481–1490
- Lu Y, Zhou J, Qin H, Wang Y, Zhang Y (2011) Chaotic differential evolution methods for dynamic economic dispatch with valve-point effects. *Eng Appl Artif Intell* 24:378–387
- Mallipeddi R, Suganthan PN, Pan QK, Tasgetiren MF (2011) Differential evolution algorithm with ensemble of parameters and mutation strategies. *Appl Soft Comput* 11:1679–1696
- Meena KY, Shashank S, Singh PV (2012) Text documents clustering using genetic algorithm and discrete differential evolution. *Int J Comput Appl* 43:16–19
- Menon P, Bates D, Postlethwaite I, Marcos A, Fernandez V, Bennani S (2008) Worst case analysis of control law for re-entry vehicles using hybrid differential evolution. In: Chakraborty U (ed) *Advances in differential evolution*. Springer, Berlin, pp 319–333

- Mezura-Montes E, Palomeque-Ortiz A (2009) Self-adaptive and deterministic parameter control in differential evolution for constrained optimization. In: Mezura-Montes E (ed) *Constraint-handling in evolutionary optimization*. Springer, Berlin, pp 95–120
- Michalski KA (2001) Electromagnetic imaging of elliptical-cylindrical conductors and tunnels using a differential evolution algorithm. *Microw Opt Technol Lett* 28:164–169
- Mohamed AW, Sabry HZ, Abd-Elaziz T (2013) Real parameter optimization by an effective differential evolution algorithm. *Egypt Inf J* 14:37–53
- Neri F, Tirronen V (2008) On memetic Differential Evolution frameworks: a study of advantages and limitations in hybridization. In: *IEEE world congress on computational intelligence (CEC 2008)*. IEEE, pp 2135–2142
- Neri F, Tirronen V (2009) Scale factor local search in differential evolution. *Memet Comput* 1:153–171
- Neri F, Tirronen V (2010) Recent advances in differential evolution: a survey and experimental analysis. *Artif Intell Rev* 33:61–106
- Nian X, Wang Z, Qian F (2013) A hybrid algorithm based on differential evolution and group search optimization and its application on ethylene cracking furnace. *Chin J Chem Eng* 21:537–543
- Nicoara ES (2009) Mechanisms to avoid the premature convergence of genetic algorithms. *Bul Univ Petro-Gaze Ploiesti* 61:87–96
- Nobakhti A, Wang H (2008) A simple self-adaptive differential evolution algorithm with application on the ALSTOM gasifier. *Appl Soft Comput* 8:350–370
- Nocedal J, Wright SJ (2006) *Introduction. Numerical optimization*, Springer, New York
- Noman N, Iba H (2008) Accelerating differential evolution using an adaptive local search. *IEEE Trans Evol Comput* 12:107–125
- Pan QK, Suganthan PN, Wang L, Gao L, Mallipeddi R (2011) A differential evolution algorithm with self-adapting strategy and control parameters. *Comput Oper Res* 38:394–408
- Pandiarajan K, Babulal CK (2014) Transmission line management using hybrid differential evolution with particle swarm optimization. *J Electr Syst* 10:21–35
- Pant M, Thangaraj R, Abraham A, Grosan C (2009) *Differential evolution with Laplace mutation operator*. Proceedings of the eleventh conference on congress on evolutionary computation. IEEE Press, New York, pp 2841–2849
- Peng L, Wang Y (2010) Differential evolution using uniform-quasi-opposition for initializing the population. *Inf Technol J* 9:1629–1634
- Price K, Storn R, Lampinen J (2005) *Differential evolution. A practical approach to global optimization*, Springer, Berlin
- Price K (2008) Eliminating drift bias from the differential evolution algorithm. In: Chakraborty U (ed) *Advances in differential evolution*. Springer, Berlin, pp 33–88
- Qian B, Wang L, Hu R, Wang WL, Huang DX, Wang X (2008) A hybrid differential evolution method for permutation flow-shop scheduling. *Int J Adv Manuf Technol* 38:757–777
- Qin AK, Suganthan PN (2005) Self-adaptive differential evolution algorithm for numerical optimization. In: *IEEE congress on evolutionary computation (CEC 2005)*, pp 1785–1791
- Rahmat NA, Musirin I (2013) Differential evolution immunized ant colony optimization technique in solving economic load dispatch problem. *Engineering* 5:157–162
- Rahmat NA, Musirin I, Abidin AF (2014) Differential evolution immunized ant colony optimization (DEIANT) technique in solving weighted economic load dispatch problem. *Asian Bull Eng Sci Technol* 1:17–26
- Raidl G (2006) A unified view on hybrid metaheuristics. In: Roli A, Sampels M (eds) *Almeida F, Blesa Aguilera M, Blum C, Moreno Vega J, Perez Perez M. Hybrid metaheuristics*. Springer, Berlin, pp 1–12
- Rogalsky T, Derksen RW (2000) Hybridization of differential evolution for aerodynamic design. In: *Proceedings of the 8th annual conference of the Computational Fluid Dynamics Society of Canada*, Canada, pp 729–736
- Ronkkonen J, Kukkonen S, Price KV (2005) Real-parameter optimization with differential evolution. In: *IEEE congress on evolutionary computation (CEC 2005)*. IEEE, pp 506–513
- Salman A, Engelbrecht AP, Omran MGH (2007) Empirical analysis of self-adaptive differential evolution. *Eur J Oper Res* 183:785–804
- Santana-Quintero LV, Hernandez-Díaz AG, Molina J, Coello Coello CA, Caballero R (2010) DEMORS: a hybrid multi-objective optimization algorithm using differential evolution and rough set theory for constrained problems. *Comput Oper Res* 37:470–480
- Sarangi PP, Sahu A, Panda M (2013) A hybrid differential evolution and back-propagation algorithm for feedforward neural network training. *Int J Comput Appl* 84:1–9
- Segura C, Coello Coello C, Segredo E, Leon C (2015) On the adaptation of the mutation scale factor in differential evolution. *Optim Lett* 9:189–198

- Singh HK, Ray T (2011) Performance of a hybrid EA-DE-memetic algorithm on CEC 2011 real world optimization problems. In: IEEE congress on evolutionary computation (CEC 2011). IEEE, pp 1322–1326
- Storn R (1996) On the usage of differential evolution for function optimization. In: 1996 biennial conference of the North American Fuzzy Information Processing Society (NAFIPS), pp 519–523
- Storn R (2008) Differential evolution research-trends and open questions. In: Chakraborty U (ed) *Advances in differential evolution*. Springer, Berlin, pp 1–31
- Storn R, Price K (1995) Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces. *International Science Computer Institute, Berkley*
- Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11:341–359
- Subudhi B, Jena D (2009a) An improved differential evolution trained neural network scheme for nonlinear system identification. *Int J Autom Comput* 6:137–144
- Subudhi B, Jena D (2009b) Nonlinear system identification using opposition based learning differential evolution and neural network techniques. *IEEE J Intell Cybern Syst* 1:1–13
- Subudhi B, Jena D (2011) A differential evolution based neural network approach to nonlinear system identification. *Appl Soft Comput* 11:861–871
- Takahama T, Sakai S (2012) Efficient constrained optimization by the e constrained rank-based differential evolution. In: IEEE congress on evolutionary computation (CEC 2012). IEEE, pp 1–8
- Tan Y-Y, Jiao YC, Li H, Wang XK (2012) A modification to MOEA/D-DE for multiobjective optimization problems with complicated Pareto sets. *Inf Sci* 213:14–38
- Tardivo ML, Cagnina L, Leguizamón G (2012) A hybrid metaheuristic based on differential evolution and local search with quadratic interpolation. In: XVIII Congreso Argentino de Ciencias de la Computación, pp 1–10
- Teo J (2006) Exploring dynamic self-adaptive populations in differential evolution. *Soft Comput* 10:673–686
- Thangaraj R, Pant M, Abraham A (2009a) A simple adaptive Differential Evolution algorithm. In: *World congress on nature and biologically inspired computing (NaBIC 2009)*, pp 457–462
- Thangaraj R, Pant M, Abraham A, Badr Y (2009b) Hybrid evolutionary algorithm for solving global optimization problems. In: Corchado E, Wu X, Oja E, Herrero A, Baruaque B (eds) *Hybrid artificial intelligence systems*. Springer, Berlin, pp 310–318
- Thangraj R, Pant M, Abraham A, Deep K, Snaes V (2010) Differential evolution using a localized Cauchy mutation operator. In: *IEEE international conference on systems man and cybernetics (SMC)*. IEEE, pp 3710–3716
- Tirronen V, Neri F, Karkkainen T, Majava K, Rossi T (2007) A memetic differential evolution in filter design for defect detection in paper production. In: Giacobini M (ed) *Applications of evolutionary computing*. Springer, Berlin, pp 320–329
- Tvrđik J (2009) Adaptation in differential evolution: a numerical comparison. *Appl Soft Comput* 9:1149–1155
- Vaisakh K, Srinivas LR (2011) Evolving ant direction differential evolution for OPF with non-smooth cost functions. *Eng Appl Artif Intell* 24:426–436
- Wang SK, Chiou JP, Liu CW (2009) Parameters tuning of power system stabilizers using improved ant direction hybrid differential evolution. *Int J Electr Power Energy Syst* 31:34–42
- Wang J, Wu Z, Wang H (2010a) Hybrid differential evolution algorithm with chaos and generalized opposition-based learning. In: Cai Z, Hu C, Kang Z, Liu Y (eds) *Advances in computation and intelligence*. Springer, Berlin, pp 103–111
- Wang L, Pan QK, Suganthan PN, Wang WH, Wang YM (2010b) A novel hybrid discrete differential evolution algorithm for blocking flow shop scheduling problems. *Comput Oper Res* 37:509–520
- Wang YN, Wu LH, Yuan XF (2010c) Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure. *Soft Comput* 14:193–209
- Wang L, Xu Y, Li L (2011) Parameter identification of chaotic systems by hybrid Nelder-Mead simplex search and differential evolution algorithm. *Expert Syst Appl* 38:3238–3245
- Wang X, Xu G (2011) Hybrid differential evolution algorithm for traveling salesman problem. *Procedia Eng* 15:2716–2720
- Wang L, Li L (2012) A coevolutionary differential evolution with harmony search for reliability-redundancy optimization. *Expert Syst Appl* 39:5271–5278
- Wang C, Gao JH (2014) A differential evolution algorithm with cooperative coevolutionary selection operation for high-dimensional optimization. *Optim Lett* 8:477–492
- Wenyn G, Zhihua C (2013) Differential evolution with ranking-based mutation operators. *IEEE Trans Cybern* 43:2066–2081
- Wu L, Wang Y, Zhou S, Yuan X (2007) Self-adapting control parameters modified differential evolution for trajectory planning of manipulators. *J Control Theory Appl* 5:365–373

- Xiangyin Z, Haibin D, Jiqiang J (2008) DEACO: hybrid ant colony optimization with differential evolution. In: IEEE world congress on computational intelligence (CEC). IEEE, pp 921–927
- Xin B, Chen J, Zhang J, Fang H, Peng ZH (2012) Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: a review and taxonomy. *IEEE Tran Syst Man Cybern Part C Appl Rev* 42:744–767
- Xu W, Zhang L, Gu X (2012) Modeling of ammonia conversion rate in ammonia synthesis based on a hybrid algorithm and least squares support vector regression. *Asia Pac J Chem Eng* 7:150–158
- Xue F, Sanderson AC, Bonissone PP, Graves RJ (2005) Fuzzy logic controlled multi-objective differential evolution. In: The 14th IEEE international conference on fuzzy systems (FUZZ '05). IEEE, pp 720–725
- Yang Z, Tang K, Yao X (2008a) Self-adaptive differential evolution with neighborhood search. In: IEEE world congress on computational intelligence (CEC 2008). IEEE, pp 1110–1116
- Yang Z, Yao X, He J (2008b) Making a difference to differential evolution. In: Siarry P, Michalewicz Z (eds) *Advances in metaheuristics for hard optimization*. Springer, Berlin, pp 397–414
- Yu Wj, Zhang J (2012) Adaptive differential evolution with optimization state estimation. In: Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference. ACM, pp 1285–1292
- Yu X, Cao J, Shan H, Zhu L, Guo J (2014) An adaptive hybrid algorithm based on particle swarm optimization and differential evolution for global optimization. *Sci World J* 2014:1–16
- Yuan X, Cao B, Yang B, Yuan Y (2008) Hydrothermal scheduling using chaotic hybrid differential evolution. *Energy Convers Manag* 49:3627–3633
- Yulin Z, Qian Y, Chunguang Z (2010) Distribution network reactive power optimization based on ant colony optimization and differential evolution algorithm. In: 2nd IEEE international symposium on power electronics for distributed generation systems (PEDG), pp 472–476
- Zade AH, Mohammadi SMA, Gharaveisi AA (2011) Fuzzy logic controlled differential evolution to solve economic load dispatch problems. *J Adv Comput Res* 2:29–40
- Zaharie D (2002a) Critical values for the control parameters of differential evolution algorithms. In: Proceedings of 8th international conference on soft computing (MENDEL 2002), pp 62–67
- Zaharie D (2002b) Parameter adaptation in differential evolution by controlling the population diversity. In: Proceedings of the international workshop on symbolic and numeric algorithms for scientific computing. pp 385–397
- Zaharie D (2003) Control of population diversity and adaptation in differential evolution algorithms. In: Proceedings of the 9th international conference on soft computing (MENDEL 2003), pp 41–46
- Zaharie D (2009) Influence of crossover on the behavior of differential evolution algorithms. *Appl Soft Comput* 9:1126–1138
- Zaharie D, Petcu D (2004) *Adaptive pareto differential evolution and its parallelization*. Parallel processing and applied mathematics, Springer, Berlin
- Zhang W-J, Xie X-F (2003) DEPSO: hybrid particle swarm with differential evolution operator. IEEE international conference on systems, man and cybernetics (SMCC). IEEE, Washington, DC, USA, pp 3816–3821
- Zhang J, Sanderson AC (2009a) JADE: adaptive differential evolution with optional external archive. *IEEE Trans Evol Comput* 13:945–958
- Zhang J, Sanderson AC (2009b) *Adaptive differential evolution: a robust approach to multimodal problem optimization*. Springer, Berlin
- Zhang R, Wu C (2011) A hybrid differential evolution and tree search algorithm for the job shop scheduling problem. *Math Probl Eng* 2011:20
- Zhang X, Chen W, Dai C, Cai W (2010) Dynamic multi-group self-adaptive differential evolution algorithm for reactive power optimization. *Int J Electr Power Energy Syst* 32:351–357
- Zhang C, Chen J, Xin B, Cai T, Chen C (2011) Differential evolution with adaptive population size combining lifetime and extinction mechanisms. In: 8th Asian control conference (ASCC), pp 1221–1226
- Zhao YL, Yu Q, Zhao CG (2011) Distribution network reactive power optimization based on ant colony optimization and differential evolution algorithm. *J Energy Power Eng* 5:548–553
- Zhao C, Xu Q, Lin S, Li X (2013) Hybrid differential evolution for estimation of kinetic parameters for biochemical systems. *Chin J Chem Eng* 21:155–162
- Zhenya H, Chengjian W, Luxi Y, Xiqi G, Susu Y, Eberhart RC, Shi Y (1998) Extracting rules from fuzzy neural network by particle swarm optimisation. In: The 1998 IEEE international conference on computational intelligence. IEEE, pp 74–77
- Zhenyu Y, Ke T, Xin Y (2008) Self-adaptive differential evolution with neighborhood search. In: IEEE congress on evolutionary computation (CEC 2008), pp 1110–1116