A PARAMETER CONTROL SCHEME FOR DE INSPIRED BY ACO

Dražen Bajer, Goran Martinović
Faculty of Electrical Engineering, Josip Juraj Strossmayer University of Osijek, Croatia
drazen.bajer@etfos.hr, goran.martinovic@etfos.hr

Abstract
Differential evolution requires a prior setting of its parameters. Appropriate values are not always easy to determine, even more since they may change during the optimisation process. This is where parameter control comes in. Accordingly, a scheme inspired by ant colony optimisation for controlling the crossover-rate and mutation factor is proposed. Conceptually, artificial ants guided by a pheromone model select parameter value pairs for each individual. The pheromone model is steadily updated in order to reflect the current optimisation state. Promising results were achieved in the comprehensive experimental analysis. Nonetheless, much room for potential improvements is available.

Keywords: Ant colony optimisation, Construction graph, Differential evolution, Parameter control, Pheromone model.

1. Introduction

Differential evolution (DE) [17, 19] is a simple and effective population-based search and optimisation method. It has been successfully applied to various global optimisation problems, although originally proposed for numerical optimisation (see, e.g., [2, 6, 10, 12]). Like other evolutionary algorithms (EAs), DE also requires the setting of its parameters prior to running and its performance is largely dependent on those settings. Another issue is the fact that different parameter settings may be suitable for different problems [5, 22, 23]. Accordingly, although various recommendations for setting the parameters exist (see, e.g., [17, 19]), the search for adequate parameter values (parameter tuning) that will result in satisfactory performance on a given problem requires a considerable effort and usually boils down to a trial and error approach. Furthermore, “optimal” values of the parameters may change during the optimisation [8, 23]. This problem cannot be solved by just finding good parameter values a priori, but requires a constant adjustment of those values.
parameter control). The need for parameter control in EAs was realised early [5]. Finally, well designed schemes can improve or enhance performance in terms of robustness and convergence-rate [23].

This paper proposes a parameter control scheme for DE, inspired by ant colony optimisation (ACO). More particularly, a scheme for controlling the parameters representing the crossover-rate and mutation factor. Two sets of parameter values are provided, and for each individual a value pair is selected. Conceptually, the selection is made by artificial ants (each solution is assigned one), while their choices are influenced by artificial pheromone.

The remainder of the paper is organised as follows. Section 2 briefly describes DE and ACO, it also provides a short overview of parameter control in DE. The proposed parameter control scheme is described in Section 3. In Section 4 the obtained experimental results are reported and discussed. Finally, the drawn conclusions are stated in Section 5.

2. Preliminaries

2.1 Differential Evolution

Differential evolution is an example of a fairly successful EA for numerical optimisation. Besides that, it is conceptually simple which makes it attractive to practitioners attempting to solve their problem or problems using a bio-inspired optimisation algorithm. A brief description of the canonical or standard DE, as outlined in Algorithm 1, follows.

The population of DE is composed of NP individuals typically called vectors $v_j = (v_{j1}, \ldots, v_{jd}) \in \mathbb{R}^d$, $j = 1, \ldots, NP$. In each generation/iteration a new population is created by mutation and crossover of individuals, i.e., vectors of the current population. Mutation and crossover produce a trial vector (offspring)

$$t_i^j = \begin{cases} v_{i1}^r + F \cdot (v_{i2}^r - v_{i3}^r), & \text{if } U_i(0,1) \leq CR \text{ or } i = r_j \\ v_{ij}, & \text{otherwise} \end{cases}, \quad i = 1, \ldots, d, \quad (1)$$

where $v_{i1}^r$, $v_{i2}^r$ and $v_{i3}^r$ are randomly selected vectors from the current population, and which are selected anew for each target vector $v_i^r$, such that $j \neq r \neq r_2 \neq r_3$. The parameter $F \in [0, \infty)$ is the scale (mutation) factor, while $CR \in [0, 1]$ is the crossover-rate, $U_i(0,1)$ is a uniform deviate in $[0, 1]$, and $r_j$ is randomly chosen from the set $\{1, \ldots, d\}$. After the new trial vector population of size $NP$ is created, a one-to-one selection takes place. More specifically, a trial vector $t_i^j$ passes into the next generation.
only if it is equal or better (in terms of the objective function $f$) than
the corresponding target vector $\mathbf{v}_j$.

The described algorithm represents the canonical DE, usually denoted
as DE/rand/1/bin [17, 19]. Many other variants that improve on the
canonical algorithm have been proposed in the literature. A comprehen-
sive review of DE variants can be found in, e.g., [14].

2.2 Parameter control in differential evolution

Essentially, the DE algorithm requires the setting of three parameters.
A multitude of different parameter control mechanisms or schemes for
DE can be found in the literature. Most of those schemes are designed
for controlling a subset of the parameters, most notably $CR$ and $F$. For
example, a few simple deterministic mechanisms for varying the value
of parameter $F$ during the optimisation process can be found in [3, 9].
Those propose to randomly vary $F$ inside a preset interval, or to linearly
reduce it with the number of performed iterations. Tvrđík [21] proposed,
among other, a scheme where for each individual one value pair of $F$ and
$CR$, from nine available, is chosen probabilistically. The probabilities as-
associated with the pairs are based on their success. The adaptive scheme
by Yu and Zhang [22] utilizes the quality of population individuals and
their distance from the best-so-far. Based on that data and defined rules
the values of $F$ and $CR$ are appropriately adjusted. In the self-adaptive
scheme proposed by Brest et al. [1], each individual is assigned its own
value of $F$ and $CR$. The assigned values are regenerated randomly before
mutation and crossover take place with set probabilities. Noman et al.
[15] took a similar approach, where the parameter values assigned to a
individual are regenerated (randomly) only if its offspring is worse than
the current population average. Furthermore, Zhang and Sanderson [23]
proposed a scheme where the values of $CR$ and $F$ assigned to each in-
dividual are generated by Gaussian and Cauchy deviates, respectively.

Algorithm 1 Canonical DE (DE/rand/1/bin)

1: Set $NP$, $CR$ and $F$, and initialize population
2: while termination condition not met do
3:   for $j := 1 \rightarrow NP$ do
4:     create trail vector $\mathbf{t}_j$ (Eq. (1))
5:   end for
6:   for $j := 1 \rightarrow NP$ do
7:     if $f(\mathbf{t}_j) \leq f(\mathbf{v}_j)$ then
8:       $\mathbf{v}_j := \mathbf{t}_j$
9:     end if
10:   end for
11: end while
In both cases the values are around the means of previously successful parameter values taken from the whole population. New values are generated each time before mutation and crossover take place. This scheme was adopted and refined in other studies, like the ones in [7, 20]. Also, Zhao et al. [24] recently proposed a scheme that is somewhat similar but differs in the calculation of the mean/location around which new values of $F$ and $CR$ are generated. Moreover, new values of $CR$ and $F$ are generated by, conversely to the approach taken in [23], by Cauchy and Gaussian deviates, respectively.

2.3 Ant Colony Optimisation

Ant colony optimisation [4, 18] is a meta-heuristic in which a population of artificial ants cooperates in the search for solutions to a given optimisation problem. The main inspiration comes from the indirect communication among ants (stigmergy) via the pheromone trails laid by them as they move. A number of ACO algorithms have been proposed and successfully applied to various optimisation problems [13].

Prior to the application of ACO, the problem at hand must be transformed into the problem of finding the best path on a weighted graph (construction graph). Solutions are constructed incrementally by the ants as they traverse the construction graph, whereby each node represents a solution component. The choices of paths to take are made probabilistically, and are influenced by artificial pheromone trails and eventually available heuristic information, which represent the pheromone model. The values associated with the pheromone model (graph weights associated with the arcs or nodes) are dynamically modified. This is achieved through pheromone deposition by selected ants, and a preceding pheromone evaporation.

3. Proposed Parameter Control Scheme

Although all DE parameters are interdependent, most of the approaches from the literature seem to be limited to the control of parameters $F$ and $CR$. This may lead to the conclusion that a well designed scheme can adjust those values in accordance with the set population size $NP$. Hence, the proposed scheme is intended for controlling parameters $F$ and $CR$.

In the proposed scheme the sets $S_F = \{F_i = 0.1 \cdot i: i = 1, \ldots, p=10\}$ and $S_{CR} = \{CR_i = 0.1 \cdot (i-1): i = 1, \ldots, q=11\}$ are given, representing available values of $F$ and $CR$, respectively. Good value pairs $(F_k, CR_l) \in S_F \times S_{CR}$ are selected for and assigned to each vector $v^j$ separately. In order to facilitate the selection and to establish good val-
ues, the problem is modeled as a complete bipartite graph, as illustrated in Fig. 1. Conceptually, artificial ants assigned to each vector, traverse the construction graph. Each node is associated with a value from the set $S_F$, i.e., the set $S_{CR}$. This way, parameters values are selected. The selection is probabilistic (roulette wheel selection), and the probability of selecting some value $F_k \in S_F$ or $CR_l \in S_{CR}$ is

$$p_{F,k} = \frac{\tau_{F,k}}{\sum_{i=1}^{p} \tau_{F,i}}, \quad p_{CR,l} = \frac{\tau_{CR,l}}{\sum_{i=1}^{q} \tau_{CR,i}}, \quad k = 1, \ldots, p, \quad l = 1, \ldots, q, \quad (2)$$

where $\tau_{F,k}$ and $\tau_{CR,l}$ are the artificial pheromone deposited on node $(F,k)$ and $(CR,l)$, respectively. This represents the pheromone model.

The pheromone values are updated after selecting the new generation. First, evaporation takes place, followed by the deposition of new pheromone, that is

$$\tau_{F,k} = (1-\rho)\tau_{F,k} + \sum_{j \in \mathcal{J}} \Delta \tau_{F}^j, \quad k = 1, \ldots, p,$$

$$\tau_{CR,l} = (1-\rho)\tau_{CR,l} + \sum_{j \in \mathcal{J}} \Delta \tau_{CR}^j, \quad l = 1, \ldots, q, \quad (3)$$

where $\rho \in [0,1]$ is the evaporation-rate ($=0.1$), $\mathcal{J}$ is the set of indices of trial vectors that made it into the new generation, $\Delta \tau_{F}^j$ and $\Delta \tau_{CR}^j$ are the pheromone to be deposited for vector $v^j$ on node $(F,k)$ and $(CR,l)$, respectively. A value of 0.1 is deposited on nodes associated with the parameter values selected for $v^j$, whereas a value of 0 is deposited on the remaining nodes. The pheromone values are bounded by $\tau_{min} = 0.1$ and $\tau_{max} = 1$.

4. Experimental Analysis

An experimental analysis was conducted in order to assess the advantages and shortcomings of the proposed parameter control scheme. The analysis was conducted on the benchmark functions prepared for the IEEE CEC2013 [11]. The test suite is composed of 28 functions.

The proposed scheme was incorporated into the canonical DE (denoted as $DE_{(PBPS)}$) for the analysis. A comparison with the canonical DE and the same algorithm incorporating the scheme utilised in DER9 [21] (denoted as $DE_{(DER9)}$), the scheme utilised in aDE [15] (denoted as $DE_{(aDE)}$), and the scheme utilised in SLADE [24] (denoted as $DE_{(SLADE)}$) was performed. This way, only the impact of the parameter control schemes was assessed and a fair comparison was enabled.
4.1 Experiment Setup

For each used algorithm and problem instance, 51 independent algorithm runs were performed. All algorithms were allowed a maximum of $10^4 \cdot d$ function evaluations. Termination occurred as soon as the targeted optimisation error $\Delta f < 10^{-8}$ was reached or the maximum number of function evaluations $\text{NFE}_{\text{max}}$ was performed. Further on, the common population size of $NP = 100$ (used in, e.g., [1, 7, 20]) was used in all algorithms, while $F = 0.5$ and $CR = 0.9$ (used in, e.g., [1, 15, 16, 22] along with $NP = 100$) were used in the canonical DE.

4.2 Results and Discussion

The results obtained on the test functions for $d = 10$ and $d = 30$ are reported in Table 1 and 4, respectively. The tables show the mean and standard deviation (std. dev.). The Wilcoxon signed rank test with a confidence interval of 95% was performed in order to find if the differences (in means) are statistically significant. Accordingly, the symbol (−) indicates a difference in favour of the DE incorporating the proposed scheme (DE_{PBPS}) compared to a given algorithm, the opposite is indicated by the symbol (+), whereas the symbol ($\approx$) indicates an absence of statistical significance. Furthermore, Tables 2 and 5 show the success-rate in reaching the targeted optimisation error for the considered algorithms.
According to the shown results, the DE\(_{(PBPS)}\) algorithm performed in summary better than the other algorithms used in the comparison. This is also evident from Fig. 2. It may be noted that best results on unimodal functions \((f_1 \sim f_5)\) for \(d=10\) were achieved by DE and DE\(_{(aDE)}\), but this was not the case for \(d=30\). Considering multimodal functions \((f_6 \sim f_{20})\), in most cases, the best results were achieved by DE\(_{(PBPS)}\). This is most prominent on problems for \(d = 10\), and slightly less for \(d = 30\). A similar observation can be made in the case of composition functions \((f_{21} \sim f_{28})\). Nonetheless, it must be noted that in just a few cases, the differences in means are very small or virtually nonexistent, but are statistically significant according to the performed test. Those are certainly of no practical importance. Figures 3 and 4 show, for several chosen functions, the average optimisation error \(\Delta f\) in relation with the number of performed function evaluations. The figures suggest a convergence-rate of DE\(_{(PBPS)}\) that is greater or close to the best competitor. Interesting to note, on multimodal and composition functions usually better performance was achieved with a DE algorithm incorporating one of the used parameter control schemes. There are several cases in which the canonical DE got trapped early on in a local optimum, unable to escape it. This hints at one of the benefits parameter control seems to provide.

Another relevant matter are the time complexities of the considered algorithms since the parameter control schemes introduce a certain computational overhead. Tables 3 and 6 provide insight into how much this affects the overall execution times and thus the complexity. The shown data have been obtained according to [11], but on function 13 instead of 14 because most of the algorithms were able to reach the targeted optimisation error on 14 however non on function 13. It must be remarked that the timings have been repeated 25 times in order to accommodate variations in execution times and that the median values (as per \(C\)—approximated complexity) are reported. As may be noted the largest complexity is ascribed to DE\(_{(DER9)}\) and DE\(_{(PBPS)}\). Nonetheless, those are not substantially larger compared to the other used algorithms incorporating parameter control schemes.

Based on the obtained results, it is clear that parameter control can provide an edge over static parameter settings. This seems to be especially the case on more complex problems (multimodal and composition functions, and higher dimensions). In that regard, the algorithms incorporating parameter control schemes provided greater convergence-rates and were principally less susceptible to being trapped in local optima.
5. Conclusion

This paper proposed a scheme for controlling the crossover-rate and scale factor of DE. From a conceptual viewpoint, artificial ants select value pairs from the sets of available parameter values. Their choices are influenced by a pheromone model which is steadily updated. The proposed scheme was incorporated into the canonical DE, and the obtained experimental results suggest good performance.

The main drawback of the proposed scheme is certainly the number of its own parameters. However, they are fixed and need not be changed. It is reasonable to assume that not all are equally important regarding performance, and an analysis in that direction may prove fruitful.

Although promising results were achieved, room for potential improvements is available. Currently, the same amount of pheromone is deposited for each successful parameter value pair, but adjusting that amount according to the achieved improvement may be beneficial. Similarly, the other parameters could be dynamically adjusted as in some successful ACO algorithms. Another direction that should be followed, is the selection of appropriate DE strategies, along with the parameters. In that regard, two possibilities are open, the selection of the crossover and mutation operator separately, or together as one strategy.
### Table 1: Results on functions for $d = 10$

<table>
<thead>
<tr>
<th>$f$</th>
<th>$g$</th>
<th>$f$</th>
<th>$g$</th>
<th>$f$</th>
<th>$g$</th>
<th>$f$</th>
<th>$g$</th>
<th>$f$</th>
<th>$g$</th>
<th>$f$</th>
<th>$g$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
</tr>
<tr>
<td>DE$_{(dE)}$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
</tr>
<tr>
<td>DE$_{(dER9)}$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
</tr>
<tr>
<td>DE$_{(SLADE)}$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
</tr>
<tr>
<td>DE$_{(PBPS)}$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
<td>$0.00E+00$</td>
</tr>
</tbody>
</table>

### Table 2: Success-rate on functions for $d = 10$

| $f$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| DE  | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| DE$_{(dE)}$ | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| DE$_{(dER9)}$ | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| DE$_{(SLADE)}$ | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
| DE$_{(PBPS)}$ | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% | 100% |
Figure 3: Convergence behaviour on several chosen functions for $d=10$.

Table 3: Complexity in terms of execution times on function 13 for $d=10$.

<table>
<thead>
<tr>
<th>Timing</th>
<th>$T_2$</th>
<th>$T_1$</th>
<th>$T_0$</th>
<th>$C = (T_2 - T_1)/T_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE</td>
<td>0.7086 s</td>
<td>0.5930 s</td>
<td>0.0699 s</td>
<td>1.6526</td>
</tr>
<tr>
<td>DE(aDE)</td>
<td>0.7198 s</td>
<td>0.5914 s</td>
<td>0.0699 s</td>
<td>1.3881</td>
</tr>
<tr>
<td>DE(DR9)</td>
<td>0.7388 s</td>
<td>0.5893 s</td>
<td>0.0698 s</td>
<td>2.1399</td>
</tr>
<tr>
<td>DE(DLADK)</td>
<td>0.7466 s</td>
<td>0.5974 s</td>
<td>0.0701 s</td>
<td>2.1296</td>
</tr>
<tr>
<td>DE(PBPS)</td>
<td>0.7441 s</td>
<td>0.5966 s</td>
<td>0.0694 s</td>
<td>2.1202</td>
</tr>
</tbody>
</table>
## Table 4: Experimental results on functions for $d = 30$

<table>
<thead>
<tr>
<th>$f$</th>
<th>DE</th>
<th>DE(aDE)</th>
<th>DE(DER9)</th>
<th>DE(SLADE)</th>
<th>DE(PBPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>std. dev.</td>
<td>mean</td>
<td>std. dev.</td>
<td>mean</td>
</tr>
<tr>
<td>1</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>2</td>
<td>4.30E+05</td>
<td>4.30E+05</td>
<td>2.09E+06</td>
<td>2.09E+06</td>
<td>1.24E+06</td>
</tr>
<tr>
<td>3</td>
<td>1.05E-03</td>
<td>1.05E-03</td>
<td>5.94E-04</td>
<td>5.94E-04</td>
<td>3.72E-04</td>
</tr>
<tr>
<td>4</td>
<td>9.04E+05</td>
<td>9.04E+05</td>
<td>5.94E+05</td>
<td>5.94E+05</td>
<td>3.72E+05</td>
</tr>
<tr>
<td>5</td>
<td>6.33E+03</td>
<td>6.33E+03</td>
<td>3.50E+03</td>
<td>3.50E+03</td>
<td>1.86E+03</td>
</tr>
<tr>
<td>6</td>
<td>7.06E+03</td>
<td>7.06E+03</td>
<td>3.50E+03</td>
<td>3.50E+03</td>
<td>1.86E+03</td>
</tr>
<tr>
<td>7</td>
<td>1.05E+03</td>
<td>1.05E+03</td>
<td>5.94E+02</td>
<td>5.94E+02</td>
<td>3.72E+02</td>
</tr>
<tr>
<td>8</td>
<td>9.04E+02</td>
<td>9.04E+02</td>
<td>5.94E+02</td>
<td>5.94E+02</td>
<td>3.72E+02</td>
</tr>
<tr>
<td>9</td>
<td>6.33E+02</td>
<td>6.33E+02</td>
<td>3.50E+02</td>
<td>3.50E+02</td>
<td>1.86E+02</td>
</tr>
<tr>
<td>10</td>
<td>7.06E+02</td>
<td>7.06E+02</td>
<td>3.50E+02</td>
<td>3.50E+02</td>
<td>1.86E+02</td>
</tr>
<tr>
<td>11</td>
<td>9.04E+02</td>
<td>9.04E+02</td>
<td>3.50E+02</td>
<td>3.50E+02</td>
<td>1.86E+02</td>
</tr>
</tbody>
</table>

## Table 5: Success-rate on functions for $d = 30$

| $f$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |
| DE  | 100% | 0% | 100% | 0% | 41% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| DE(aDE) | 100% | 0% | 100% | 0% | 41% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| DE(DER9) | 100% | 0% | 100% | 0% | 41% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| DE(SLADE) | 100% | 0% | 100% | 0% | 41% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
| DE(PBPS) | 100% | 0% | 100% | 0% | 41% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% | 0% |
Figure 4: Convergence behaviour on several chosen functions for $d=30$.

Table 6: Complexity in terms of execution times on function 13 for $d=30$.

<table>
<thead>
<tr>
<th>Timing</th>
<th>$T_2$</th>
<th>$T_1$</th>
<th>$T_0$</th>
<th>$C = (T_2 - T_1)/T_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE</td>
<td>2.8882 s</td>
<td>2.5804 s</td>
<td>0.0700 s</td>
<td>4.3995</td>
</tr>
<tr>
<td>DE(aDE)</td>
<td>2.9027 s</td>
<td>2.5717 s</td>
<td>0.0699 s</td>
<td>4.7350</td>
</tr>
<tr>
<td>DE(DER9)</td>
<td>2.9882 s</td>
<td>2.5931 s</td>
<td>0.0715 s</td>
<td>5.5254</td>
</tr>
<tr>
<td>DE(SLADK)</td>
<td>2.9089 s</td>
<td>2.5887 s</td>
<td>0.0703 s</td>
<td>4.5523</td>
</tr>
<tr>
<td>DE(PBPS)</td>
<td>2.9598 s</td>
<td>2.5876 s</td>
<td>0.0705 s</td>
<td>5.2804</td>
</tr>
</tbody>
</table>
References


