Generalized Differential Evolution for Numerical and Evolutionary Optimization

Saku Kukkonen and Carlos A. Coello Coello

Abstract This chapter is about Generalized Differential Evolution (GDE), which is a general purpose optimizer for global nonlinear optimization. It is based on Differential Evolution (DE), which has been gaining popularity because of its simplicity and good observed performance. GDE extends DE for problems with several objectives and constraints. The chapter concentrates on describing different development phases and performance of GDE but it also contains a brief listing of other multi-objective DE approaches. Ability to solve multi-objective problems is mainly discussed, but constraint handling and the effect of control parameters are also covered. It is found that the latest GDE version is effective and efficient for solving constrained multi-objective problems having different types of decision variables.

Keywords Multi-objective optimization · Constrained optimization · Differential evolution · Generalized differential evolution

1 Introduction

Multi-objective optimization means the simultaneous optimization of more than one objective opposite to single-objective optimization where one objective is optimized [19, p. 1]. Many practical problems are multi-objective by their nature but in the past they have been converted into a single-objective form to ease the optimization process [19, p. 3]. Improved optimization techniques and greater computing power have made it possible to solve many problems in their original, multi-objective form.

S. Kukkonen (✉)
Centre of Computational Engineering and Integrated Design (CEID),
Lappeenranta University of Technology, P.O. Box 20, 53851 Lappeenranta, Finland
e-mail: dr.saku.kukkonen@gmail.com

C.A. Coello Coello
Computer Science Department, CINVESTAV-IPN, Av. IPN No. 2508, San Pedro Zacatenco,
07300 Mexico City, DF, Mexico
e-mail: ccoello@cs.cinvestav.mx

O. Schütze et al. (eds.), *NEO 2015*, Studies in Computational
Intelligence 663, DOI 10.1007/978-3-319-44003-3_11

Multi-objective optimization has, therefore, become an important research topic in the field of optimization.

Evolutionary algorithms (EAs) [6] are population based stochastic optimization methods that are inspired on Darwin's evolution theory. EAs are able to deal with difficult objective functions, which are, e.g., discontinuous, non-convex, multi-modal, nonlinear, and non-differentiable, and which pose difficulties to most traditional optimization methods. Since many practical problems include such difficult objectives, EAs have become popular during the last couple of decades. Developments in computer technology have also facilitated the use of EAs.

EAs have become popular also in multi-objective optimization since EAs are capable of providing multiple solution candidates during the search process that is desirable with a multi-objective optimization problem (MOOP). Some of the most well known but already older multi-objective EAs (MOEAs) are the elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) [21] and the improved Strength Pareto Evolutionary Algorithm (SPEA2) [99]. Some later MOEAs are the *S* Metric Selection Evolutionary Multiobjective Optimization Algorithm (SMS-EMOA) [26], Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D) [94], and Many-Objective NSGA-II (NSGA-III) [20].

Differential Evolution (DE) was proposed in 1995 [83] and it has been gaining popularity because of its simplicity and good observed performance. Several extensions of DE for multi-objective optimization have already been proposed. Some early approaches just transformed a MOOP into a single-objective problem and use DE to solve the single-objective problem [5, 11, 89], whereas more recent and advanced approaches use mainly the concept of Pareto-dominance. The next section mentions most of the multi-objective DE approaches as well as constraint handling techniques used with DE.

The starting point of Generalized Differential Evolution (GDE) was an idea on how DE could be extended to handle multiple constraints and objectives just by modifying the selection rule of DE. The idea was originally proposed in [55] but implemented and tested for [44]. When the results of the initial investigation were published in [44], the method was named Generalized DE as it was generalization of DE for multiple objectives and constraints. Already during preliminary studies it was found that for good performance the method needed different control parameter values than usually adopted with single-objective DE and the diversity of the obtained solutions could have been better. Therefore, work continued by studying the effect of control parameter values and by developing the diversity preservation part of the method. Different GDE versions differ mainly in their ability to maintain the diversity of the solutions.

The rest of the chapter is organized as follows: In Sect. 2, the concept of multi-objective optimization with constraints is handled briefly. Also, basic DE and its extensions for multi-objective and constrained optimization have been covered. Section 3 describes different development phases of GDE. Subjects of future work are given in Sect. 4, and finally conclusions are drawn in Sect. 5.

2 Background and Related Studies

This section contains information about multi-objective optimization with constraints and evolutionary computation. The basic Differential Evolution and its extensions to constrained multi-objective optimization are also covered.

2.1 Multi-objective Optimization with Constraints

Many practical problems have multiple objectives and several aspects create constraints to problems. For example, mechanical design problems may have several objectives such as obtained performance and manufacturing costs, and available resources may be limited. Constraints can be divided into boundary constraints and constraint functions [56]. Boundary constraints are used when the value of a decision variable is limited to some range. Constraint functions represent more complicated constraints that are typically divided into equality and inequality constraints.

Mathematically, an inequality constrained multi-objective optimization problem can be presented in the form [65, p. 37]:

$$\begin{aligned} \text{minimize} \ & \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})\} \\ \text{subject to} \ & g_1(\mathbf{x}) \leq 0 \\ & g_2(\mathbf{x}) \leq 0 \\ & \vdots \\ & g_K(\mathbf{x}) \leq 0 \\ & \mathbf{x} \in \mathbf{R}^D. \end{aligned} \tag{1}$$

Thus, there are M objective functions f_m to be minimized, K inequality constraints presented with functions limiting the search space, and D decision variables. Decision variables, which define values of objectives and constraints, form together a decision vector $\mathbf{x}$. The goal of multi-objective optimization is to find a decision vector $\mathbf{x}$, which minimizes the objectives without violating any of the constraints.

Usually, the objectives are conflicting and it is not possible to find a single solution that would be optimal for all the objectives [19, pp. 1–2]. Therefore, there will be a set of solutions which represent the best possible compromises between different objectives. For such solutions it holds that none of the objectives can be improved without impairing at least one other objective. This definition is commonly known as Pareto-optimality after a nineteen century scientist, Vilfredo Pareto. Pareto-optimal solutions constitute the so-called Pareto optimal set. The image of the Pareto optimal set in the objective space is called the Pareto front [17, pp. 11–12]. Finding Pareto-optimal solutions for a MOOP is sometimes also called Pareto-optimization [37]. The goal of Pareto-optimization is to find a set of solutions approximating the Pareto front having both a proper convergence and a distribution/diversity, as uniform as possible, along the Pareto front [17, pp. 3–4]

Here we define that if there exist decision vectors $\mathbf{x}$ and $\mathbf{y}$ such that

$$\forall m \in \{1, 2, \ldots, M\} : f_m(\mathbf{x}) \leq f_m(\mathbf{y}) \ , \tag{2}$$

then $\mathbf{x}$ weakly (Pareto-)dominates $\mathbf{y}$ and this is expressed as $\mathbf{x} \preceq \mathbf{y}$. If

$$\mathbf{x} \preceq \mathbf{y} \quad \wedge \quad \exists m \in \{1, 2, \ldots, M\} : f_m(\mathbf{x}) < f_m(\mathbf{y}) \ , \tag{3}$$

then $\mathbf{x}$ (Pareto-)dominates $\mathbf{y}$ and this is expressed as $\mathbf{x} \prec \mathbf{y}$.

Solution candidates can be sorted based on dominance with non-dominated sorting [19, pp. 40–44]. Solutions which belong to the same non-domination level do not dominate each other. When solution candidates are sorted, non-dominated solutions of all the solution candidates form the first non-domination level, non-dominated solutions of the rest of the solution candidates form the second non-domination level, and so on.

Many optimization algorithms are capable of creating solution candidates outside the original initialization bounds of the decision variables. If a problem definition contains boundaries for decision variable values, then some technique must be used for handling boundary constraint violations. One could just reject the violating solution candidate and create a new one. Decision variable values can be also corrected according to some rule to be inside the boundaries as described in [74, pp. 202–206]. A value that violates a constraint can be reset to the violated boundary or between boundaries. One approach is to reflect any violation back into the feasible solution area by the same amount by which the boundary was violated and this approach is used in this chapter.

More complicated constraints are presented in the form of functions on one side of the inequality having zero on the other side. A classical method to handle inequality constraints is a penalty function approach [19, p. 127]. The idea is to penalize an infeasible solution by increasing the value of the corresponding objective function by the constraint violation. This approach needs penalty parameters for constraint violation that is not trivial and different penalty parameter values lead to different results.

Parameter free approaches also exist and have been becoming popular lately since they do not have the problem of choosing or adjusting appropriate parameter values. Often these techniques are based on the following simple principles when comparing two solutions at a time [19, pp. 131–132]:

1. A feasible solution is better than an infeasible solution.
2. Among two feasible solutions, the better one has a better objective value.
3. Among two infeasible solutions, the better one violates the constraints less.

The difference between the approaches is in the case of two infeasible solutions. One popular approach is to add constraint violations together and compare the sums as used by Deb [19, p. 131]. Another approach is to use the dominance-relation in the space of constraint violations [56]: If constraint violations of solution $\mathbf{x}$ dominate constraint violations of solution $\mathbf{y}$,[1] then $\mathbf{x}$ is considered to be better.

The principles above can be extended to multiple objectives to have a constrain-domination relation. *Constrain-domination* $\prec_c$ is defined here such that $\mathbf{x}$ constrain-dominates $\mathbf{y}$, i.e., $\mathbf{x} \prec_c \mathbf{y}$ iff any of the following conditions is true [38]:

1. $\mathbf{x}$ and $\mathbf{y}$ are infeasible and $\mathbf{x}$ dominates $\mathbf{y}$ in respect to constraint violations.
2. $\mathbf{x}$ is feasible and $\mathbf{y}$ is not.
3. $\mathbf{x}$ and $\mathbf{y}$ are feasible and $\mathbf{x}$ dominates $\mathbf{y}$ in the objective function space.

The definition for weak constrain-domination $\preceq_c$ is analogous when the dominance relation is changed to weak dominance in the above definition.

2.2 The Elitist Non-dominated Sorting Genetic Algorithm

The elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) [21] has been the most used and cited MOEA. The working principle of NSGA-II is as follows: At each generation, a GA is used to create a child population which has an equal size compared to the parent population. After a generation, the parent and child populations are combined together. If the population size is *NP*, then the combined population has size 2*NP*. The combined population is sorted using non-dominated sorting and the best *NP* individuals are selected based on non-domination level. Thus, individuals from the best non-domination level are selected first, then individuals from the second best non-domination level, and so on until the number of selected individuals is *NP*. Probably, the number of solutions in the last non-domination level to be selected is too big to fit totally into the set of *NP* individuals. Then the number of solutions is reduced based on a crowding estimation among the individuals of the last non-domination level to be selected. The idea is to remove the most crowded individuals until the remaining individuals fit into the selected set of *NP* individuals.

Crowding estimation in NSGA-II is based on a distance metric called the *crowding distance*. The crowding distance for a member of a non-dominated set tries to approximate the perimeter of a cuboid formed by using the nearest neighbors of the member. For a member of a non-dominated set, the crowding distance is calculated by finding the distance between two nearest solutions on either side of the member along each of the objectives. These distances are normalized by dividing them by the difference between the maximum and minimum values of the corresponding objectives, and then these normalized distances are summed up giving a crowding distance for the corresponding member. For those non-dominated members which have a maximum or minimum value for any objective, the crowding distance is assigned to have an infinite value, i.e., those members are considered as the least crowded and removed last. Finally, the members of the non-dominated set are sorted

[1] We define that $\mathbf{x}$ dominates $\mathbf{y}$ with respect to constraints iff $\forall k : g'_k(\mathbf{x}) \leq g'_k(\mathbf{y}) \wedge \exists k : g'_k(\mathbf{x}) < g'_k(\mathbf{y}), \quad g'_k(\mathbf{z}) = \max\left(g_k(\mathbf{z}), 0\right)$.

in monotonically decreasing order according to the crowding distances and a desired number of members having the smallest crowding distance values are removed. It should be noted that pruning based on diversity is done only among the members of the last non-domination level of solutions that is to be selected for the next generation.

In [19, pp. 245–246], it is claimed that with early generations there exist several different non-domination levels and the diversity preservation has only a little effect on the selection process. When the population starts to converge to the Pareto front, the non-dominated sets become larger and eventually it is likely that the number of solutions in the first non-domination level is larger than *NP*. Thus, only little diversity preservation is performed at the early generations but more during the late generations. This kind of strategy gives a nice way to balance between convergence and diversity, but unfortunately, it works only with a low number of objectives, because the crowding distance metric used in NSGA-II does not estimate crowding well when the number of objectives is more than two [41]. Even if there were a working diversity preservation technique, the balance between convergence and diversity changes when the number of objectives increases. When the number of objectives increases, the proportion of non-dominated members in the population will also increase rapidly and the selection based on Pareto-dominance is not able to sort the members and diversity preservation becomes a dominating operation in the survival selection [52]. Therefore it has became evident that, NSGA-II in its original form performs well only with problems having two objectives.

2.3 Basic Differential Evolution, DE/rand/1/bin

Basic DE is meant for unconstrained single-objective optimization and therefore notations in this section are for single-objective optimization. As in a typical EA, the idea in DE is to start with a randomly generated initial population, which is then improved using selection, mutation, and crossover operations. Several ways exist to determine a termination criterion for an EA, but usually a predefined upper limit G_{max} for the number of generations to be computed is used.

2.3.1 Initialization of the Population

Values for the initial population in DE are typically drawn from a uniform distribution. Formally this can be presented as [73]:

$$\begin{aligned}
&P_G = \left\{\mathbf{x}_{1,G}, \mathbf{x}_{2,G}, \ldots, \mathbf{x}_{NP,G}\right\}, \quad \mathbf{x}_{i,G} = \left(x_{1,i,G}, x_{2,i,G}, \ldots, x_{D,i,G}\right) \\
&x_{j,i,0} = x_j^{(lo)} + rand_j[0,1] \cdot \left(x_j^{(hi)} - x_j^{(lo)}\right) \\
&i = 1, 2, \ldots, NP, \quad NP \geq 4, \quad j = 1, 2, \ldots, D\,.
\end{aligned} \tag{4}$$

In this representation, P_G denotes a population after G generations (0 is the index of an initial generation), $\mathbf{x}_{i,G}$ denotes a decision vector (or individual) of the population, and $rand_j[0, 1]$ denotes a uniformly distributed random variable in the value range $[0, 1]$. Terms $x_j^{(lo)}$ and $x_j^{(hi)}$ denote lower and upper parameter bounds in initialization, respectively. The size of the population is denoted by NP and the dimension of decision vectors is denoted by D.

2.3.2 Mutation and Crossover

DE goes through each decision vector $\mathbf{x}_{i,G}$ of the population and creates a corresponding trial vector $\mathbf{u}_{i,G}$ as follows [73]:

$$
\begin{array}{l}
r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}\,, \\
\text{(randomly selected,} \\
\quad \text{except mutually different and different from } i) \\
j_{rand} \in \{1, 2, \ldots, D\} \\
\text{for}(j = 1; j \leq D; j = j + 1) \\
\{ \\
\qquad \text{if}(rand_j[0, 1) < CR \vee j == j_{rand}) \\
\qquad\qquad u_{j,i,G} = x_{j,r_3,G} + F \cdot \left(x_{j,r_1,G} - x_{j,r_2,G}\right) \\
\qquad \text{else} \\
\qquad\qquad u_{j,i,G} = x_{j,i,G} \\
\}.
\end{array}
$$

Indices r_1, r_2, and r_3 are mutually different and drawn from the set of the population indices. Functions $rand_i[0, 1)$ and $rand_j[0, 1)$ return a random number drawn from the uniform distribution between 0 and 1 for each different i and j. Both CR and F are user defined control parameters for the DE algorithm and they remain fixed during the whole execution of the algorithm. Parameter CR, controlling the crossover operation, represents the probability that an element for the trial vector is chosen from a linear combination of three randomly chosen vectors and not from the old decision vector $\mathbf{x}_{i,G}$. The condition $j == j_{rand}$ ensures that at least one element of the trial vector is different compared to the elements of the old vector. Parameter F is a scaling factor for mutation and its value is typically $(0, 1+]$ (i.e., larger than 0 and the upper limit is in practice around 1 although there is no hard upper limit). Effectively, CR controls the rotational invariance of the search,[2] and its small value (e.g., 0.1) is more suitable with separable problems while larger values (e.g., 0.9) are for non-separable problems [73]. Control parameter F controls the speed and robustness of the search, i.e., a lower value for F increases the convergence rate but also the risk of getting

[2] The search is rotationally invariant if it is independent from the rotation of coordinate axis of the search space. Rotationally invariant search is preferable if the problem is not separable as it is the case with most practical problems [58, 78].

stuck into a local optimum [73]. Parameters *CR* and *NP* have a similar effect on the convergence rate as F [49, 73].

The difference between two randomly chosen vectors, $\mathbf{x}_{r_1,G} - \mathbf{x}_{r_2,G}$, defines the magnitude and direction of mutation. When the difference is added to a third randomly chosen vector $\mathbf{x}_{r_3,G}$, this change corresponds to mutation of this third vector. The basic idea of DE is that the mutation is self-adaptive to the objective function space and to the current population. At the beginning of the optimization process with DE, the magnitude of mutation is large because vectors in the population are far away from each other in the search space. When the evolution proceeds and the population converges, the magnitude of mutations gets smaller.

The self-adaptive mutation of DE allows to perform both global and local search. Other strengths are its simplicity, linear scalability (i.e., computational cost of the algorithm increases linearly with the number of decision variables), and ability to perform a rotationally invariant search.

2.3.3 Selection

After each mutation and crossover operation the trial vector $\mathbf{u}_{i,G}$ is compared to the old decision vector $\mathbf{x}_{i,G}$. If the trial vector has equal[3] or lower objective value, then it replaces the old vector. This can be presented as follows [73]:

$$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G} & \text{if} \quad f(\mathbf{u}_{i,G}) \leq f(\mathbf{x}_{i,G}) \\ \mathbf{x}_{i,G} & \text{otherwise} \end{cases} . \tag{5}$$

The average objective value of the population will never deteriorate, because the trial vector replaces the old vector only if it has equal or lower objective value. Therefore, DE is an elitist search method.

2.3.4 Overall Algorithm

The overall presentation of basic DE (sometimes also referred to as "classic DE") is presented in Fig. 1 [73]. This DE strategy is identified with the notation DE/rand/1/bin in the DE literature. In this notation,'rand' indicates how the vector for mutation is selected. The number of vector differences used in the mutation is indicated next, and'bin' indicates the way the old vector and the trial vector are recombined. A number of other DE strategy variants also exists [17, 18, 68, 73, 74, 84].

An empirical comparison study between different DE strategies with a set of single-objective optimization problems has been conducted in 2006 [64]. It was concluded that DE/best/1/bin generally performed best for the problem set but based

[3] Preferring the trial vector in the case of equal objective values has importance if the objective landscape contains a plateau; preferring the old vector would cause the search to stagnate on the plateau.

$$\text{Input}: D, G_{max}, NP \geq 4, F \in (0, 1+], CR \in [0,1], \text{and initial bounds: } \mathbf{x}^{(lo)}, \mathbf{x}^{(hi)}$$

$$\text{Initialize}: \begin{cases} \forall i \leq NP \wedge \forall j \leq D : x_{j,i,0} = x_j^{(lo)} + rand_j[0,1] \cdot \left(x_j^{(hi)} - x_j^{(lo)}\right), \\ i = \{1, 2, \ldots, NP\}, j = \{1, 2, \ldots, D\}, G = 0, rand_j[0,1] \in [0,1] \end{cases}$$

$$\begin{cases} \text{While } G < G_{max} \\ \forall i \leq NP \begin{cases} \text{Mutate and recombine:} \\ r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}, \text{randomly selected,} \\ \quad \text{except mutually different and different from } i \\ j_{rand} \in \{1, 2, \ldots, D\}, \text{randomly selected for each } i \\ \forall j \leq D, u_{j,i,G} = \begin{cases} x_{j,r_3,G} + F \cdot \left(x_{j,r_1,G} - x_{j,r_2,G}\right) \\ \quad \text{if} \quad rand_j[0,1) < CR \vee j == j_{rand} \\ x_{j,i,G} \quad \text{otherwise} \end{cases} \\ \text{Select}: \\ \mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G} \text{ if } f(\mathbf{u}_{i,G}) \leq f(\mathbf{x}_{i,G}) \\ \mathbf{x}_{i,G} \text{ otherwise} \end{cases} \end{cases} \\ G = G + 1 \end{cases}$$

Fig. 1 The basic DE algorithm

on the result, also DE/rand/1/bin performed well. In general, performance difference between the two above mentioned strategies is that DE/best/1/bin is greedier and faster but DE/rand/1/bin is more reliable and therefore performs better with harder problems [74, pp. 154–156].

The stagnation possibility of the DE/rand/1/bin strategy has been discussed in [58]. It is possible that the search stagnates or premature convergence occurs before reaching the global optimum. These two cases can be distinguished by observing the diversity of the population (diversity is lost in the case of premature convergence). Probability of stagnation or premature convergence can be reduced by increasing the size of the population and/or F. The search can be also repeated several times to increase confidence.

2.4 Differential Evolution for Multiple Objectives and with Constraints

Several extensions of DE for multi-objective optimization have been proposed. As mentioned earlier, first proposals converted a MOOP into a single-objective form (e.g., [5, 11, 89]). Later proposals are mainly based on Pareto-dominance. In the following, methods are listed in chronological order. More detailed review of most of the approaches can be found in [17, pp. 596–604] and [63]. In many cases, new proposals are slight modifications of earlier methods.

The first method extending DE for multi-objective optimization using the Pareto approach was the Pareto-based DE approach in 1999 [12]. Pareto DE [8] was also mentioned around the same time, unfortunately without an explicit description of the method. After these in 2001–2002, the Pareto(-frontier) DE (PDE) algorithm [2, 3], the first version of GDE [55], Self-adaptive PDE (SPDE) [1], and the Pareto DE Approach (PDEA) [61] were introduced. Next in 2003–2004, Adaptive Pareto DE (APDE) [92], Multi-Objective DE (MODE) [90], Vector Evaluated DE (VEDE) [69], the second version of GDE [45], and Non-dominated Sorting DE (NSDE) [31] were proposed. In 2005, DE for Multiobjective Optimization (DEMO) [75], the third version of GDE [48], and ε-MyDE [79] were introduced. In 2006, DE for Multiobjective Optimization with Random Sets (DEMORS) [29], Multiobjective DE based Decomposition (MODE/D) [59], ε-constraint with Cultured DE (ε-CCDE) [7] were published. Next in 2007–2012, the DE algorithm based on ε-dominance and an orthogonal design method (ε-ODEMO) [9], Opposition-based Multi-Objective DE (OMODE) [70], Cluster-Forming DE (CFDE) [36], DE with local dominance and a scalar selection mechanism (MODE-LD+SS) [66], Adaptive Multi-objective DE with Stochastic Coding Strategy (AS-MODE) [96], and Multi-Objective DE Algorithm (MODEA) [4] were published. Some of the latest proposals are Integrated Multi-Objective DE (I-MODE) [81], DE with Pareto Tournaments (DEPT) [13], Opposition-based Self-adaptive Hybridized DE Algorithm for Multi-objective Optimization (OSADE) [16], and Variable-Size Multi-Objective DE (VSMODE) [14]. Not all the later proposals are based on Pareto dominance, e.g., MODE/D and ε-CCDE convert a multi-objective problem to a single-objective form for solving.

In addition to new algorithm proposals, there exist also some other relevant studies. One study is about incorporating directional information in the selection of vectors for the mutation step of DE [32]. Comparison between GA and DE in multi-objective optimization has been done in [88] and it has been concluded that DE explores the decision variable space better than GA. A comparison between four different multi-objective DE variants is presented in [97]. The variants differ in balancing between convergence and diversity. Based on experimental results it is found that the balancing technique that is used, e.g., in DEMO and the third version of GDE is better than the one used, e.g., in PDEA. This same observation has been noted also in [86].

Besides solving problems with multiple objectives, DE has also been modified for handling problems with constraints [10, 57, 60, 82]. First approaches were based on applying penalty functions, which has the problem of selecting penalty parameters as noted earlier. To overcome this problem, the selection rules given in [19, pp. 131–132] (cf. Sect. 2.1) have been used extensively later on [62].

3 Generalized Differential Evolution

The leading goal of GDE has been to keep changes as little as possible and to avoid unnecessary complexity. The key idea and justification for the name is that the extension falls back to basic DE in the case of an unconstrained single-objective problem. This property is contrary to all the other multi-objective DE approaches mentioned in Sect. 2.4.

GDE has been using the classic DE described in Sect. 2.3. This was chosen for GDE because of its simplicity and good observed performance [76, 77]. This strategy is also the most used DE strategy in the literature [17, p. 594]. However, some other strategy or recent modification as described in [18, 68, 80] could be used instead.

Several GDE versions exist and they differ in the way multi-objective optimization is performed—more precisely, how diversity of solutions is maintained during the search. In the following, different versions of GDE are described. Performance is demonstrated here only for the last version of GDE. More results and numerical comparisons between GDE versions can be found in [38].

3.1 First Version, GDE1

The first version, GDE1, extends the basic DE algorithm for constrained multi-objective optimization by just modifying the selection operation of DE. In GDE1, the selection operation is based on constrain-domination (cf. Sect. 2.1) and can be simply defined as:

$$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G} \text{ if } & \mathbf{u}_{i,G} \preceq_c \mathbf{x}_{i,G} \\ \mathbf{x}_{i,G} \text{ otherwise.} \end{cases} \tag{6}$$

The weak constrain-domination relation is used to maintain congruity with the selection operation of DE. Thus, in the case of equality, the trial vector is preferred. One should note that the selection is fully elitist in the sense of Pareto-dominance, i.e., the best solutions cannot be lost during the search.

GDE1 does not have any kind of diversity preservation, which is rare compared to present MOEAs. Nevertheless, GDE1 has been able to provide surprisingly good results with the some problems in [43, 46] but has been found to be rather sensitive to the selection of the control parameter values as noted in [47].

3.2 Second Version, GDE2

The second version, GDE2, introduced a diversity preservation operation to GDE in [45]. Again, only the selection operation of basic DE was modified. The selection is done based on crowding in the objective space when the trial and old vector are feasible and non-dominating. More formally, the selection operation is now:

$$\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G} \text{ if } \begin{cases} \mathbf{u}_{i,G} \preceq_c \mathbf{x}_{i,G} \\ \vee \\ \begin{cases} \forall j \in \{1, \ldots, K\} : g_j(\mathbf{u}_{i,G}) \leq 0 \\ \wedge \\ \mathbf{x}_{i,G} \not\prec \mathbf{u}_{i,G} \\ \wedge \\ d_{\mathbf{u}_{i,G}} \geq d_{\mathbf{x}_{i,G}} \end{cases} \end{cases} \\ \mathbf{x}_{i,G} \text{ otherwise} \end{cases}, \quad (7)$$

where d_i measures the distance of a particular solution i to its neighbor solutions. Implementation was done by using the crowding distance of NSGA-II. However, as noted in [45], any other distance measure could be used instead of the crowding distance.

Since non-dominated sorting is not used, crowding is measured among the whole population. This improves the extent and distribution of the obtained set of solutions but slows down the convergence of the overall population because it favors isolated solutions far from the Pareto front until all the solutions have converged near the Pareto front. Also, GDE2 has been noted to be rather sensitive to the selection of the control parameter values [45].

3.3 Third Version, GDE3

The third version, GDE3, was published in [41, 48]. Besides selection, another part of basic DE was also modified. Now, in the case of comparing feasible and non-dominating solutions, both vectors are saved. Therefore, at the end of a generation, the size of the population may be larger than it originally was. If this is the case, the population is then decreased back to the original size based on a similar selection approach as used in NSGA-II. The worst population members according to non-dominance and crowding are removed to decrease the size of the population to the original size. Non-dominance is the primary sorting criterion and crowding is the secondary sorting criterion as in NSGA-II.

Non-dominated sorting was modified to also take constraints into consideration following principles of constrain-domination. The selection based on the crowding distance was improved over the original method of NSGA-II to provide a better distributed set of vectors. This improvement is described in the following section.

The whole GDE3 is presented in Fig. 2. Parts that are new compared to previous GDE versions are framed in Fig. 2. Without these parts, the algorithm is identical to GDE1.

In NSGA-II and PDEA, the size of the population after a generation is 2*NP*, which is then decreased to *NP*. In GDE3 and DEMO, the size of the population after a generation is between *NP* and 2*NP* because the size of the population is increased only if the trial vector and the old vector are feasible and non-dominated. This will reduce the computational cost of the whole algorithm. DEMO and GDE3 emphasize

$$
\begin{array}{l}
\text{Input}:D, G_{max}, NP \geq 4, F \in (0, 1+], CR \in [0,1], \text{and initial bounds: } \mathbf{x}^{(lo)}, \mathbf{x}^{(hi)} \\
\text{Initialize}: \begin{cases} \forall i \leq NP \wedge \forall j \leq D : x_{j,i,0} = x_j^{(lo)} + rand_j[0,1] \cdot \left(x_j^{(hi)} - x_j^{(lo)}\right) \\ i = \{1,2,\ldots,NP\}, j = \{1,2,\ldots,D\}, G = 0, n = 0, rand_j[0,1] \in [0,1], \end{cases}
\end{array}
$$

$$
\begin{cases}
\text{While } G < G_{max} \\
\forall i \leq NP \begin{cases}
\text{Mutate and recombine:} \\
r_1, r_2, r_3 \in \{1,2,\ldots,NP\}, \text{randomly selected,} \\
\qquad \text{except mutually different and different from } i \\
j_{rand} \in \{1,2,\ldots,D\}, \text{randomly selected for each } i \\
\forall j \leq D, u_{j,i,G} = \begin{cases} x_{j,r_3,G} + F \cdot \left(x_{j,r_1,G} - x_{j,r_2,G}\right) \\ \qquad \text{if} \quad rand_j[0,1) < CR \vee j == j_{rand} \\ x_{j,i,G} \quad \text{otherwise} \end{cases} \\
\text{Select}: \\
\mathbf{x}_{i,G+1} = \begin{cases} \mathbf{u}_{i,G} \text{ if } \quad \mathbf{u}_{i,G} \preceq_c \mathbf{x}_{i,G} \\ \mathbf{x}_{i,G} \text{ otherwise} \end{cases} \\
\boxed{\begin{array}{l} \text{Set}: \\ \begin{array}{l} n = n + 1 \\ \mathbf{x}_{NP+n,G+1} = \mathbf{u}_{i,G} \end{array} \quad \text{if} \begin{cases} \forall j : g_j(\mathbf{u}_{i,G}) \leq 0 \\ \wedge \\ \mathbf{x}_{i,G+1} == \mathbf{x}_{i,G} \\ \wedge \\ \mathbf{x}_{i,G} \not\preceq \mathbf{u}_{i,G} \end{cases} \end{array}}
\end{cases} \\
\boxed{\begin{cases}
\text{While } n > 0 \\
\quad \text{Select } \mathbf{x} \in \mathscr{P} = \{\mathbf{x}_{1,G+1}, \mathbf{x}_{2,G+1}, \ldots, \mathbf{x}_{NP+n,G+1}\}: \\
\qquad \begin{cases} \mathbf{x} \text{ belongs to the last non-dominated set of } \mathscr{P} \\ \wedge \\ \mathbf{x} \text{ is the most crowded in the last non-dominated set} \end{cases} \\
\quad \text{Remove } \mathbf{x} \text{ from } \mathscr{P} \\
\quad n = n - 1
\end{cases}} \\
G = G + 1
\end{cases}
$$

Fig. 2 The GDE3 algorithm

convergence over diversity a bit more compared to NSGA-II and PDEA as noted in [97].

Decreasing the size of the population at the end of a generation is the most complex operation in the algorithm. The reduction is done by using non-dominated sorting and pruning based on crowding. The run-time complexity of non-dominated sorting is $O\left(NP \log^{M-1} NP\right)$ [33]. The pruning of population members based on crowding is also a complex operation in general. When the operation is performed using the crowding distance, it can be performed in time $O(MNP \log NP)$ [41]. Therefore overall run-time complexity of GDE3 is $O\left(G_{max}NP \log^{M-1} NP\right)$.

Compared to the earlier GDE versions, GDE3 improves the ability to handle MOOPs by giving a better distributed set of solutions and being less sensitive to the selection of control parameter values. GDE3 has been compared to NSGA-II and has been found to be at least comparable based on experimental results in [48].

3.3.1 Diversity Preservation for Bi-objective Problems

The first diversity preservation technique used in GDE3 was an improved version of the approach used in NSGA-II. In NSGA-II, the crowding distance values are calculated once for all the members of a non-dominated set. Then members having the smallest crowding distance values are removed without taking into account that the removal of a member will affect the crowding distance value of its neighbors. The outcome is that the diversity of the remaining members after removal is non-optimal.

The diversity preservation operation in GDE3 removes the most crowded members from a non-dominated set one by one and updates the crowding distance value of the remaining members after each removal. A straightforward approach would have time complexity class $O\left(MNP^2\right)$ but a more sophisticated algorithm exists and it has time complexity class $O(MNP \log NP)$, which is the same as for the approach used with NSGA-II. This approach is described in [41] and was implemented when GDE3 was originally introduced but published later because the detailed description of the diversity preservation technique did not fit into [48].

In [41] it has been shown that the crowding distance does not estimate crowding properly when the number of objectives is more than two. This is a significant observation since NSGA-II is the most popular MOEA and the crowding distance has been used in many studies even during recent years. The crowding distance has also been used in cases with more than two objectives, e.g., in [4, 71]. It should be mentioned that several multi-objective DE approaches mentioned in Sect. 2.4 use the crowding distance and therefore do not provide good diversity when the number of objectives is more than two.

3.3.2 Diversity Preservation for Many-Objective Problems

Based on observations in [41], a new efficient diversity preservation technique was needed for many-objective problems. The term many-objective is used in the MOEA literature when the number of objectives is more than three. In this chapter, many-objective refers to a situation when the number of objectives is three or more.

A pruning method intended to be both effective and relatively fast was proposed in [40]. The basic idea of the method is to eliminate the most crowded members of a non-dominated set one by one, and update the crowding information of the remaining members after each removal. The crowding estimation is based on distances to the nearest neighbors of solution candidates in the objective space and an efficient search method to find the nearest neighbors.

The diversity preservation technique used in GDE3 was replaced with the diversity preservation technique intended for a large number of objectives presented in [40]. Although the last published version number for GDE is 3, the version having diversity maintenance technique for many-objective problems, can be considered as a version 3.1.

The final populations for several ZDT and DTLZ problems [23, 98] solved using GDE3 with the diversity maintenance technique for many-objective optimization are shown in Figs. 3 and 4. Good results can be observed for all the problems. With ZDT problems 250 generations and a population size of 100 were used. The DTLZ problems were solved using 250 generations and population size 200. It was justified to use a larger population size than with the ZDT problems to approximate the Pareto front since the objective space has a higher dimensionality. The control parameter values $CR = 0.2$ and $F = 0.2$ were used with the problems with the exception that values $CR = 0.0$ and $F = 0.5$ were used with ZDT4.[4] Small CR values were used for the problems since they are separable (cf. Sect. 2.3). Another reason for small CR values was that the objectives of the ZDT problems, especially ZDT4, are in different order of difficulty. This means that solving different individual objectives needs a different computational effort. Using a large CR value would lead to a premature convergence along the first objective far before the second objective converges as noted in [44, 47, 50].

GDE3 with the diversity preservation technique for many-objective optimization was one of the participants in a multi-objective optimization competition arranged at the 2007 IEEE Congress on Evolutionary Computation (CEC 2007). The task was to solve a set of multi-objective problems having from two to five objectives defined in [30]. Based on the results reported in [51], GDE3 with the described diversity preservation technique received a winning entry nomination in the competition. Two years later the same method participated another multi-objective optimization competition arranged at the 2009 IEEE Congress on Evolutionary Computation (CEC 2009) [53] and this time the method was ranked among the top five best-performing algorithms.

GDE3 has been also implemented into publicly available object-oriented framework for multi-objective optimization [25] and it has been used in several comparative studies, e.g., in [22, 95, 96].

3.4 Studies of Control Parameter Values for GDE

The effect of control parameters CR and F was studied with GDE1 and GDE3 in [47, 50], respectively. Different control parameter values were tested using bi-objective

[4]ZDT4 has multiple equally spaced local Pareto fronts and $F = 0.5$ advances moving from one local front to another [47, 50].

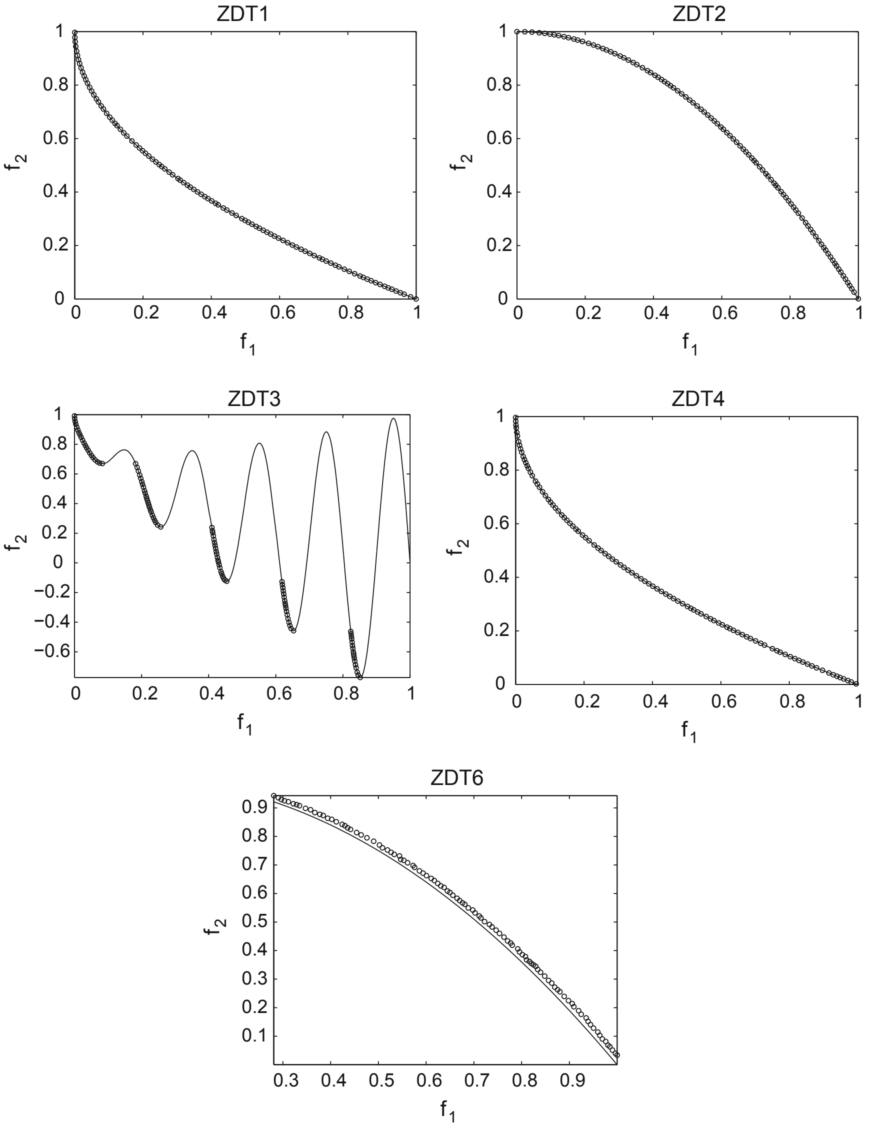

Fig. 3 Results for ZDT problems using GDE3 with the diversity maintenance technique for many-objective optimization

test problems and performance metrics described in [19, pp. 326–328, 338–360]. The experiments were restricted for the two objective problems due to space limitation in the articles. Similar experiments were repeated with the GDE version 3.1 and the DTLZ test problems varying the number of objectives from two to five in [38].

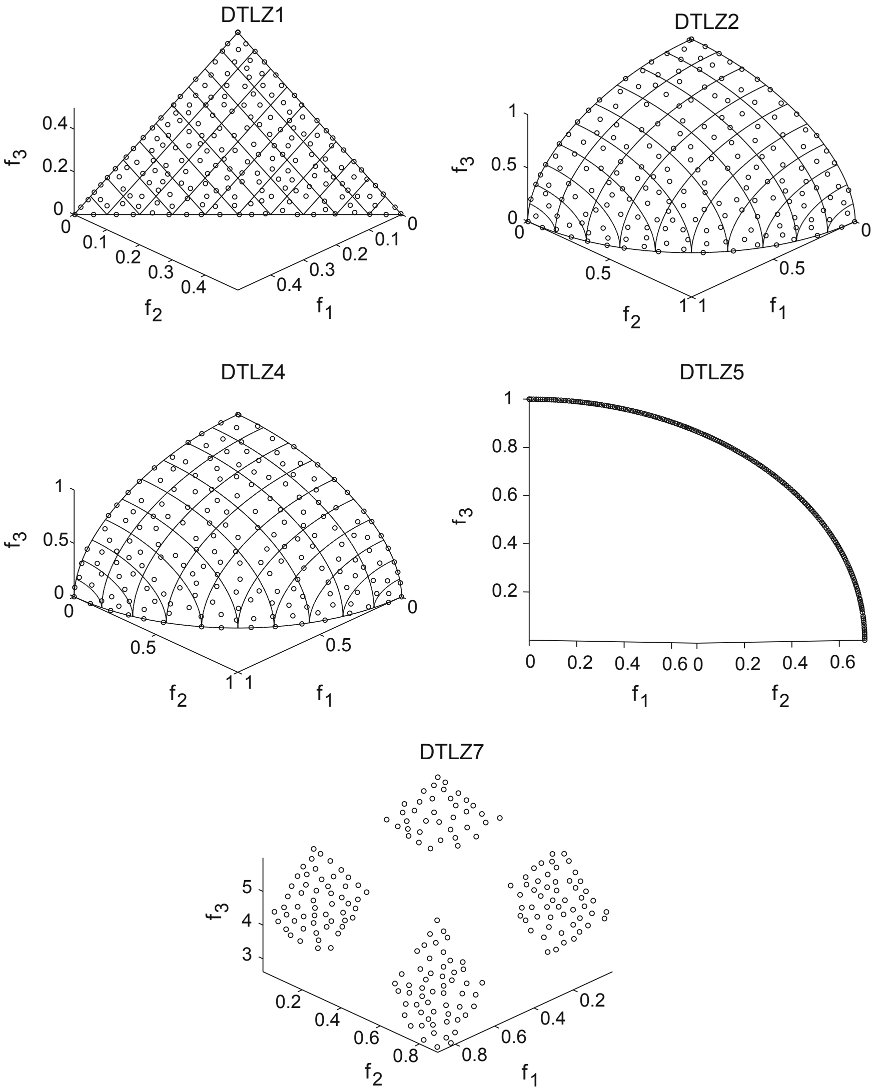

Fig. 4 Results for DTLZ problems using GDE3 with the diversity maintenance technique for many-objective optimization

According to the diversity and cardinality measures in [47, 50], GDE3 provides a better distribution with a larger number of non-dominated solutions than GDE1 in general. The results (especially the number of non-dominated solutions) of GDE3 is also less sensitive to the variation of the control parameter values compared to GDE1, i.e., GDE3 appears to be more robust in terms of the control parameter values selection.

Based on the empirical results in [38, 47, 50], suitable control parameter value ranges for multi-objective optimization are the same as for single-objective optimization, i.e., $CR \in [0, 1]$ and $F \in (0, 1+]$. As noted in [47, 50], if the complexity of objectives differs greatly (i.e., the objectives demand considerably different computational effort if solved individually), it is better to use a smaller CR value to prevent the population from converging to a single point of the Pareto front.

From the results in [38, 47, 50], an inverse relationship between the values of CR and F was observed, i.e., a larger F value can be used with a small CR value than with a large CR value, and this relationship is nonlinear. An explanation for this was found from theoretical analysis of the single-objective DE algorithm. A formula for the relationship between the control parameters of DE and the evolution of the population variance / standard deviation has been presented in [91]. The change of the population standard deviation between successive generations due to the crossover and mutation operations is denoted with c and its value is calculated as:

$$c = \sqrt{2F^2CR - 2CR/NP + CR^2/NP + 1}\,. \tag{8}$$

When $c < 1$, the crossover and mutation operations decrease the population's standard deviation. When $c = 1$, the standard deviation does not change, and when $c > 1$, the standard deviation increases. Since the selection operation of an EA usually decreases the population standard deviation, $c > 1$ is recommended in order to prevent premature convergence. On the other hand, if c is too large, the search process proceeds reliably, but too slowly. In [38, 47, 50], it has been observed that $c = 1.5$ is a suitable upper limit. This limit has been noticed also with single-objective problems [91]. When the size of the population is relatively large (e.g., $NP > 50$), the value of c depends mainly on the values of CR and F.

Since the selection of the control parameters for unknown problems still cause difficulty, an automated control parameter adaptation approach for CR and F has been studied in [39] and found to increase robustness so that a user can incorporate this adaptation method instead of selecting fixed control parameter values. A selection rule for NP has also been given in [39]. Several other parameter control mechanisms have been compared in [24]. A good control parameter control mechanism with an automated stopping criterion would release users from selecting any parameters.

3.5 Constrained Optimization with GDE

The GDE versions include in their definition also a constraint handling approach, which is identical in all the versions. This constraint handling approach was first introduced and evaluated for single-objective optimization with DE in [56] and later extended into multi-objective optimization with GDE.

In [46], a small set of mechanical design problems including several constraints was solved using GDE1. GDE1 has been also used to solve a given set of constrained single-objective optimization problems in the CEC 2006 Special Session on Constrained Real-Parameter Optimization [49]. GDE1 was able to solve almost all the problems in a given maximum number of solution candidate evaluations. A better solution than previously known was found for some problems. It was also demonstrated that GDE actually needs a lower number of function evaluations than required if all the constraints are to be evaluated (as it is in the case of several other constraint handling techniques).

In [48], the ability of GDE versions to handle several constraints and different types of decision variables has been demonstrated using a bi-objective spring design problem. GDE versions use real-coded variables, which are converted into corresponding actual variable types before evaluation of the solution candidate.

The GDE versions have been successfully applied also for more challenging constrained multi-objective optimization problems such as scaling filter design [54], multi-objective scheduling for NASA's space science missions [34, 35], balanced surface acoustic wave and microwave filters design [27, 85], Yagi-Uda antenna design [28], the software project scheduling problem [15], and the molecular sequence alignment problem [42]. The last problem is a nonlinear problem with thousands of integer decision variables. Such large problems have rarely been successfully solved with an EA.

4 Future Directions

Many real world problems have computationally expensive objectives and constraints. These have been problematic for EAs since they generally require a large number of function evaluations. One possible remedy is parallelization of the algorithm. GDE, like EAs in general, can be easily parallelized. Another approach for computationally expensive functions is to use approximations of functions, meta-models, during most of the search and evaluate the actual functions only when really required. These modifications to GDE are possible when GDE is applied to practical problems. Also, the basic DE can be modified as has been done in several approaches described in [18, 68].

GDE as DE are best suited for real-parameter optimization but also for the cases when the parameters are of different types since they can be converted easily to real-parameters. In the case of combinatorial optimization, some other methods have been considered to be more suitable. However, there are studies extending DE also to other domains to be applicable also for combinatorial and discrete optimization [67, 72, 87, 93].

Further investigation of the automated control parameter adaptation is still needed in order to increase the usability of GDE. The ideal situation would be to have all the parameters automated to free the user from their selection.

5 Conclusions

The development history of Generalized Differential Evolution (GDE) has been described with a brief review of other multi-objective approaches based on Differential Evolution (DE). GDE is a real-coded general purpose EA extended from DE to handle multiple objectives and constraints. Each GDE version falls back to DE in the case of an unconstrained single-objective problem. DE was chosen as a basic search "engine" because it is an effective and widely applicable evolutionary algorithm characterized with simplicity, linear scalability, and ability to perform a rotationally invariant search.

The first version, GDE1, extends DE for constrained multi-objective optimization by modifying the selection rule of basic DE. The basic idea in the selection rule is that the trial vector is selected to replace the old vector in the next generation if the trial vector weakly constrain-dominates the old vector. There is neither explicit non-dominated sorting during the optimization process, nor an extra repository for non-dominated vectors, nor any mechanism for preserving diversity. GDE1 has been observed to perform well with some problems but found rather sensitive to the selection of the control parameter values. Also, the diversity of the obtained solutions could have been better.

The second version, GDE2, makes a selection between the old and the trial vector based on crowding in the objective function space when the vectors are feasible and not dominating each other in the objective function space. This improves the extent and distribution of an obtained set of solutions but slows down the convergence of the population because it favors isolated solutions far from the Pareto front until all the solutions have converged near the Pareto front. This GDE version, too, has been observed to be rather sensitive to the selection of the control parameter values.

The third version is GDE3. In addition to the selection operation change, a further modification to basic DE is population reduction at the end of each generation, if the size of the population has grown during the generation. In the case of being feasible and non-dominated, both the old and the trial vectors are saved for the population of the next generation. At the end of each generation, the size of the population is reduced using non-dominated sorting and pruning based on crowding estimation. GDE3 provides better distribution of solutions than the earlier GDE versions and it is also more robust in terms of the selection of the control parameter values.

The diversity preservation technique of GDE3 is an improved version of the technique in NSGA-II based on the crowding distance. The technique has been noticed to provide a good diversity in the case of two objectives but the diversity deteriorates with a larger number of objectives because the crowding distance metric does not estimate crowding well when the number of objectives is more than two. This observation is noteworthy because NSGA-II is the most used MOEA, several multi-objective DE variants apply crowding distance, and because the crowding distance metric has subsequently been used in several studies with more than two objectives.

Because of the defect in the crowding distance metric, GDE3 has been further developed with the diversity preservation technique designed for many-objective problems. This technique provides a good diversity also in the cases of more than two objectives and is relatively fast, especially with a low number of objectives. The time needed by the pruning technique increases when the number of objectives but is substantially less compared to the other effective approaches in MOEAs. GDE with this diversity preservation technique can be considered as version 3.1.

The influence of the control parameters has been studied and discussed with respect to GDE1 and GDE3. Multi-objective optimization is fundamentally different compared to single-objective optimization since the population is not expected to converge to a single point. It was found that GDE3 is more robust with respect to control parameter values and provides a better diversity than GDE1. It appears that suitable control parameter ranges for multi-objective optimization are the same as for single-objective optimization, i.e., $CR \in [0, 1]$ and $F \in (0, 1+]$. However, it is better to use a smaller CR value to prevent premature convergence of one objective if the difficulty of objectives differ, i.e., different objectives demand considerably different computational effort if solved one at time.

The nonlinear relationship between CR and F was observed following the theory of basic single-objective DE concerning the relationship between the control parameters and the evolution of the population's variance / standard deviation. Based on this observation, it is advisable to select the values for CR and F satisfying the condition $1.0 < c < 1.5$, where c denotes the change of the population's standard deviation between successive generations due to the variation (crossover and mutation) operators.

The GDE versions have been used in a number of problems having different number of objectives and constraints. GDE3 with the diversity preservation technique for many-objective optimization has been able to solve successfully some difficult problems involving up to five objectives and has performed well compared to several other MOEAs.

Currently, GDE is a potentially general purpose optimizer for nonlinear optimization with constraints and objectives. However, some limitations in GDE exist: GDE is not applicable for optimization having large number of objectives (e.g. over 20) because selection based on Pareto-dominance does not function well then anymore.

Finally, it can be concluded that GDE3 with the diversity preservation technique for many-objective problems is a good choice for global nonlinear optimization with different types of decision variables, constraints, and a few (e.g., one to five) objectives.

Acknowledgments The first author would like to acknowledge support of the South Savo Regional Fund of the Finnish Cultural Foundation (Suomen Kulttuurirahaston Etelä-Savon rahasto) and support of the Mexican Government thought the Foreign Ministry (Gobierno de México, a través de la Secretería de Relaciones Exteriores). The second author gratefully acknowledges support from CONACyT project no. 221551.

References

1. Abbass, H.A.: The self-adaptive Pareto differential evolution algorithm. In: Yao, X. (ed.) Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002), Honolulu, HI, USA, pp. 831–836. IEEE Service Center, Piscataway (2002)
2. Abbass, H.A., Sarker, R.: The Pareto differential evolution algorithm. Int. J. Artif. Intell. Tools **11**(4), 531–552 (2002)
3. Abbass H.A., Sarker R., Newton, C.: PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems. In: Proceedings of the 2001 Congress on Evolutionary Computation (CEC 2001), Seoul, South Korea, pp. 971–978. IEEE Service Center, Piscataway (2001)
4. Ali, M., Siarry, P., Pant, M.: An efficient differential evolution based algorithm for solving multi-objective optimization problems. Eur. J. Oper. Res. **217**(2), 404–416 (2012)
5. Babu, B.V., Jehan, M.M.L.: Differential Evolution for multi-objective optimization. In: Sarker, R., Reynolds, R., Abbass, H., Tan, K.C., McKay, B., Essam, D., Gedeon, T. (eds.) Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003), Canberra, Australia, pp. 2696–2703. IEEE Press, Piscataway (2003)
6. Bäck, T.: Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York (1996)
7. Becerra, R.L., Coello Coello, C.A.: Solving hard multiobjective optimization problems using ε-constraint with cultured differential evolution. In: Runarsson, T.P., Beyer, H.-G., Burke, E., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.), Proceedings of the 9th International Conference on Parallel Problem Solving from Nature (PPSN IX), vol. 4193, pp. 543–552. Lecture Notes in Computer Science (LNCS), Reykjavik, Iceland. Springer, Berlin (2006)
8. Bergey, P.K.: An agent enhanced intelligent spreadsheet solver for multi-criteria decision making. In: Haseman, W.D., Nazareth, D.L. (eds.) Proceedings of the Fifth Americas Conference on Information Systems (AMCIS 1999), Milwaukee, Wisconsin, USA, pp. 966–968. Association for Information Systems, Atlanta (1999)
9. Cai, Z., Gong, W., Huang, Y.: A novel differential evolution algorithm based on ε-domination and orthogonal design method for multiobjective optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007), Matsushima, Japan, pp. 286–301. Springer, Berlin (2007)
10. Chang, T.-T., Chang, H.-C.: Application of differential evolution to passive shunt harmonic filter planning. In: 8th International Conference on Harmonics and Quality of Power, Athens, Greece, pp. 149–153. IEEE Press (1998)
11. Chang, C.S., Xu, D.Y.: Differential Evolution based tuning of fuzzy automatic train operation for mass rapid transit system. IEE Proc. Electr. Power Appl. **147**(3), 206–212 (2000)
12. Chang, C.S., Xu, D.Y., Quek, H.B.: Pareto-optimal set based multiobjective tuning of fuzzy automatic train operation for mass transit system. IEE Proc. Electr. Power Appl. **146**(5), 577–583 (1999)
13. Chaves-González, J.M., Vega-Rodríguez, M.A.: DNA strand generation for DNA computing by using a multi-objective differential evolution algorithm. Biosystems **116**, 49–64 (2014)
14. Chen, Y., Mahalec, V., Chen, Y., Liu, X., He, R., Sun, K.: Reconfiguration of satellite orbit for cooperative observation using variable-size multi-objective differential evolution. Eur. J. Oper. Res. **242**(1), 10–20 (2015)
15. Chicano, F., Luna, F., Nebro, A.J., Alba, E.: Using multi-objective metaheuristics to solve the software project scheduling problem. In: Krasnogor, N., Lanzi, P.L. (eds.) Proceedings of the 13th Annual Conference on Genetic and evolutionary computation (GECCO 2011), Dublin, Ireland, pp. 1915–1922. ACM (2011)
16. Chong, J.K., Tan, K.C.: Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Volume 1, Chapter An Opposition-based Self-adaptive Hybridized Differential Evolution Algorithm for Multi-objective Optimization (OSADE), pp. 447–461. Springer International Publishing, Cham (2015)

17. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Springer Science+Business Media. Springer, Berlin (2007)
18. Das, S., Suganthan, P.N.: Differential evolution: a survey of the state-of-the-art. IEEE Trans. Evolut. Comput. **15**(1), 4–31 (2011)
19. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, Chichester (2001)
20. Deb, K., Jain, H.: Algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. IEEE Trans. Evolut. Comput. **18**(4), 577–601 (2014)
21. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evolut. Comput. **6**(2), 182–197 (2002)
22. Deb, K., Sinha, A., Kukkonen, S.: Multi-objective test problems, linkages, and evolutionary methodologies. In: Keijzer, M., et al. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006), pp. 1141–1148. ACM Press, Seattle (2006)
23. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Abraham, A., Jain, L., Goldberg, R. (eds.) Evolutionary Multiobjective Optimization, pp. 105–145. Springer, London (2005)
24. Drozdik, M., Aguirre, H., Akimoto, Y., Tanaka, K.: Comparison of parameter control mechanisms in multi-objective Differential Evolution. In: Dhaenens, C., Jourdan, L., Marmion, M.-E. (eds.) Proceedings of the 9th International Conference in Learning and Intelligent Optimization (LION 9). Lecture Notes in Computer Science, vol. 8994, pp. 89–103. Springer, Lille (2015)
25. Durillo, J.J., Nebro, A.J.: jMetal: A Java framework for multi-objective optimization. Adv. Eng. Softw. **42**(10), 760–771 (2011)
26. Emmerich, M., Beume, N., Naujoks, B.: An EMO algorithm using the hypervolume measure as selection criterion. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005), pp. 62–76. Springer, Guanajuato (2005)
27. Goudos, S.K., Sahalos, J.N.: Pareto optimal microwave filter design using multiobjective differential evolution. IEEE Trans. Antennas Propag. **58**(1), 132–144 (2010)
28. Goudos, S.K., Siakavara, K., Vafiadis, E., Sahalos, J.N.: Pareto optimal Yagi-Uda antenna design using multi-objective differential evolution. Prog. Electromagn. Res. **10**, 231–251 (2010)
29. Hernández-Diaz, A.G., Santana-Quintero, L.V., Coello Coello, C., Caballero, R., Molina, J.: A new proposal for multi-objective optimization using differential evolution and rough sets theory. In: Keijzer, M., et al. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006), pp. 675–682. ACM Press, Seattle (2006)
30. Huang, V.L., Qin, A.K., Deb, K., Zitzler, E., Suganthan, P.N., Liang, J.J., Preuss, M., Huband, S.: Problem definitions for performance assessment on multi-objective optimization algorithms. Technical report, School of EEE, pp. 639–798. Nanyang Technological University, Singapore (2007)
31. Iorio, A., Li, X.: Solving rotated multi-objective optimization problems using Differential Evolution. In: Webb, G.I., Yu, X. (eds.) Proceedings of the 17th Australian Joint Conference on Artificial Intelligence (AI 2004). Lecture Notes in Computer Science (LNCS), vol. 3339, pp. 861–872. Springer, Cairns (2004)
32. Iorio, A.W., Li, X.: Incorporating directional information within a Differential Evolution for multi-objective optimization. In: Keijzer, M., et al. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006), pp. 691–697. ACM Press, Seattle (2006)
33. Jensen, M.T.: Reducing the run-time complexity of multiobjective EAs: the NSGA-II and other algorithms. IEEE Trans. Evolut. Comput. **7**(5), 503–515 (2003). Oct
34. Johnston, M.D.: Multi-objective scheduling for NASA's future deep space network array. In: Proceedings of the 5th International Workshop on Planning and Scheduling for Space (IWPSS 2006), Baltimore, MD, USA, pp. 27–35. Space Telescope Science Institute (STScI) (2006)
35. Johnston, M.D., Giuliano, M.E.: Multi-objective scheduling for space science missions. J. Adv. Comput. Intell. Intell. Inf. **15**(8), 1140–1148 (2011)

36. Justesen, P.D., Ursem, R.K.: Multiobjective distinct candidates optimization (MODCO): a cluster-forming differential evolution algorithm. In: Ehrgott, M., Fonseca, C.M., Gandibleux, X., Hao, J.-K., Sevaux, M. (eds.) Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2009). Lecture Notes in Computer Science (LNCS), vol. 5467, pp. 525–539. Nantes, France (2009)
37. Knowles, J.D.: Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization. Ph.D. thesis, The University of Reading, Reading (2002)
38. Kukkonen, S.: Generalized Differential Evolution for Global Multi-Objective Optimization with Constraints. Ph.D. thesis, Lappeenranta University of Technology, Acta Universitatis Lappeenrantaensis 475 (2012)
39. Kukkonen, S., Coello Coello, C.A.: Applying exponential weighting moving average control parameter adaptation technique with generalized differential evolution. In: Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC 2016), Vancouver, Canada (2016). Accepted for publication
40. Kukkonen, S., Deb, K.: A fast and effective method for pruning of non-dominated solutions in many-objective problems. In: Runarsson, T.P., Beyer, H.-G., Burke, E., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) Proceedings of the 9th International Conference on Parallel Problem Solving from Nature (PPSN IX), vol. 4193. Lecture Notes in Computer Science (LNCS), Reykjavik, Iceland, pp. 553–562. Springer, Berlin (2006)
41. Kukkonen, S., Deb, K.: Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems. In: Yen, G.G., Lucas, S.M., Fogel, G., Kendall, G., Salomon, R., Zhang, B.-T., Coello Coello, C.A., Runarsson, T.P. (eds.) Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006), pp. 3995–4002. IEEE Press, Vancouver (2006)
42. Kukkonen, S., Jangam, S.R., Chakraborti, N.: Solving the molecular sequence alignment problem with Generalized Differential Evolution 3 (GDE3). In: Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM 2007), Honolulu, HI, USA, pp. 302–309. IEEE (2007)
43. Kukkonen, S., Lampinen, J.: Comparison of generalized differential evolution algorithm to other multi-objective evolutionary algorithms. In: Proceedings of the 4th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004), Jyväskylä, Finland. 20 p. (2004)
44. Kukkonen, S., Lampinen, J.: A Differential Evolution algorithm for constrained multi-objective optimization: initial assessment. In: Hamza, M.H. (ed.) Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2004), pp. 96–102. ACTA Press, Innsbruck (2004)
45. Kukkonen, S., Lampinen, J.: An extension of generalized differential evolution for multi-objective optimization with constraints. In: Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII), vol. 3242. Lecture Notes in Computer Science (LNCS), pp. 752–761. Birmingham, England (2004)
46. Kukkonen, S., Lampinen, J.: Mechanical component design for multiple objectives using generalized differential evolution. In: Parmee, I.C. (ed.) Proceedings of the 6th International Conference on Adaptive Computing in Design and Manufacture (ACDM 2004), pp. 261–272. Springer, Bristol (2004)
47. Kukkonen, S., Lampinen, J.: An empirical study of control parameters for generalized differential evolution. In: Proceedings of the Sixth Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems (EUROGEN 2005), Munich, Germany. 12 p. (2005)
48. Kukkonen, S., Lampinen, J.: GDE3: the third evolution step of generalized differential evolution. In: Proceedings of the 2005 Congress on Evolutionary Computation (CEC 2005), pp. 443–450, Edinburgh, Scotland. IEEE Press (2005)
49. Kukkonen, S., Lampinen, J.: Constrained real-parameter optimization with Generalized Differential Evolution. In: Yen, G.G., Lucas, S.M., Fogel, G., Kendall, G., Salomon, R., Zhang, B.-T.,

Coello Coello, C.A., Runarsson, T.P. (eds.) Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006), pp. 911–918. IEEE Press, Vancouver (2006)
50. Kukkonen, S., Lampinen, J.: An empirical study of control parameters for the third version of generalized differential evolution (GDE3). In: Yen, G.G., Lucas, S.M., Fogel, G., Kendall, G., Salomon, R., Zhang, B.-T., Coello Coello, C.A., Runarsson, T.P. (eds.) Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006), pp. 7355–7362. IEEE Press, Vancouver (2006)
51. Kukkonen, S., Lampinen, J.: Performance assessment of generalized differential evolution 3 (GDE3) with a given set of problems. In: Srinivasan, D., Wang, L. (eds.) Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC 2007), pp. 3593–3600. IEEE, Singapore (2007)
52. Kukkonen, S., Lampinen, J.: Ranking-dominance and many-objective optimization. In: Srinivasan, D., Wang, L. (eds.) Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC 2007), pp. 3983–3990. IEEE, Singapore (2007)
53. Kukkonen, S., Lampinen, J.: Performance assessment of generalized differential evolution 3 with a given set of constrained multi-objective test problems. In: Proceedings of the 2009 IEEE Congress on Evolutionary Computation (CEC 2009), Trondheim, Norway, pp. 1943–1950 (2009)
54. Kukkonen, S., Sampo, J., Lampinen, J.: Applying generalized differential evolution for scaling filter design. In: Matoušek, R., Ošmera, P. (eds.) Proceedings of Mendel 2004. 10th International Conference on Soft Computing, pp. 28–33. Brno, Czech Republic (2004)
55. Lampinen, J.: DE's selection rule for multiobjective optimization. Technical report, Lappeenranta University of Technology, Department of Information Technology (2001)
56. Lampinen, J.: A constraint handling approach for the differential evolution algorithm. In: Yao, X. (ed.) Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002), Honolulu, HI, USA, pp. 1468–1473. IEEE Service Center, Piscataway (2002)
57. Lampinen, J., Zelinka, I.: Mechanical engineering design optimization by differential evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 128–146. McGraw-Hill, London (1999)
58. Lampinen, J., Zelinka, I.: On stagnation of the differential evolution algorithm. In: Proceedings of Mendel 2000. 6th International Conference on Soft Computing, pp. 76–83. Czech Republic (2000)
59. Li, H., Zhang, Q.: A multiobjective differential evolution based on decomposition for multiobjective optimization with variable linkages. In: Runarsson, T.P., Beyer, H.-G., Burke, E., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) Proceedings of the 9th International Conference on Parallel Problem Solving from Nature (PPSN IX), vol. 4193. Lecture Notes in Computer Science (LNCS), Reykjavik, Iceland, pp. 583–592. Springer, Berlin (2006)
60. Lin, Y.-C., Hwang, K.-S., Wang, F.-S.: Hybrid differential evolution with multiplier updating method for nonlinear constrained optimization problems. In: Yao, X. (ed.) Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002), Honolulu, HI, USA, pp. 872–877. IEEE Service Center, Piscataway (2002)
61. Madavan, N.K.: Multiobjective optimization using a Pareto differential evolution approach. In: Yao, X. (ed.) Proceedings of the 2002 Congress on Evolutionary Computation (CEC 2002), Honolulu, HI, USA, pp. 1145–1150. IEEE Service Center, Piscataway (2002)
62. Mezura-Montes, E., Miranda-Varela, M.E., del Carmen Gómez-Ramón, R.: Differential evolution in constrained numerical optimization: an empirical study. Inf. Sci. **180**(22), 4223–4262 (2010)
63. Mezura-Montes, E., Reyes-Sierra, M., Coello Coello, C.A.: Multi-objective optimization using differential evolution: a survey of the state-of-the-art. In: Chakraborty, U.K. (ed.) Advances in Differential Evolution, vol. 143, pp. 173–196. Studies in Computational Intelligence Series. Springer, Berlin (2008)
64. Mezura-Montes, E., Velázquez-Reyes, J., Coello Coello, C.A.: A comparative study of Differential Evolution variants for global optimization. In: Keijzer, M., et al. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006), pp. 485–492. ACM Press, USA (2006)

65. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston (1998)
66. Montaño, A.A., Coello Coello, C.A., Mezura-Montes, E.: MODE-LD+SS: a novel differential evolution algorithm incorporating local dominance and scalar selection mechanisms for multi-objective optimization. In *Proceedings of the 2010 Congress on Evolutionary Computation (CEC 2010)*, Barcelona, Spain, pp. 1–8. IEEE Press (2010)
67. Moraglio, A., Togelius, J., Silva, S.: Geometric differential evolution for combinatorial and programs spaces. Evolut. Comput. **21**(4), 591–624 (2013)
68. Neri, F., Tirronen, V.: Recent advances in differential evolution: a survey and experimental analysis. Artif. Intell. Rev. **33**(1), 61–106 (2010)
69. Parsopoulos, K.E., Tasoulis, D.K., Pavlidis, N.G., Plagianakos, V.P., Vrahatis, M.N.: Vector evaluated differential evolution for multiobjective optimization. In: Proceedings of the 2004 Congress on Evolutionary Computation (CEC 2004), Portland, Oregon, USA, pp. 204–211. IEEE (2004)
70. Peng, L., Wang, Y., Dai, G.: Advances in Computation and Intelligence: Third International Symposium, ISICA 2008 Wuhan, China, December 19-21, 2008 Proceedings, Chapter A Novel Opposition-Based Multi-objective Differential Evolution Algorithm for Multi-objective Optimization, pp. 162–170. Springer, Berlin (2008)
71. Praditwong, K., Yao, X.: How well do multi-objective evolutionary algorithms scale to large problems. In: Srinivasan, D., Wang, L. (eds.) Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC 2007), pp. 3959–3966. IEEE, Singapore (2007)
72. Prado, R.S., Silva, R.C.P., Guimarães, F.G., Neto, O.M.: Using differential evolution for combinatorial optimization: a general approach. In: Proceedings of the 2010 IEEE International Conference on Systems Man and Cybernetics (SMC), Istanbul, Turkey, pp. 11–18. IEEE (2010)
73. Price, K.V.: An introduction to differential evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 79–108. McGraw-Hill, London (1999)
74. Price, K.V., Storn, R., Lampinen, J.: Differential Evolution: A Practical Approach to Global Optimization. Springer, Berlin (2005)
75. Robič, T., Filipič, B.: DEMO: differential evolution for multiobjective optimization. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005), pp. 520–533. Springer, Guanajuato (2005)
76. Rönkkönen, J., Kukkonen, S., Lampinen, J.: A comparison of differential evolution and generalized generation gap model. J. Adv. Comput. Intell. Intell. Inf. **9**(5), 549–555 (2005)
77. Rönkkönen, J., Kukkonen, S., Price, K.V.: Real-parameter optimization with differential evolution. In: Proceedings of the 2005 Congress on Evolutionary Computation (CEC 2005), Edinburgh, Scotland, pp. 506–513. IEEE Press (2005)
78. Salomon, R.: Re-evaluating genetic algorithm performance under coordinate rotation of becnmark function. A survey of some theoretical and practical aspects of genetic algorithms. BioSystems **39**(3), 263–278 (1996)
79. Santana-Quintero, L.V., Coello Coello, C.A.: An algorithm based on Differential Evolution for multi-objective optimization. Int. J. Comput. Intell. Res. **1**(2), 151–169 (2005)
80. O. Schütze, S. Alvarado, C. Segura, and R. Landa. Gradient Subspace Approximation: A direct search method for memetic computing. *Soft Computing*, 2016. accepted for publication
81. Sharma, S., Rangaiah, G.P.: An improved multi-objective differential evolution with a termination criterion for optimizing chemical processes. Comput. Chem. Eng. **56**(13), 155–173 (2013)
82. Storn, R.: System design by constraint adaptation and differential evolution. IEEE Trans. Evolut. Comput. **3**(1), 22–34 (1999)
83. Storn, R., Price, K.V.: Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report, ICSI, University of California, Berkeley (1995). http://www.icsi.berkeley.edu/ftp/global/pub/techreports/1995/tr-95-012.pdf. Accessed 12 June 2005

84. Storn, R., Price, K.V.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997)
85. Tagawa, K.: Multi-objective optimum design of balanced SAW filters using generalized differential evolution. WSEAS Trans. Syst. **8**(8), 923–932 (2009)
86. Tagawa, K., Sasaki, Y., Nakamura, H.: Optimum design of balanced SAW filters using multi-objective differential evolution. In: Deb, K., Bhattacharya, A., Chakraborti, N., Chakroborty, P., Das, S., Dutta, J., Gupta, S., Jain, A., Aggarwal, V., Branke, J., Louis, S., Tan, K. (eds.) Proceedings of the 8th International Conference on Simulated Evolution and Learning (SEAL), Kanpur, India, pp. 466–475. Springer, Berlin (2010)
87. Tasgetiren, M.F., Suganthan, P.N., Pan, Q.-K.: An ensemble of discrete differential evolution algorithms for solving the generalized traveling salesman problem. Appl. Math. Comput. **215**(9), 3356–3368 (2010)
88. Tušar, T., Filipič, B.: Differential evolution versus genetic algorithms in multiobjective optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007), Matsushima, Japan, pp. 257–271. Springer, Berlin (2007)
89. Wang, F.-S., Sheu, J.-W.: Multiobjective parameter estimation problems of fermentation processes using a high ethanol tolerance yeast. Chem. Eng. Sci. **55**(18), 3685–3695 (2000)
90. Xue, F., Sanderson, A.C., Graves, R.J.: Pareto-based multi-objective differential evolution. In: Sarker, R., Reynolds, R., Abbass, H., Tan, K.C., McKay, B., Essam, D., Gedeon, T. (eds.) In: Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003), Canberra, Australia, pp. 862–869. IEEE Press, Piscataway (2003)
91. Zaharie, D.: Critical values for the control parameters of differential evolution algorithms. In: Matoušek, R., Ošmera, P. (eds.) Proceedings of Mendel 2002. 8th International Conference on Soft Computing, pp. 62–67. Brno, Czech Republic (2002)
92. Zaharie, D.: Multi-objective optimization with adaptive Pareto differential evolution. In: Teodorescu, H.-N., Gaindric, C., Sofron, E. (eds.) Proceedings of Symposium on Intelligent Systems and Applications (SIA 2003), Iasi, Romania. Performantica Press (2003)
93. Zapotecas-Martínez, S., Derbel, B., Liefooghe, A., Aguirre, H.E., Tanaka, K.: Geometric differential evolution in MOEA/D: a preliminary study. In: Sidorov, G., Galicia-Haro, S.N. (eds.) Proceedings of the 14th Mexican International Conference on Artificial Intelligence (MICAI 2015). Lecture Notes in Computer Science, vol. 9413, pp. 364–376. Springer, Cuernavaca (2015)
94. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evolut. Comput. **11**(6), 712–731 (2007)
95. Zhang, Q., Zhou, A., Jin, Y.: RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm. Evolutionary Computation, IEEE Transactions on **12**(1), 41–63 (2008). February
96. Zhong, J.-H., Zhang, J.: Adaptive multi-objective Differential Evolution with stochastic coding strategy. In: Krasnogor, N., Lanzi, P.L. (eds.) Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO 2011), Dublin, Ireland, pp. 666–672. ACM (2011)
97. Zielinski, K., Laur, R.: Variants of differential evolution for multi-objective optimization. In: Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM 2007), Honolulu, HI, USA, pp. 91–98. IEEE (2007)
98. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: empirical results. Evolut. Comput. **8**(2), 173–195 (2000)
99. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: Giannakoglou, K.C., Tsahalis, D.T., Périaux, J., Papaliouy, K.D., Fogarty, T. (eds.) Proceedings of the Third Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems (EUROGEN 2001), Athens, Greece, pp. 95–100. International Center for Numerical Methods in Engineering (CIMNE) (2002)