# How Far Are We From an Optimal, Adaptive DE?

Ryoji Tanabe[1] and Alex Fukunaga[2]

[1] Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency
[2] Graduate School of Arts and Sciences, The University of Tokyo

**Abstract.** We consider how an (almost) optimal parameter adaptation process for an adaptive DE might behave, and compare the behavior and performance of this approximately optimal process to that of existing, adaptive mechanisms for DE. An optimal parameter adaptation process is an useful notion for analyzing the parameter adaptation methods in adaptive DE as well as other adaptive evolutionary algorithms, but it cannot be known generally. Thus, we propose a Greedy Approximate Oracle method (GAO) which approximates an optimal parameter adaptation process. We compare the behavior of GAODE, a DE algorithm with GAO, to typical adaptive DEs on six benchmark functions and the BBOB benchmarks, and show that GAO can be used to (1) explore how much room for improvement there is in the performance of the adaptive DEs, and (2) obtain hints for developing future, effective parameter adaptation methods for adaptive DEs.

## 1 Introduction

Differential Evolution (DE) is an Evolutionary Algorithm (EA) that was primarily designed for continuous optimization [17], and has been applied to many real-world problems [4]. A DE population $\boldsymbol{P} = \{\boldsymbol{x}^1, ..., \boldsymbol{x}^N\}$ is represented as a set of real parameter vector $\boldsymbol{x}^i = (x_1^i, ..., x_D^i)^{\mathrm{T}}$, $i \in \{1, ..., N\}$, where $D$ is the dimensionality of the target problem and $N$ is the population size.

After initialization of the population, for each generation $t$, for each $\boldsymbol{x}^{i,t}$, a mutant vector $\boldsymbol{v}^{i,t}$ is generated from the individuals in $\boldsymbol{P}^t$ by applying a mutation strategy. The most commonly used mutation strategy is the rand/1 strategy: $\boldsymbol{v}^{i,t} = \boldsymbol{x}^{r_1,t} + F_{i,t}\,(\boldsymbol{x}^{r_2,t} - \boldsymbol{x}^{r_3,t})$. The indices $r_1$, $r_2$, $r_3$ are randomly selected from $\{1, ..., N\}$ such that they differ from each other as well as $i$. The scale factor $F_{i,t} \in (0, 1]$ controls the magnitude of the differential mutation operator. Then, the mutant vector $\boldsymbol{v}^{i,t}$ is crossed with the parent $\boldsymbol{x}^{i,t}$ in order to generate a trial vector $\boldsymbol{u}^{i,t}$. Binomial crossover, the most commonly used crossover method in DE, is implemented as follows: For each $j \in \{1, ..., D\}$, if $\mathrm{rand}[0, 1] \leq C_{i,t}$ or $j = j_r$ (where, $\mathrm{rand}[0, 1]$ denotes a uniformly generated random number from $[0, 1]$, and $j_r$ is a decision variable index which is uniformly randomly selected from $\{1, ..., D\}$), then $u_j^{i,t} = v_j^{i,t}$. Otherwise, $u_j^{i,t} = x_j^{i,t}$. $C_{i,t} \in [0, 1]$ is the crossover rate. After all of the trial vectors $\boldsymbol{u}^{i,t}$, $i \in \{1, ..., N\}$ have been generated, each individual $\boldsymbol{x}^{i,t}$ is compared with its corresponding trial vector $\boldsymbol{u}^{i,t}$, keeping the better vector in the population, i.e., if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$, $\boldsymbol{x}^{i,t+1} = \boldsymbol{u}^{i,t}$ for minimization problems. Otherwise, $\boldsymbol{x}^{i,t+1} = \boldsymbol{x}^{i,t}$.

It is well-known that the performance of EAs is significantly influenced by control parameter settings [6, 11], and DE is no exception [4]. Since identifying optimal control parameter values *a priori* is impractical, *adaptive* DE algorithms, which automatically adjust their control parameters online during the search process, have been studied by many researchers. Most of the well-known adaptive DEs [3, 10, 13, 18, 20] automatically adjust the $F$ and $C$ parameters. However, while many adaptive DEs have been proposed, their parameter adaptation methods are poorly understood. Previous work such as [3, 10, 13, 18, 20] only proposed a novel adaptive DE variant and evaluated its performance on some benchmark functions, but analysis of their adaptation methods have been minimal. The situation is not unique to the DE community – Karafotias et al [11] have pointed out the lack of the analysis of adaptation mechanisms in EA. There are several previous work that try to analyze the parameter adaptation method in adaptive DE [3,5,13,16,20]. However, almost all merely visualized how $F$ and $C$ values change during a typical run on functions, and the analysis is limited to qualitative descriptions such as "a meta-parameter of $C$ in adaptive DE quickly drops down to $[0, 0.2]$ after several generations on the Rastrigin function".

In this paper, we consider how an (almost) *optimal* parameter adaptation process might behave, and compare the behavior and performance of this approximately optimal process to that of existing, adaptive mechanisms for DE. We first define what we mean by an optimal parameter adaptation process, and propose a simulation process which can be used in order to greedily approximate the behavior of such an optimal process. We propose GAODE, which applies this methodology to DE and simulates an approximately optimal parameter adaptation process for a specific adaptive DE framework. We compare the behavior of GAODE to typical adaptive DE algorithms on six benchmark functions and the BBOB benchmarks [8], and discuss (1) the performance of current adaptive DE algorithms compared to GAODE, and (2) the implications of these results for developing more effective parameter adaptation method for adaptive DEs.

## 2 The proposed GAO framework for adaptive DEs

First, note that this paper focuses on *parameter adaptation methods* for $F$ and $C$ in adaptive DEs such as jDE [3], EPSDE [13], JADE [20], MDE [10], SHADE [18]. In general, the term "adaptive DE" denotes a complex algorithm composed of multiple algorithm components. For example, "JADE" consists of three key components: (a) current-to-$p$best/1 mutation strategy, (b) binomial crossover, (c) JADE's parameter adaptation method of $F$ and $C$. In this paper we want to focus on analyzing (c) *only*, rather than "JADE", the complex DE algorithm composed of (a), (b) and (c). Therefore, we extracted only (c) from each adaptive DE variant, and generalized it so that it can be combined with arbitrary mutation and crossover methods. This approach is taken in recent work [5, 16].

Due to space limitations, the parameter adaptation methods in jDE, EPSDE, JADE, MDE, and SHADE cannot be described here (see Section **??** in the supplemental materials [1]), but the general framework can be described as follows:

(i) At the beginning of each generation $t$, the $F_{i,t}$ and $C_{i,t}$ values are assigned to each individual $\boldsymbol{x}^{i,t}$. (ii) For each $\boldsymbol{x}^{i,t}$, a trial vector $\boldsymbol{u}^{i,t}$ is generated using a mutation strategy with $F_{i,t}$ and crossover method with $C_{i,t}$. (iii) At the end of each generation $t$, the $F$ and $C$ values used by successful individuals influence the parameter adaptation on the next generation $t+1$, where we say that an individual $i$ is *successful* if $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$.

### 2.1 Optimal parameter adaptation process $\boldsymbol{\theta}^*$

We define the notion of an optimal parameter adaptation process in an adaptive DE. Below, DE-$(a, m)$ denotes an adaptive DE algorithm using $a$ and $m$, where $a$ is a parameter adaptation method, and $m$ is a DE mutation operator. Let $L$ be the number of function evaluations (FEvals) until the search finds an optimal solution. An *adaptation process* $\boldsymbol{\theta}^a_m = (\{F_1, C_1\}, ..., \{F_L, C_L\})^{\mathrm{T}}$ is defined as the series of the $F$ and $C$ parameters generated when DE-$(a, m)$ is executed with some adaptation mechanism $a$ and some DE mutation operator $m$.

For some fixed $m$, an *optimal parameter adaptation process* $\boldsymbol{\theta}^*_m = (\{F^*_1, C^*_1\}, ..., \{F^*_L, C^*_L\})^{\mathrm{T}}$ is defined as an adaptation process which minimizes the expected value of $L$, i.e., there exists no $a'$ such that $E[|\boldsymbol{\theta}^{a'}_m|] < E[|\boldsymbol{\theta}^*_m|]$. In the rest of the paper, we abbreviate this as $\boldsymbol{\theta}^*$. An *optimal parameter adaptation method* $a^*$ is an adaptation method such that $\boldsymbol{\theta}^{a^*}_m = \boldsymbol{\theta}^*$.

$a^*$ and $\boldsymbol{\theta}^*$ are useful notions for analyzing the parameter adaptation methods in adaptive DE. If $\boldsymbol{\theta}^*$ is known for some problem instance $I$, this by definition is a lower bound on the performance of DE-$(a, m)$ (no other adaptation process can have a shorter expected length). This allows quantitative discussions regarding the performance of DE-$(a, m)$ relative to the lower bound, e.g., "DE-(jDE, best/1) is 12.34 times slower than DE-($a^*$, best/1)". We can also use such bounds in order to assess whether further improvements to a certain class of methods are worthwhile, e.g., "DE-(JADE, rand/2) performs worse than CMA-ES [9], but the performance of DE-($a^*$, rand/2) is better than CMA-ES. Therefore, further improvements to the adaptation method $a$ may result in a version of DE-($a$, rand/2) which could outperform CMA-ES."

Besides providing a bound on the performance of DE-$(a, m)$, $\boldsymbol{\theta}^*$ might be useful in guiding the development of more efficient parameter adaptation methods. For example, if for some problem instance $I$, the $F$ values in $\boldsymbol{\theta}^*$ are relatively high at the beginning of the search while they are low at the end of the search, then this suggests that we might be able to improve the performance of DE-$(a, m)$ on problems similar to $I$ by designing $a$ so that the adaptation process of DE-$(a, m)$ more closely resembles of $\boldsymbol{\theta}^*$ for $I$.

However, in practice, it is generally not possible to know $\boldsymbol{\theta}^*$. It is well-known that the appropriate parameter settings depend on the current search situation, and are not fixed values such as $F = 0.5$ and $C = 0.9$, i.e., there are different optimal parameter values as $(\{F^*_1, C^*_1\}, \{F^*_2, C^*_2\}, \{F^*_3, C^*_3\}, ...)$ for each FEvals $(1, 2, 3, ...)$. $\{F^*_l, C^*_l\}$ are also context-dependent, so we can not compute $\{F^*_l, C^*_l\}$ for some time step $l$ in isolation – the search state at $l$ depends on the control parameter settings used in steps $1, ..., l-1$.

**Algorithm 1:** GAODE (the DE with GAO)

```
1  $t \leftarrow 1$, initialize $\boldsymbol{P}^t = \{\boldsymbol{x}^{1,t}, ..., \boldsymbol{x}^{N,t}\}$, $l \leftarrow 1$, $\boldsymbol{\theta}^{\mathrm{GAO}} \leftarrow \emptyset$;
2  while The termination criteria are not met do
3      for $i = 1$ to $N$ do
4          $\boldsymbol{U}^l \leftarrow \emptyset$;
5          for $j = 1$ to $\lambda$ do
6              $F_{l,j} = \mathrm{rand}(F^{\min}, F^{\max}]$, $C_{l,j} = \mathrm{rand}[C^{\min}, C^{\max}]$;
7              The (virtual) trial vector $\boldsymbol{u}^{l,j}$ is generated using an arbitrary mutation
               strategy with $F_{l,j}$ and crossover method with $C_{l,j}$, then $\boldsymbol{u}^{l,j} \rightarrow \boldsymbol{U}^l$;
8          Evaluates the (virtual) trial vectors in $\boldsymbol{U}^l$ by $f$, and select $\boldsymbol{u}^{l,\mathrm{best}}$;
9          $\boldsymbol{u}^{i,t} = \boldsymbol{u}^{l,\mathrm{best}}$, $\boldsymbol{\theta}^{\mathrm{GAO}} \leftarrow \{F_{l,\mathrm{best}}, C_{l,\mathrm{best}}\}$, $l \leftarrow l + 1$;
10         If $f(\boldsymbol{u}^{i,t}) \leq f(\boldsymbol{x}^{i,t})$, $\boldsymbol{x}^{i,t+1} = \boldsymbol{u}^{i,t}$. Otherwise, $\boldsymbol{x}^{i,t+1} = \boldsymbol{x}^{i,t}$;
11     $t \leftarrow t + 1$;
```

### 2.2 Approximating an optimal adaptation process $\boldsymbol{\theta}^*$

As discussed above, $\boldsymbol{\theta}^*$ would be very useful for analyzing the parameter adaptation methods, but it cannot be obtained in practice. Thus, we propose a *Greedy Approximate Oracle method* (GAO) in order to approximate $\boldsymbol{\theta}^*$, and apply the proposed GAO method to DE.

The basic idea is as follows: suppose that in step (i) of the adaptive DE framework described in the beginning of Section 2, we could enumerate all possible parameter settings $\{F, C\}$, and then retroactively select the $\{F, C\}$ pair which results in the best child – this would give us the optimal, 1-step adaptation process. Similarly, the optimal $k$-step adaptation process can be obtained by recursively simulating the execution of the DE for all possible $k$-step adaptation processes and then selecting the best $k$-step process. Of course, the number of possible adaptation processes grows exponentially in the number of steps, so in general, the $k$-step process can not be obtained, and in fact, fully enumerating *all* possible 1-step processes is impractical. We therefore obtain an approximation to the 1-step optimal process by randomly sampling $\{F, C\}$ values.

This is implemented as GAODE, shown in Algorithm 1. For each current FEvals $l$, let us consider that the individual $\boldsymbol{x}^l$ $(= \boldsymbol{x}^{i,t})$ generates the trial vector $\boldsymbol{u}^l$ $(= \boldsymbol{u}^{i,t})$ using parameter settings $\boldsymbol{\theta}^l = \{F_l, C_l\}$ $(= \boldsymbol{\theta}^{i,t})$. The optimal 1-step greedy parameter settings for step $l$ is $\boldsymbol{\theta}^{g^*,l} = \{F_l^{g^*}, C_l^{g^*}\}$, and GAO seeks $\boldsymbol{\theta}^{\mathrm{GAO},l}$, which approximates values of $\boldsymbol{\theta}^{g^*,l}$, by random sampling of $\{F, C\}$ values.

For each $\boldsymbol{x}^l$, $\lambda$ trial vectors $\boldsymbol{U}^l = \{\boldsymbol{u}^{l,1}, ..., \boldsymbol{u}^{l,\lambda}\}$ are generated (Algorithm 1, lines $4 \sim 7$). Parameter values $\boldsymbol{\theta}^{l,j} = \{F_{l,j}, C_{l,j}\}$, $j \in \{1, ..., \lambda\}$ used for generating $\boldsymbol{u}^{l,j}$ are uniformly randomly selected from $(F^{\min}, F^{\max}]$ and $[C^{\min}, C^{\max}]$ respectively (Algorithm 1, line 6). In DE, pseudo-random numbers are used for (a) parent selection in the mutation operator, and (b) the crossover operator. If two different virtual DE configurations which have different $\{F, C\}$ parameter values also use different random numbers for (a) and (b), it complicates the analysis because we cannot determine whether the configuration which generates the better trial vector did so because of its $\{F, C\}$ values or because of the random numbers used in (a) and (b). Therefore, in our experiments, we synchronized the pseudorandom generators for all of the virtual DE configurations so that they all used the same random numbers at both (a) and (b) for generating all trial vectors in $\boldsymbol{U}^l$ – this eliminates the possibility that a virtual DE configuration outperforms another due to fortunate random numbers used for (a) and (b).

The trial vectors in $\boldsymbol{U}^l$ are evaluated according to the function $f$, and the $\boldsymbol{u}^{l,\mathrm{best}}$ with the best (lowest) function value in $\boldsymbol{U}^l$ is selected (Algorithm 1, line 8). The selected $\boldsymbol{u}^{l,\mathrm{best}}$ is treated as $\boldsymbol{u}^l$ $(= \boldsymbol{u}^{i,t})$ of $\boldsymbol{x}^l$ $(= \boldsymbol{x}^{i,t})$. Note that, $\lambda$ times evaluations according to $f$ which are used to select $\boldsymbol{u}^{l,\mathrm{best}}$ in $\boldsymbol{U}^l$ (Algorithm 1, line 8), are *not counted* as the FEvals used in the search – this simulates a powerful oracle which "guesses" $\boldsymbol{u}^{l,\mathrm{best}}$ in one try. $\boldsymbol{\theta}^{l,\mathrm{best}} = \{F_{l,\mathrm{best}}, C_{l,\mathrm{best}}\}$ used for generating $\boldsymbol{u}^{l,\mathrm{best}}$ can be considered a approximation to $\boldsymbol{\theta}^{g^*,l} = \{F_l^{g^*}, C_l^{g^*}\}$, and is stored in $\boldsymbol{\theta}^{\mathrm{GAO}}$ (Algorithm 1, line 9).

Previous work has investigated optimal parameter values in adaptive EAs, especially in Evolution Strategies (ES) community [2,6,7]. For example, the optimal step size $\sigma^*$ in (1+1)-ES on the Sphere function is $\sigma^* = 1.224\|\boldsymbol{x}^* - \boldsymbol{x}\|/D$ [7], where $\|\boldsymbol{x}^* - \boldsymbol{x}\|$ is the Euclidean distance between the optimal solution $\boldsymbol{x}^*$ and the current search point $\boldsymbol{x}$. The optimal mutation rate $p_m$ schedule of $(1+1)$-GA on the one-max problem is also studied by Bäck [2]. While theoretically well-founded, these results are limited to a specific algorithm running on a specific problem, and have also been limited to one parameter value, e.g., $\sigma$ and $p_m$. In contrast, the proposed GAO framework is more general. While we focus on applying GAO to DE for black-box optimization benchmarks in this paper, we believe the GAO approach can be straightforwardly generalized and applied to combinations of various problem domains (e.g., combinatorial problems, single/multi-objective problems, etc.), algorithms (e.g., GA, ES, MOEA, etc.), and parameters (e.g., crossover and mutation rate, crossover method, etc.).

## 3 Evaluating the proposed GAO framework

We compare GAODE, the DE with GAO, to the parameter adaptation methods used by representative adaptive DEs on six benchmark functions. We show that GAO can be used to (1) explore how much room for improvement there is in the performance of the adaptive DEs (Section 3.1), and (2) obtain hints for developing future, effective parameter adaptation methods (Section 3.2).

We used six benchmark functions: Sphere, Ellipsoid, Rotated-Ellipsoid, Rosenbrock, Ackley, Rastrigin functions. The first three are unimodal, and the last three are multimodal (the Rosenbrock function is unimodal for $D \leq 3$). The Rotated-Ellipsoid and Rosenbrock functions are nonseparable, and the (Rotated-) Ellipsoid functions are ill-conditioned functions. For details, see Table **??** in [1].

The dimensionality $D$ of each function was set to 2, 3, 5, 10, and 20. The number of runs per problem was 51. Random number seeds for parts of the DE are synchronized as explained in Section 2.2. Each run continues until either (i) $|f(\boldsymbol{x}^{\mathrm{bsf}}) - f(\boldsymbol{x}^*)| \leq 10^{-8}$, in which case we treat the run as a "success", or (ii) the number of fitness evaluations (FEvals) exceeds $D \times 10^5$, in which case the run is treated as a "failure". $\boldsymbol{x}^{\mathrm{bsf}}$ is the best-so-far solution found during the search

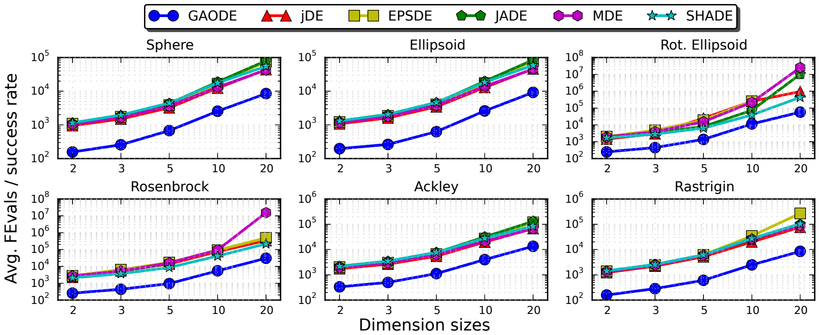


Fig. 1: Comparison of GAODE with the parameter adaptation methods in the adaptive DEs on each function. The horizontal axis represents the dimensionality $D$, and the vertical axis represents the SP1 values. Data with success rate = 0 is not shown.

process, and $\boldsymbol{x}^*$ is the optimal solution of the target problem. Following [9], we used the Success Performance 1 (SP1) metric, which is the average FEvals in successful runs divided by the number of successes, as a performance metric of the DE algorithms. SP1 represents the expected FEvals to reach the optimal solution, i.e., a small SP1 value indicates a fast and stable search.

We used five parameter adaptation methods in the representative adaptive DE variants: jDE [3], EPSDE [13], JADE [20], MDE [10], and SHADE [18]. For details, see Section **??** in [1]. The most basic rand/1/bin operator [17], described in Section 1, was used for all DEs. Following [14], the population size $N$ was set to $5 \times D$ for $D > 5$, and $N = 20$ for $D = 2$ and 3. For each algorithm, we used the control parameter values that were suggested in the original papers as follows: $\tau_F = 0.1$ and $\tau_C = 0.1$ for jDE, $F$-pool= $\{0.4, ..., 0.9\}$ and $C$-pool= $\{0.1, ..., 0.9\}$ for EPSDE, $c = 0.1$ for JADE, and $H = 10$ for SHADE.

In the GAO framework, the parameter generation ranges $(F^{\min}, F^{\max}]$ and $[C^{\min}, C^{\max}]$ have to be set. In preliminary experiments, GAODE failed on some nonseparable functions when these ranges were set to $(0, 1]$ and $[0, 1]$ respectively. We believe this failure is due to small $F$ values, so we also evaluated GAODE with $F^{\min} = 0.4$, where 0.4 is a lowest $F$ value suggested by Rönkkönen et al [15]. Unless explicitly noted, we denote GAODE with the former and later settings as GAODE00 and GAODE04 respectively, and a virtual DE algorithm that is a composition of GAODE00 and GAODE04 as GAODE (GAODE returns the best result obtained by running both GAODE00 and GAODE04). $\lambda$, the number of configurations sampled by GAODE at each individual, was set to 200.

### 3.1 Experiment 1: How much room is there for improvement with adaptive DE algorithms with the rand/1/bin operator?

Figure 1 shows the results of GAODE, jDE, EPSDE, JADE, MDE, and SHADE on the six functions. For GAODE, instead of SP1, we show the lowest FEvals for reaching the optimal solution in the composed results of GAODE00 and GAODE04. The data of GAODE indicates an approximate bound on the performance that can be obtained by an adaptive DE using the rand/1/bin operator.

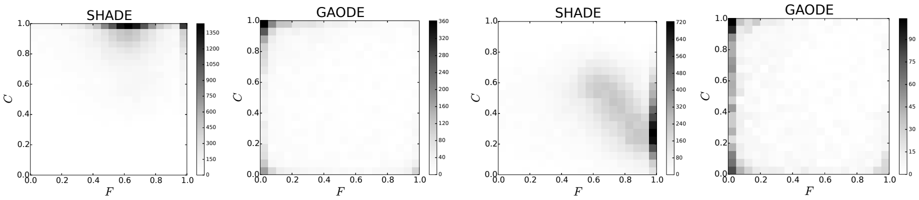
As shown in Figure 1, all runs of EPSDE fail on the Rotated-Ellipsoid function for $D = 20$. JADE also fails on all runs on the Rosenbrock function when $D \geq 10$. MDE can reach the optimal solution on the both functions, but its SP1 values are significantly worse than other methods. Consistent with the results in [16, 20], adaptation methods tend to perform poorly when used with operators that are different from the operators used in the original papers where the adaptation methods were proposed. Although the performance rank among the methods depends on the functions and the dimensionality $D$, jDE and SHADE perform better than other compared methods in almost all cases. However, as shown in Figure 1, jDE and SHADE converge to the optimal solution $4 \sim 20$ times slower than GAODE. This shows that even the best current adaptive methods perform poorly compared to an approximation of a 1-step greedy optimal process (and are therefore even worse compared to a $k$-step optimal process). *Thus, it appears that despite significant progress in recent years, there is still significant room for improvement in parameter adaptation methods for DE.*

### 3.2 Experiment 2: How should we adapt the control parameters?

Let us consider how the behavior of GAODE differs from existing adaptation methods. Figure 2 shows the frequency of appearance of $\{F, C\}$ value pairs during the search process for SHADE and GAODE on the 10-dimensional Rosenbrock and Rastrigin functions. Data from the best run out of 51 runs is shown. The results of jDE, EPSDE, JADE, and MDE can be seen in Figure **??** in [1].

As shown in Figure 2, SHADE frequently generates $F$ and $C$ values in the range $[0.5, 0.7]$ and $[0.9, 1.0]$ on the Rosenbrock function, and $[0.9, 1.0]$ and $[0.1, 0.4]$ on the Rastrigin function respectively. These results are consistent with previous studies for DE [3, 4] and adaptive DEs [3, 20]. On the other hand, GAODE mainly generates $F$ values in the range $[0.0, 0.1]$ on both functions. The $C$ values frequently appear in $[0.0, 0.2]$ and $[0.8, 1.0]$ on the Rastrigin function, and GAODE mainly generates $C$ values in *both* $[0.9, 1.0]$, and $[0.0, 0.1]$ on the Rosenbrock function, i.e., the $C$ values are bimodal. Interestingly, for the both functions, GAODE occasionally generates $F$ and $C$ values in the extreme regions $[0.9, 1.0]$ and $[0.0, 0.1]$ respectively (see bottom right in the figures).

In summary, GAODE frequently generates small $F$ values, and $C$ values in the range $[0, 0.2]$ and $[0.8, 1]$. Although CoBiDE [19], a recently proposed *non-adaptive* DE, generates the $F_{i,t}$ and $C_{i,t}$ values for each $\boldsymbol{x}^{i,t}$ according to a bimodal (two Cauchy) distribution, we are not aware of such a bimodal sampling approach in any previously proposed adaptive method. An adaptive DE algorithm using such sampling method may also perform better than the existing methods [3, 10, 13, 18, 20]. Thus, analysis of the approximate optimal parameter adaptation process obtained by GAO suggests that instead of unimodal sampling procedures implemented in previous adaptation methods, adaptive mechanisms using multimodal sampling may be a promising direction for future work.



(a) Rosenbrock ($D = 10$)

(b) Rastrigin ($D = 10$)

Fig. 2: Frequency of appearance of $\{F, C\}$ value pairs during the search process for SHADE and GAODE on the 10-dimensional (a) Rosenbrock and (b) Rastrigin functions. Darker colors indicate more frequent generation of the corresponding values by the parameter adaptation method.

## 4 Comparing GAODE with state-of-the-art EAs

GAODE, which is an approximate simulation of an optimal, 1-step adaptation process, significantly performs better than the current state-of-the-art parameter adaptation methods for DE using the rand/1/bin operator, as described in Section 3.1 (again, we reemphasize that *GAODE is not a practical algorithm and is for analysis only* – the "performance" of GAODE ignores the $\lambda - 1$ samples which are discarded by GAODE at each iteration). It is interesting to compare GAODE with other state-of-the-art EAs. Here, we compare the adaptive DE variants including GAODE[3] with HCMA [12] and best-2009 on the BBOB benchmarks, consisting of 24 various functions [8]. HCMA, an efficient surrogate-assisted algorithm portfolio, represents the state-of-the-art on the BBOB benchmarks. Best-2009 is a *virtual* algorithm portfolio that is retrospectively constructed from the performance data of 31 algorithms participating in the GECCO BBOB 2009 workshop. Is it possible for an adaptive DE algorithm using the *classical* rand/1/bin operator to be competitive with these methods?

Figure 3 shows the Empirical Cumulative Distribution Function (ECDF) for each algorithm for 24 BBOB benchmark problems ($D = 5, 10, 20$) when maximum FEvals = $D \times 10^4$. The results for each function class and for $D = 2, 3$ can be found in Figures **??** $\sim$ **??** in [1]. As shown in Figure 3, GAODE clearly outperforms jDE, EPSDE, JADE, MDE, and SHADE for all dimensions, in terms of both the quality of the best-so-far solution obtained during the search process and the anytime performance. GAODE also performs significantly better than HCMA and best-2009 for $D \leq 5$. This result suggests that if we can find a parameter adaptation method which performs similarly to the GAODE model, then an adaptive DE algorithm using the classical rand/1/bin could possibly outperform state-of-the-art algorithm portfolios such as HCMA for $D \leq 5$.

[3] The BBOB benchmarks provide 15 instances for each function, i.e., there are $24 \times 15 = 360$ function instances. In this study, we applied GAODE00 and GAODE04 three times for each instance, and only the best result among them is used for GAODE.

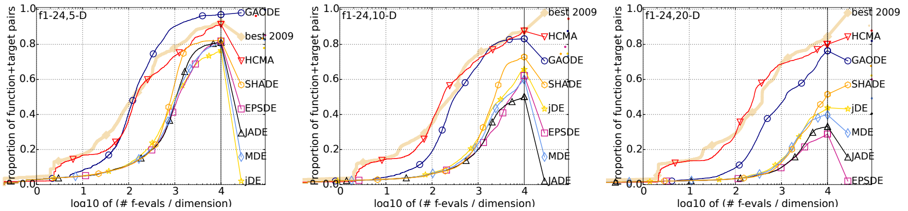


Fig. 3: Comparisons of GAODE with the adaptive DE variants, HCMA, and best-2009 on BBOB benchmarks ($D \in \{5, 10, 20\}$). These figures show the bootstrapped Empirical Cumulative Distribution Function (ECDF) of the FEvals divided by dimension for 50 targets in $10^{[-8..2]}$ for 5, 10, 20 dimensional all functions (higher is better). For details of the ECDF, see a manual of `COCO software` (`http://coco.gforge.inria.fr/`).

On the other hand, when the dimensionality increases, the performance of GAODE degrades compared to HCMA and best-2009. For $D = 20$, GAODE is outperformed by HCMA and best-2009. This may indicate that for high-dimensionality problems, it may not be possible to develop an adaptive DE using the rand/1/bin operator which is competitive with methods such as HCMA. However, this result may be due to the fact that GAODE only simulates an approximately optimal 1-step adaptation process – increasing the number of steps (i.e., a $k$-step optimal process) may result in better results, and is a direction for future work. In addition, different mutation operators (e.g., best/2, current-to-$p$best/1, etc.) may enable significantly better performance for adaptive DEs.

## 5 Conclusion

We proposed a Greedy Approximate Oracle method (GAO) which approximates an optimal parameter adaptation process $\boldsymbol{\theta}^*$. In GAO, $\lambda$ parameter sets are randomly generated for each individual in the population, and the best parameter set with respect to the objective function value is used as a greedily approximated optimal parameter set (the other $\lambda - 1$ sets are discarded and are not counted). We evaluated GAODE, a DE algorithm with GAO, on 6 standard benchmark functions and the BBOB benchmarks [8], and compared it with the parameter adaptation methods in 5 adaptive DE variants. We showed that (1) current adaptive DEs are significantly worse than even an approximate, 1-step optimal adaptation, suggesting that there is still much work to be done in the development of adaptive mechanisms (Section 3.1), and (2) GAO can be used to identify promising directions for developing an efficient parameter adaptation method in adaptive DE (Section 3.2). We also compared GAODE with HCMA [12] and best-2009 on the BBOB benchmarks [8] in Section 4, and showed that a better adaptive mechanism may enable a DE using the classical rand/1/bin operator to achieve state-of-the-art performance.

The proposed GAO framework is a first attempt to approximate the optimal parameter adaptation process, and there is much room for improvement, as discussed in Section 2. In this paper, we applied GAO to the DE with the rand/1/bin operator, and evaluated its performance on single-objective continuous optimization problems. Future work will explore GAO as a general framework that can be applied to analyze the behavior of any adaptive EA (independent of specific operators and problem domains).

## References

1. Supplement. https://sites.google.com/site/tanaberyoji/home/tf-ppsn16-sup.pdf.
2. T. Bäck. Optimal Mutation Rates in Genetic Search. In *ICGA*, pages 2–8, 1993.
3. J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Žumer. Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. *IEEE TEVC*, 10(6):646–657, 2006.
4. S. Das, S. S. Mullick, and P. N. Suganthan. Recent advances in differential evolution – An updated survey. *Swarm and Evolutionary Computation*, 2016 (in press).
5. M. Drozdik, H. E. Aguirre, Y. Akimoto, and K. Tanaka. Comparison of parameter control mechanisms in multi-objective differential evolution. In *LION*, pages 89–103, 2015.
6. A. E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE TEVC*, 3(2):124–141, 1999.
7. N. Hansen, D. V. Arnold, and A. Auger. *Evolution Strategies*. Springer, 2015.
8. N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Technical report, INRIA, 2009.
9. N. Hansen and S. Kern. Evaluating the CMA Evolution Strategy on Multimodal Test Functions. In *PPSN*, pages 282–291, 2004.
10. S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan. An Adaptive Differential Evolution Algorithm With Novel Mutation and Crossover Strategies for Global Numerical Optimization. *IEEE Trans. on SMC. B*, 42(2):482–500, 2012.
11. G. Karafotias, M. Hoogendoorn, and A. E. Eiben. Parameter Control in Evolutionary Algorithms: Trends and Challenges. *IEEE TEVC*, 19(2):167–187, 2015.
12. I. Loshchilov, M. Schoenauer, and M. Sebag. Bi-population CMA-ES agorithms with surrogate models and line searches. In *GECCO Com.*, pages 1177–1184, 2013.
13. R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren. Differential evolution algorithm with ensemble of parameters and mutation strategies. *ASC*, 11:1679–1696, 2011.
14. P. Pošík and V. Klema. JADE, an adaptive differential evolution algorithm, benchmarked on the BBOB noiseless testbed. In *GECCO Com.*, pages 197–204, 2012.
15. J. Rönkkönen, S. Kukkonen, and K. V. Price. Real-Parameter optimization with Differential Evolution. In *IEEE CEC*, pages 506–513, 2005.
16. C. Segura, C. A. C. Coello, E. Segredo, and C. León. On the adaptation of the mutation scale factor in differential evolution. *Opt. Letters*, 9(1):189–198, 2015.
17. R. Storn and K. Price. Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Glo. Opt.*, 11(4):341–359, 1997.
18. R. Tanabe and A. Fukunaga. Success-History Based Parameter Adaptation for Differential Evolution. In *IEEE CEC*, pages 71–78, 2013.
19. Y. Wang, H. Li, T. Huang, and L. Li. Differential evolution based on covariance matrix learning and bimodal distribution parameter setting. *ASC*, 18:232–247, 2014.
20. J. Zhang and A. C. Sanderson. JADE: Adaptive Differential Evolution With Optional External Archive. *IEEE TEVC*, 13(5):945–958, 2009.