COMPARISON OF GENERALIZED DIFFERENTIAL EVOLUTION TO OTHER MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

Saku Kukkonen and Jouni Lampinen

Department of Information Technology
Lappeenranta University of Technology
P.O. Box 20, FIN-53851 Lappeenranta, Finland
e-mail: Saku.Kukkonen@lut.fi, Jouni.Lampinen@lut.fi

Key words: multi-objective optimization, Pareto-optimization, constraints, evolutionary algorithms, differential evolution

Abstract. In this paper an Evolutionary Algorithm, Differential Evolution, and its extension for constrained multi-objective (Pareto-)optimization, Generalized Differential Evolution, are described. Performance of Generalized Differential Evolution is tested with a set of five benchmark multi-objective test problems. Suitable control parameter values for these test problems are surveyed and the results are compared numerically with other multi-objective evolutionary algorithms including the Strength Pareto Evolutionary Algorithm and the Non-dominated Sorting Genetic Algorithm. Several metrics commonly used in the literature are applied to measure convergence to the Pareto-optimal front and diversity of the obtained solution. The results are suggesting that the performance of Generalized Differential Evolution is well comparable to the performance of the compared multi-objective evolutionary algorithms.
1 INTRODUCTION

Many situations in engineering and economics deal with optimization. One may want to optimize, e.g., manufacturing processes, shape of products, and number of different products to be manufactured. Typical goals are minimizing costs, maximizing profits, and improving performance. Several natural aspects limits feasible solutions, e.g., resources may cause limitations and/or the number of products cannot be a negative number.

Optimization is an intensively studied problem field in mathematics. However, functions to be optimized in traditional mathematics are relatively simple, e.g., continuous, convex, unimodal, differentiable, etc., yet functions to be optimized in practice are often far more complicated, e.g., discontinuous, non-convex, multi-modal, non-differentiable, etc. In these cases different stochastic optimization methods have shown their effectiveness.

Most optimization research deals with single-objective optimization problems. However, multi-objective and these problems are usually first converted to single-objective problems. Single-objective problems are commonly considered easier to solve but conversion from multi-objective problem to single-objective problem requires some a priori knowledge which is not necessarily available or which is hard to determine, e.g., the relative importance of each individual sub-objective. For this reason interest exists in solving multi-objective problems in multi-objective form.

Several extensions of Differential Evolution (DE) for multi-objective optimization have already been proposed. Most of these methods use a non-dominated sorting for reproduction in each generation and some distance metric to prevent crowding. Chang et al. introduce an extension of DE for solving multi-objective optimization problems [1]. This method uses selection of non-dominated solutions for further generations and uses a distance metric to maintain diversity. Abbass et al. propose several extensions of DE which modifies the original algorithm in several ways, e.g., the use of only non-dominated solutions for reproduction in each generation and a distance metric to prevent crowding [2-4]. Madavan describes an extension which uses the non-dominated sorting and a ranking selection procedure [5]. Babu and Jehan apply DE for multiple objectives using a penalty function and a weighted sum approaches [6]. Graves et al. and Xue use the non-dominated sorting for reproduction and a distance measure to prevent crowding [7,8].

This paper continues with the following parts: In Section 2 the concept of multi-objective optimization with constraints is handled briefly. Section 3 describes Differential Evolution algorithm and Section 4 describes its extension for constrained multi-objective optimization. Section 5 describes an empirical evaluation of the extension and finally conclusions are given in Section 6.
2 MULTI-OBJECTIVE OPTIMIZATION WITH CONSTRAINTS

Many practical problems have multiple objectives. For example, designing a wing of an aircraft may have objectives such as maximizing strength, minimizing weight, minimizing manufacturing costs, maximizing lifting force, minimizing drag, etc. Multiple objectives are almost always more or less conflicting.

Several aspects cause constraints to problems. In the previous example of the wing, the thickness of the metal parts used must be a positive number, shape limitations exist, some parts are available only in some predefined standard sizes, etc. Constraints can be divided into box or boundary constraints and constraint functions. Boundary constraints are used when the value of some optimized variable is limited to some range and constraint functions are representing more complicated constraints which are expressed as functions.

Multi-objective problems are often converted to single-objective problems by predefining weighting factors for different objectives, expressing the relative importance of each objective. However, this is impossible in many cases because a decision maker does not necessarily know beforehand how he/she wants to weight different objectives. Thus, a more convenient way is to keep multiple objectives of multi-objective problems and try to solve them in this form even though this may be harder to do in practice. Optimizing several objectives simultaneously without articulating the relative importance for each objective a priori, is often called as Pareto-optimization [9]. An obtained solution is Pareto-optimal if none of the objectives cannot be improved without impairing at least one other objective [10, p. 11–12]. If the obtained solution can be improved in such way that at least one objective improves and other objectives do not decline, then the new solution dominates the original solution. A set of Pareto-optimal solutions form a Pareto-optimal front. Approximation of the Pareto-optimal front is called as a set of non-dominated solutions because the solutions of this set are not dominating each others in the space of objective functions. From the set of the non-dominated solutions the decision maker may select one which has suitable values for different objectives. This can be viewed as a posteriori articulation of decision-makers preferences concerning the relative importance of each objective.

A mathematically constrained multi-objective optimization problem can be presented in the form [10, p. 37]

\[
\begin{align*}
\text{minimize} \quad & \{f_1(\bar{x}), f_2(\bar{x}), \ldots, f_K(\bar{x})\} \\
\text{subject to} \quad & \bar{x} \in S = \{\bar{x} \in \mathbb{R}^D | g(\bar{x}) = (g_1(\bar{x}), g_2(\bar{x}), \ldots, g_M(\bar{x}))^T \leq \bar{0}\}. 
\end{align*}
\]

Thus, there are \(K\) functions to be optimized and \(M\) constraint functions.

Maximization problems can be converted to minimization problems by multiplying the objective function by -1 because \(\max f_k(\bar{x}) \Leftrightarrow \min -f_k(\bar{x})\). All constraints can be converted to the form \(g_j(\bar{x}) \leq 0\) in the following way: \(g_j(\bar{x}) \geq 0 \Leftrightarrow -g_j(\bar{x}) \leq 0\), \(g_j(\bar{x}) = 0 \Leftrightarrow g_j(\bar{x}) \leq 0 \land -g_j(\bar{x}) \leq 0\). Boundary constraints can be presented also in the form of constraint functions, e.g., \(a \leq x_i \leq b \Leftrightarrow a - x_i \leq 0 \land x_i - b \leq 0\). Thereby the formulation in Eq. 1 is without loss of generality.
The major part of earlier mathematical research has concentrated on optimization problems where the functions are linear, differentiable, convex, or otherwise mathematically well behaving. However, in practical problems objective functions are often nonlinear, non-differentiable, discontinuous, multi-modal, etc. and no presumptions can be made about their behavior. Variables may also be integers or discrete instead of being continuous. Most traditional optimization methods cannot handle such complexity or do not perform in these cases in which the assumptions they are based on do not hold. For such problems stochastic optimization methods such as Simulated Annealing (SA) and Evolution Algorithms (EAs) have been demonstrated to be effective because they do not rely assumptions concerning the objective and constraint functions.

3 DIFFERENTIAL EVOLUTION

The Differential Evolution (DE) algorithm [11, 12] [13, pp. 79–108] belongs to the family of Evolution Algorithms and was introduced by Storn and Price in 1995 [14]. Design principles in DE were simplicity, efficiency, and use of floating-point encoding instead of binary numbers, which is usual way of coding in Genetic Algorithms (GA).

Like in a typical EA, the idea in DE is to have some random initial population which is then improved using selection, mutation, and crossover operations. Several ways exist to determine a stopping criteria for EAs but usually a predefined upper limit $G_{\text{max}}$ for the number of generations to be computed provides appropriate stopping condition.

3.1 Initialization of population

Values for the initial population in DE are typically drawn from uniform distribution. Formally this can be presented as [13, p. 81]:

$$\begin{align*}
P_G &= \{\vec{x}_{1,G}, \vec{x}_{2,G}, \ldots, \vec{x}_{NP,G}\}, \\
\vec{x}_{i,G} &= x_{j;i,G}, \\
x_{j;i,G=0} &= x^{(lo)}_j + \text{rand}_j[0, 1] \cdot (x^{(hi)}_j - x^{(lo)}_j), \\
i &= 1, 2, \ldots, NP, \quad NP \geq 4, \quad j = 1, 2, \ldots, D
\end{align*}$$

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

Other ways of initialization also exist, e.g., if some knowledge exist about the position of the optimum, part of the initial population may be initialized around the possible position of the optimum using normal distribution.
3.2 Mutation and crossover

DE goes through each objective vector $\vec{x}_{i,G}$ of the population and creates a corresponding trial vector $\vec{u}_{i,G}$ as follows [13, p. 82]:

\[
\begin{align*}
  r_1, r_2, r_3 &\in \{1, 2, \ldots, NP\}, \\
  (\text{randomly selected, except: } r_1 \neq r_2 \neq r_3 \neq i) \\
  j_{\text{rand}} &= \text{int} (\text{rand}[0,1) \cdot D) + 1 \\
  \text{for}(j = 1; j \leq D; j = j + 1) \\
  \{ \\
    \text{if} (\text{rand}_j[0,1) < CR \lor j = j_{\text{rand}}) \\
    u_{j,i,G} = x_{j,r_3,G} + F \cdot (x_{j,r_1,G} - x_{j,r_2,G}) \\
    \text{else} \\
    u_{j,i,G} = x_{j,i,G} \\
  \}
\end{align*}
\]

Indices $r_1$, $r_2$, and $r_3$ are mutually different and drawn from the set of the population indices. 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 instead of from the old objective vector $\vec{x}_{i,G}$. Parameter $F$ is a scaling factor for mutation and its value is typically $(0, 1+]$. The condition “$j = j_{\text{rand}}$” is to make sure that at least one element is different compared to elements of the old population member.

The difference between two randomly chosen vectors ($\vec{x}_{r_1,G} - \vec{x}_{r_2,G}$) defines magnitude and direction of mutation. When the difference is added to a third randomly chosen vector $\vec{x}_{r_3,G}$, this corresponds mutation of this third vector. The basic idea of DE is that mutation is self-adaptive to the objective function space and to the current population. At the beginning of generations a magnitude of mutation is large because vectors in the population are far away in the search space. When evolution proceeds and population converges, the magnitude of mutation gets smaller. The self-adaptive mutation of DE permits to perform global search.

3.3 Selection

After each mutation and crossover operation the trial vector $\vec{u}_{i,G}$ is compared to the old objective vector $\vec{x}_{i,G}$. If the trial vector has equal or lower cost value, then it replaces the old vector. This can be presented as follows [13, p. 82]:

\[
\vec{x}_{i,G+1} = \begin{cases} 
  \vec{u}_{i,G} & \text{if } f(\vec{u}_{i,G}) \leq f(\vec{x}_{i,G}) \\
  \vec{x}_{i,G} & \text{otherwise}
\end{cases}
\]

The average cost value of the population will never increase, because the trial vector replaces the old vector only if it has equal or lower cost value.
Overall presentation of the whole DE/rand/1/bin algorithm is presented in Eq. 5 [13, p. 83]. Several variations of the basic DE algorithm exist and this is a reason for complicated notation. In the notation DE/x/y/z x indicates how the mutated vector is selected (it could be selected also to be best among the current population), y indicates number of vector differences used in the mutation, and z indicates the way the old vector and the trial vector are recombined (an alternative exponential recombination procedure is also mentioned in the literature [13, p. 98]).

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

$$\text{Initialize : } \forall i \leq NP \wedge \forall j \leq D : x_{j,i,G=0} = x_{j}^{(lo)} + \text{rand}[0,1] \cdot (x_{j}^{(hi)} - x_{j}^{(lo)})$$

While \( G < G_{\text{max}} \)

$$\forall i \leq NP \text{ Mutate and recombine: }$$

$$\begin{cases} 
  r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}, \text{ randomly selected,} \\
  j_{\text{rand}} \in \{1, 2, \ldots, D\}, \text{ randomly selected for each } i \\
  \text{except: } r_1 \neq r_2 \neq r_3 \neq i 
\end{cases}$$

$$\forall j \leq D, u_{j,i,G} = \begin{cases} 
  x_{j,r_3,G} + F \cdot (x_{j,r_1,G} - x_{j,r_2,G}) & \text{if } \text{rand}[0,1] < CR \vee j = j_{\text{rand}} \\
  x_{j,i,G} & \text{otherwise} 
\end{cases}$$

Select :

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

$$G = G + 1$$

4 GENERALIZED DIFFERENTIAL EVOLUTION

Several extensions of DE for multi-objective optimization exists as well for constrained optimization [1–8, 15–18]. The approach presented in this paper combines these by modifying the selection operation of the basic DE algorithm [19–22]. Compared to other DE extensions for multi-objective optimization, this approach makes DE suitable for constrained multi-objective optimization with minimum changes to the original algorithm. The modified selection operation for \( M \) constraint and \( K \) objective functions is presented
Saku Kukkonen and Jouni Lampinen

formally in Eq. 6 [21].

\[
\tilde{x}_{i,G+1} = \begin{cases} 
\tilde{u}_{i,G} & \text{if } \exists j \in \{1, \ldots, M\} : g_j(\tilde{u}_{i,G}) > 0 \\
\land \\
\forall j \in \{1, \ldots, M\} : g_j(\tilde{u}_{i,G}) \leq g_j(\tilde{x}_{i,G}) \\
\lor \\
\forall j \in \{1, \ldots, M\} : g_j(\tilde{x}_{i,G}) > 0 \\
\lor \\
\forall j \in \{1, \ldots, M\} : g_j(\tilde{x}_{i,G}) \leq 0 \\
\lor \\
\forall k \in \{1, \ldots, K\} : f_k(\tilde{u}_{i,G}) \leq f_k(\tilde{x}_{i,G}) \\
\tilde{x}_{i,G} & \text{otherwise}
\end{cases}
\]

(6)

where \(g_j'(\tilde{x}_{i,G}) = \max(g_j(\tilde{x}_{i,G}), 0)\) and \(g_j'(\tilde{u}_{i,G}) = \max(g_j(\tilde{u}_{i,G}), 0)\) are representing the constraint violations.

The selection rule given in Eq. 6 selects the trial vector \(\tilde{u}_{i,G}\) to replace the old vector \(\tilde{x}_{i,G}\) in three cases:

1. Both the trial vector \(\tilde{u}_{i,G}\) and the old vector \(\tilde{x}_{i,G}\) violate at least one constraint but the trial vector does not violate any of the constraints more than the old vector.

2. The old vector \(\tilde{x}_{i,G}\) violates at least one constraint whereas the trial vector \(\tilde{u}_{i,G}\) is feasible.

3. Both vectors are feasible and the trial vector \(\tilde{u}_{i,G}\) has less or equal cost value for each objective than the old vector \(\tilde{x}_{i,G}\).

Otherwise the old vector \(\tilde{x}_{i,G}\) is preserved.

The basic idea in the selection rule is that the trial vector \(\tilde{u}_{i,G}\) is required to dominate the compared old population member \(\tilde{x}_{i,G}\) in constraint violation space or in objective function space, or at least provide an equally good solution as \(\tilde{x}_{i,G}\). The principle is effectively rather similar to the method described in [23, pp. 131–132] even though the formulation is different. In this other method selection is based on the value of a penalized objective function \(F(\tilde{x})\) [24]:

\[
F(\tilde{x}) = \begin{cases} 
f(\tilde{x}) & \text{if } \tilde{x} \text{ is feasible} \\
 f_{\max} + \sum_{j=1}^{M} \max(g_j(\tilde{x}), 0) & \text{otherwise}
\end{cases}
\]

(7)

Here, \(f_{\max}\) denotes the objective function value of the worst feasible solution in the population. The selection rules given in Eq. 6 and based on Eq. 7 perform in the same way.
when two feasible solutions are compared or when a feasible solution is compared to an infeasible solution. In the case of two infeasible solutions, the selection rule in Eq. 6 compares solutions based on dominance of the constraint violations whereas the selection method based on Eq. 7 compares solutions based on a sum of the constraint violations. The selection method based on Eq. 7 needs search of the objective function value of the worst feasible solution in the population, calculation of all constraint function values in the case of infeasible solution, and it also permits worsening of individual constraint function values because sums of the constraint violations are compared instead of individual constraint violations. The selection method based on Eq. 7 may also need normalization of different constraints in the case of different order of magnitude whereas this is not needed in the selection rule in Eq. 6.

The selection rule in Eq. 6 can be implemented in such a way that the number of function evaluations is reduced because not always all the constraints and objectives need to be evaluated, e.g., inspecting constraint violations (even one constraint) is often enough to determine which vector to select for the next generation [21,22].

One should note that the selection rule in Eq. 6 handles any number $M$ of constraints and any number $K$ of objectives, including cases $M = 0$ (unconstrained problem) and $K = 0$ (constraint satisfaction problem). When $M = 0$ and $K = 1$, the selection rule is identical to the selection rule of the basic DE algorithm. Because the described selection method extends DE for constrained multi-objective optimization and the basic DE algorithm is a special case, the method with the selection rule described is named Generalized Differential Evolution (GDE).

After the selected number of generations the final population presents a solution for the optimization problem. The non-dominated solutions can be separated from the final population if desired. There is no sorting of non-dominated solutions during the optimization process or explicit mechanism for maintaining diversity of solutions.

5 EXPERIMENTS

GDE was implemented in C and tested with a set of five multi-objective benchmark problems described in [25] and [26, pp. 57–59]. These problems are known as ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 [23, pp. 356–360]. They are designed to test the ability of a multi-objective optimization method to handle convexity (ZDT1), non-convexity (ZDT2), discontinuity (ZDT3), multi-modality (ZDT4), and non-uniformity (ZDT6) of the Pareto-optimal front. Preliminary tests with these problems and problems including constraint functions are reported in [27,28].
Saku Kukkonen and Jouni Lampinen

\begin{align*}
\text{Minimize} \\
\quad f_1(\bar{x}) &= x_1 \\
\quad f_2(\bar{x}) &= g \times \left(1 - \sqrt{f_1/g}\right) \\
\quad g &= 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i \\
\text{subject to} \\
\quad x_i &\in [0, 1], n = 30
\end{align*}
\tag{8}

\begin{align*}
\text{Minimize} \\
\quad f_1(\bar{x}) &= x_1 \\
\quad f_2(\bar{x}) &= g \times \left(1 - \frac{f_1}{g} \right)^2 \\
\quad g &= 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i \\
\text{subject to} \\
\quad x_i &\in [0, 1], n = 30
\end{align*}
\tag{9}

\begin{align*}
\text{Minimize} \\
\quad f_1(\bar{x}) &= x_1 \\
\quad f_2(\bar{x}) &= g \times \left(1 - \sqrt{f_1/g} - \left(\frac{f_1}{g}\right) \sin(10\pi f_1)\right) \\
\quad g &= 1 + \frac{9}{n-1} \sum_{i=2}^{n} x_i \\
\text{subject to} \\
\quad x_i &\in [0, 1], n = 30
\end{align*}
\tag{10}

\begin{align*}
\text{Minimize} \\
\quad f_1(\bar{x}) &= x_1 \\
\quad f_2(\bar{x}) &= g \times \left(1 - \sqrt{f_1/g}\right) \\
\quad g &= 1 + 10(n - 1) + \sum_{i=2}^{n} \left(x_i^2 - 10 \cos(4\pi x_i)\right) \\
\text{subject to} \\
\quad x_1 &\in [0, 1], x_i \in [-5, 5], n = 10
\end{align*}
\tag{11}

\begin{align*}
\text{Minimize} \\
\quad f_1(\bar{x}) &= 1 - \exp(-4x_1) \sin^6(6\pi x_1) \\
\quad f_2(\bar{x}) &= g \times (1 - \left(1/f_1/g\right)^2) \\
\quad g &= 1 + 9 \left(\sum_{i=2}^{n} x_i / 9\right)^{0.25} \\
\text{subject to} \\
\quad x_i &\in [0, 1], n = 10
\end{align*}
\tag{12}

To illustrate two of these problems, a partial search region with the global Pareto-optimal front for ZDT4 and a Pareto-optimal front for ZDT6 are presented in Figure 1. Solution for ZDT6 contains points which are uniformly distributed in the decision variable space. One can notice that the density of solutions across the Pareto-optimal front is non-uniform.
Boundary constraint violations were handled here according to the following rule:

\[ u_{j,i,G} = \begin{cases} 
  x_j^{(lo)} & \text{if } u_{j,i,G} < x_j^{(lo)} \\
  x_j^{(hi)} & \text{if } u_{j,i,G} > x_j^{(hi)} \\
  u_{j,i,G} & \text{otherwise}
\end{cases} \tag{13} \]

5.1 Experimental results and discussions

In all the test problems the size of the population was 100, the number of generations was 250, and the control parameter values were \( CR = 0.10, \ F = 0.10 \). In preliminary tests different control parameter values in the range \( CR \in [0,1] \) and \( F \in [0,5] \) with a resolution of 0.05 were tested and a suitable crossover rate and mutation factor was visually thereby approximately determined. The results for GDE solving the multi-objective benchmark problems are shown in Figures 2–4, where results obtained with the Strength Pareto Evolutionary Algorithm (SPEA) [29] and the Non-dominated Sorting Genetic Algorithm (NSGA) [23, pp. 209–218] are also shown with known global Pareto-optimal fronts. SPEA and NSGA were selected for comparison because of their good performance in previous comparison tests with the other multi-objective optimization methods [25] and since both are well known within the multi-objective optimization community. The results for SPEA, NSGA, and GDE given in Figures 2–4 are after one run, and solutions shown for GDE contain the non-dominated members of the final population.

The tests for GDE were repeated 30 times with different seeds of the random number generator and these results were compared with the corresponding results of SPEA,
Figure 2: Global Pareto-optimal front and solutions obtained with SPEA, NSGA, and GDE for a) ZDT1 b) ZDT2.

Figure 3: Global Pareto-optimal front and solutions obtained with SPEA, NSGA, and GDE for a) ZDT3 b) ZDT4.
NSGA, the Fonseca and Fleming’s multi-objective GA (FFGA) [23, pp. 200–209], the Niched-Pareto Genetic Algorithm (NPGA) [23, pp. 218–223], and the Vector Evaluated Genetic Algorithm (VEGA) [23, pp. 179–188]. The results for these MOEAs were obtained from the Internet [30]. From the methods in comparison, GDE and SPEA are elitist MOEAs and the rest methods are non-elitist MOEAs.

Closeness to the Pareto-optimal front was measured with an error ratio (ER) and a generational distance (GD) [23, pp. 324–327]. Diversity of the obtained solution was measured using spacing (S), spread (Δ), and maximum spread (D) metrics [23, pp. 328–331]. Smaller values for the error ratio, generational distance, spacing, and spread are preferable. The optimal value for the maximum spread is 1. Results of the MOEAs were compared against each other using a set coverage $C$ metric [26] and a $V$ measure [31, 32]. The $C(A, B)$ metric measures the fraction of members of $B$ that are dominated by members of $A$. The $V(A, B)$ measures the fraction of the volume of the minimal hypercube containing both fronts that is dominated by members of $A$ but is not dominated by members of $B$. Greater values for the $C$ and the $V$ metrics are desirable.

Average execution times and average numbers of needed function evaluations of GDE for the benchmark problems are reported in Table 1. Tables 2–6 contain the performance measurements for different MOEAs solving the benchmark problems. Solutions of GDE contained the non-dominated members of the final population.

In the most of the test cases GDE has more non-dominated solution members than the other MOEAs in this comparison. The results show that GDE converged closer to the true Pareto-optimal front than the other MOEAs but the diversity metrics show that obtained solution is not optimally diverse. Even though GDE has members in the final solution close to the true Pareto-optimal fronts, the end points of the Pareto-optimal fronts are not exactly reached and the distribution of solutions could be better. It was
Table 1: Average execution time and average number of needed function evaluations (standard deviations in parenthesis) of GDE for the multi-objective benchmark test problems. All the tests were run on a Sun Sparc Ultra2.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Execution time</th>
<th>Number of function evaluations</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZDT1</td>
<td>1.019 (0.01) s</td>
<td>$f_1$ 25100 (0.0) $f_2$ 23474.2 (32.4)</td>
</tr>
<tr>
<td>ZDT2</td>
<td>1.019 (0.01) s</td>
<td>$f_1$ 25100 (0.0) $f_2$ 23558.9 (69.0)</td>
</tr>
<tr>
<td>ZDT3</td>
<td>1.041 (0.01) s</td>
<td>$f_1$ 25100 (0.0) $f_2$ 23490.0 (38.7)</td>
</tr>
<tr>
<td>ZDT4</td>
<td>0.547 (0.01) s</td>
<td>$f_1$ 25100 (0.0) $f_2$ 22732.3 (41.2)</td>
</tr>
<tr>
<td>ZDT6</td>
<td>0.570 (0.01) s</td>
<td>$f_1$ 25100 (0.0) $f_2$ 22184.2 (93.1)</td>
</tr>
</tbody>
</table>

Table 2: Means of the solution cardinality ($N$), error ratio (ER), generational distance (GD), spacing (S), spread ($\Delta$), maximum spread (D), $C$, and $V$ of MOEAs for ZDT1. Standard deviations are in parenthesis.
<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>ER</th>
<th>GD</th>
<th>S</th>
<th>Δ</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFGA</td>
<td>15.2(4.8)</td>
<td>1.000(0.000)</td>
<td>0.522(0.069)</td>
<td>0.080(0.040)</td>
<td>0.884(0.044)</td>
<td>1.711(0.552)</td>
</tr>
<tr>
<td>NPGA</td>
<td>9.8(2.4)</td>
<td>1.000(0.000)</td>
<td>0.333(0.043)</td>
<td>0.109(0.046)</td>
<td>0.916(0.053)</td>
<td>0.828(0.103)</td>
</tr>
<tr>
<td>VEGA</td>
<td>2.9(1.3)</td>
<td>1.000(0.000)</td>
<td>0.450(0.111)</td>
<td>0.244(0.231)</td>
<td>0.920(0.096)</td>
<td>0.360(0.212)</td>
</tr>
<tr>
<td>SPEA</td>
<td>43.5(9.1)</td>
<td>0.872(0.196)</td>
<td>1.000(0.000)</td>
<td>0.061(0.027)</td>
<td>0.807(0.071)</td>
<td>0.826(0.036)</td>
</tr>
<tr>
<td>NSGA</td>
<td>19.9(3.3)</td>
<td>0.016(0.049)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.963(0.128)</td>
<td></td>
</tr>
<tr>
<td>GDE</td>
<td>45.2(11.0)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.956(0.040)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>V</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FFGA</td>
<td>0.016(0.049)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.963(0.128)</td>
</tr>
<tr>
<td>NPGA</td>
<td>0.963(0.128)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.956(0.040)</td>
</tr>
<tr>
<td>VEGA</td>
<td>1.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.956(0.040)</td>
</tr>
<tr>
<td>SPEA</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.956(0.040)</td>
</tr>
<tr>
<td>NSGA</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.956(0.040)</td>
</tr>
<tr>
<td>GDE</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.956(0.040)</td>
</tr>
</tbody>
</table>

Table 3: Means of the solution cardinality (R), error ratio (ER), generational distance (GD), spacing (S), spread (Δ), and maximum spread (D) of MOEAs for ZDT2. Standard deviations are in parenthesis.

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>ER</th>
<th>GD</th>
<th>S</th>
<th>Δ</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFGA</td>
<td>28.0(6.0)</td>
<td>0.989(0.024)</td>
<td>0.268(0.037)</td>
<td>0.044(0.019)</td>
<td>0.849(0.043)</td>
<td>1.405(0.193)</td>
</tr>
<tr>
<td>NPGA</td>
<td>28.1(4.5)</td>
<td>0.953(0.076)</td>
<td>0.194(0.029)</td>
<td>0.044(0.015)</td>
<td>0.840(0.044)</td>
<td>1.229(0.081)</td>
</tr>
<tr>
<td>VEGA</td>
<td>23.1(5.0)</td>
<td>0.981(0.032)</td>
<td>0.166(0.029)</td>
<td>0.058(0.025)</td>
<td>0.838(0.052)</td>
<td>0.990(0.100)</td>
</tr>
<tr>
<td>SPEA</td>
<td>69.7(9.7)</td>
<td>0.034(0.017)</td>
<td>0.002(0.001)</td>
<td>0.013(0.011)</td>
<td>0.784(0.045)</td>
<td>0.931(0.019)</td>
</tr>
<tr>
<td>NSGA</td>
<td>46.1(6.8)</td>
<td>0.530(0.064)</td>
<td>0.290(0.066)</td>
<td>0.022(0.012)</td>
<td>0.813(0.041)</td>
<td>0.977(0.038)</td>
</tr>
<tr>
<td>GDE</td>
<td>78.1(4.6)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.016(0.003)</td>
<td>1.085(0.064)</td>
<td>0.901(0.018)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>V</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FFGA</td>
<td>0.016(0.026)</td>
<td>0.000(0.001)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
</tr>
<tr>
<td>NPGA</td>
<td>0.072(0.036)</td>
<td>0.000(0.001)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
</tr>
<tr>
<td>VEGA</td>
<td>0.223(0.060)</td>
<td>0.198(0.048)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
</tr>
<tr>
<td>SPEA</td>
<td>0.399(0.051)</td>
<td>0.416(0.036)</td>
<td>0.324(0.031)</td>
<td>0.044(0.017)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
</tr>
<tr>
<td>NSGA</td>
<td>0.347(0.051)</td>
<td>0.357(0.034)</td>
<td>0.245(0.034)</td>
<td>0.000(0.001)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
</tr>
<tr>
<td>GDE</td>
<td>0.402(0.052)</td>
<td>0.421(0.036)</td>
<td>0.334(0.031)</td>
<td>0.036(0.008)</td>
<td>0.166(0.014)</td>
<td>0.000(0.000)</td>
</tr>
</tbody>
</table>

Table 4: Means of the solution cardinality (R), error ratio (ER), generational distance (GD), spacing (S), spread (Δ), and maximum spread (D) of MOEAs for ZDT3. Standard deviations are in parenthesis.
### Table 5: Means of the solution cardinality (N), error ratio (ER), generational distance (GD), spacing (S), spread (Δ), and maximum spread (D) of MOEAs for ZDT4. Standard deviations are in parenthesis.

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>ER</th>
<th>GD</th>
<th>S</th>
<th>Δ</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFGA</td>
<td>8.7(3.4)</td>
<td>1.000(0.000)</td>
<td>19.699(3.403)</td>
<td>0.203(0.154)</td>
<td>0.912(0.062)</td>
<td>30.917(10.658)</td>
</tr>
<tr>
<td>NPGA</td>
<td>4.7(1.8)</td>
<td>1.000(0.000)</td>
<td>9.266(2.834)</td>
<td>0.232(0.168)</td>
<td>0.926(0.041)</td>
<td>6.755(3.890)</td>
</tr>
<tr>
<td>VEGA</td>
<td>2.2(1.1)</td>
<td>1.000(0.000)</td>
<td>8.301(2.574)</td>
<td>0.346(0.342)</td>
<td>0.964(0.049)</td>
<td>1.053(1.203)</td>
</tr>
<tr>
<td>SPEA</td>
<td>84.8(50.5)</td>
<td>1.000(0.000)</td>
<td>0.570(0.322)</td>
<td>0.029(0.049)</td>
<td>0.950(0.041)</td>
<td>1.861(0.388)</td>
</tr>
<tr>
<td>NSGA</td>
<td>7.6(2.7)</td>
<td>1.000(0.000)</td>
<td>1.983(0.784)</td>
<td>0.216(0.102)</td>
<td>0.924(0.038)</td>
<td>1.492(0.558)</td>
</tr>
<tr>
<td>GDE</td>
<td>39.3(21.2)</td>
<td>0.316(0.416)</td>
<td>0.016(0.025)</td>
<td>0.054(0.036)</td>
<td>0.865(0.113)</td>
<td>0.902(0.105)</td>
</tr>
</tbody>
</table>

### Table 6: Means of the solution cardinality (N), error ratio (ER), generational distance (GD), spacing (S), spread (Δ), and maximum spread (D) of MOEAs for ZDT6. Standard deviations are in parenthesis.

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>ER</th>
<th>GD</th>
<th>S</th>
<th>Δ</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFGA</td>
<td>11.8(3.2)</td>
<td>1.000(0.000)</td>
<td>1.541(0.245)</td>
<td>0.098(0.055)</td>
<td>0.939(0.030)</td>
<td>2.010(0.703)</td>
</tr>
<tr>
<td>NPGA</td>
<td>5.9(2.3)</td>
<td>1.000(0.000)</td>
<td>1.473(0.352)</td>
<td>0.191(0.132)</td>
<td>0.945(0.030)</td>
<td>0.881(0.292)</td>
</tr>
<tr>
<td>VEGA</td>
<td>3.0(1.3)</td>
<td>1.000(0.000)</td>
<td>2.016(0.477)</td>
<td>0.207(0.196)</td>
<td>0.948(0.052)</td>
<td>0.987(0.906)</td>
</tr>
<tr>
<td>SPEA</td>
<td>12.0(5.1)</td>
<td>0.944(0.216)</td>
<td>0.151(0.057)</td>
<td>0.136(0.095)</td>
<td>0.956(0.168)</td>
<td>0.847(0.087)</td>
</tr>
<tr>
<td>NSGA</td>
<td>6.7(2.3)</td>
<td>1.000(0.000)</td>
<td>0.724(0.159)</td>
<td>0.154(0.095)</td>
<td>0.951(0.039)</td>
<td>0.747(0.076)</td>
</tr>
<tr>
<td>GDE</td>
<td>99.8(1.0)</td>
<td>0.067(0.254)</td>
<td>0.013(0.050)</td>
<td>0.013(0.003)</td>
<td>0.906(0.064)</td>
<td>0.979(0.064)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>C(1)</th>
<th>C(2)</th>
<th>V(1)</th>
<th>V(2)</th>
<th>V(3)</th>
<th>V(4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFGA</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
</tr>
<tr>
<td>NPGA</td>
<td>0.067(0.221)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.885(0.250)</td>
<td>0.000(0.000)</td>
</tr>
<tr>
<td>VEGA</td>
<td>0.054(0.212)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>1.000(0.000)</td>
<td>0.933(0.254)</td>
</tr>
<tr>
<td>SPEA</td>
<td>0.574(0.484)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>1.000(0.000)</td>
<td>1.000(0.000)</td>
</tr>
<tr>
<td>NSGA</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.993(0.027)</td>
<td>1.000(0.000)</td>
</tr>
<tr>
<td>GDE</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>C(1)</th>
<th>C(2)</th>
<th>V(1)</th>
<th>V(2)</th>
<th>V(3)</th>
<th>V(4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFGA</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
</tr>
<tr>
<td>NPGA</td>
<td>0.024(0.109)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.498(0.122)</td>
<td>0.000(0.000)</td>
</tr>
<tr>
<td>VEGA</td>
<td>0.022(0.095)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.747(0.227)</td>
<td>0.000(0.000)</td>
</tr>
<tr>
<td>SPEA</td>
<td>0.022(0.095)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.489(0.119)</td>
<td>0.000(0.000)</td>
</tr>
<tr>
<td>NSGA</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.885(0.221)</td>
<td>0.000(0.000)</td>
</tr>
<tr>
<td>GDE</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.000(0.000)</td>
<td>0.993(0.027)</td>
<td>1.000(0.000)</td>
</tr>
</tbody>
</table>

15
also noticed that suitable values for the crossover rate and the mutation factor should be drawn from a rather narrow range for some problems, e.g., problem ZDT4, while the underlying reason for this remains open.

Usually large values (such as 0.9) are suggested as initial settings for the crossover rate \( CR \) and the mutation factor \( F \) in the case of single-objective problems. In the case of multiple objectives, it was observed that using a large crossover rate often leads to faster convergence along one objective compared to another. This causes the solution to converge to one point of the Pareto-optimal front, which is not desired. When \( CR \approx 0.05 \ldots 0.5 \), the obtained final population typically remains diverse as desired. Thus, searching the decision variable space along the directions of the coordinate axis performs better than a rotationally invariant search. Use of small values for \( CR \) was also reported in [2,3,5]. However, the current recommendations are based on limited experimentation and the problem of selecting the control parameter values is remaining mostly open.

Increasing the size of the population seemed to provide better approximation of the Pareto-optimal front without needing to compute more generations. However, the value \( NP = 100 \) was used here for comparison with the results of other MOEAs.

6 CONCLUSIONS AND FUTURE RESEARCH

In this paper the Differential Evolution algorithm and its extension for constrained multi-objective optimization are described. The method described extends the basic DE algorithm for constrained multi-objective optimization with minor changes to the original algorithm of DE and for this reason the method is named Generalized DE (GDE). It is effective and does not introduce any extra control parameters.

GDE is tested with five benchmark multi-objective test problems. The numerical results show that the method is able to provide a solution for all the test problems and performs well compared to other MOEAs in comparison, providing a relatively good approximation of the Pareto-optimal front. However, despite the distribution of the solution points along the approximated Pareto-optimal front was well comparable to the other MOEAs, it is still far from ideal. When the method is used for multi-objective optimization problems, our preliminary recommendations for the control parameter values are \( CR \in [0.05, 0.5] \) and \( F \in [0.05, 1+) \) for initial settings. The recommendation for \( CR \) differ clearly from those given in literature for solving single-objective problems, e.g., \( CR = 0.9 \) [12, p. 129].

More intensive research of the effect of parameters on the optimization process, extensive comparison of GDE with latest multi-objective evolutionary algorithms and test problems, and applying GDE for practical multi-objective problems remains to be studied. Also, distribution of solutions and extent of the obtained non-dominated front could be improved because now GDE does not contain any mechanism for maintaining these.
ACKNOWLEDGEMENTS

S. Kukkonen gratefully acknowledges support from the East Finland Graduate School in Computer Science and Engineering (ECSE) and the Academy of Finland. The authors would like to thank K. E. Parsopoulos for useful discussions about performance metrics and E. Zitzler for providing results of other MOEAs.
REFERENCES


