

# Patterns of Convergence and Bound Constraint Violation in Differential Evolution on SBOX-COST Benchmarking Suite

Mădălina-Andreea Mitran  
West University of Timisoara  
Timisoara, Romania  
madalina.mitran96@e-uvt.ro

Fabio Caraffini  
Swansea University  
Swansea, UK  
fabio.caraffini@swansea.ac.uk

Anna V. Kononova  
LIACS, Leiden University  
Leiden, The Netherlands  
a.kononova@liacs.leidenuniv.nl

Daniela Zaharie  
West University of Timisoara  
Timisoara, Romania  
daniela.zaharie@e-uvt.ro

## ABSTRACT

This study investigates the influence of several bound constraint handling methods (BCHMs) on the search process specific to Differential Evolution (DE), with a focus on identifying similarities between BCHMs and grouping patterns with respect to the number of cases when a BCHM is activated. The empirical analysis is conducted on the SBOX-COST benchmarking test suite, where bound constraints are enforced on the problem domain. This analysis provides some insights that might be useful in designing adaptive strategies for handling such constraints.

## CCS CONCEPTS

• **Computing methodologies** → **Continuous space search**; • **Theory of computation** → **Continuous optimization**.

## KEYWORDS

differential evolution, bound violation probability, bound constraint handling method, SBOX-COST benchmarking suite, convergence behaviour

### ACM Reference Format:

Mădălina-Andreea Mitran, Anna V. Kononova, Fabio Caraffini, and Daniela Zaharie. 2023. Patterns of Convergence and Bound Constraint Violation in Differential Evolution on SBOX-COST Benchmarking Suite. In *Genetic and Evolutionary Computation Conference Companion (GECCO '23 Companion)*, July 15–19, 2023, Lisbon, Portugal. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3583133.3596410>

## 1 INTRODUCTION

For most real-world optimisation problems, the optimum should be searched in a region limited by a boundary because outside this region the objective function might not be evaluable. Ensuring that the search is conducted within the bounds requires strategies for dealing with the candidate solutions which are outside the bounds,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
*GECCO '23 Companion, July 15–19, 2023, Lisbon, Portugal*

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 979-8-4007-0120-7/23/07...\$15.00  
<https://doi.org/10.1145/3583133.3596410>

strategies which are commonly called bound constraint handling methods (BCHM)<sup>1</sup>. The influence of a BCHM on the performance of different population-based metaheuristics has been studied in several works (Covariance Matrix Adaptation Evolutionary Strategy [3, 28], Particle Swarm Optimisation [12, 14, 20], Differential Evolution [2, 4, 5, 16, 17, 21]). The importance of considering BCHM as a specific algorithmic component that could interfere with the behaviour of the metaheuristic algorithm has recently been acknowledged in [16, 18] where the specification of the used BCHM is considered critical for the unambiguous reproducibility of the results.

Despite the rather large number of BCHMs analysed in the papers mentioned above, the current knowledge on the role of a BCHM is rather limited, and it is not yet clear to which extent a BCHM has also a guidance role in the search process, beyond just bringing back the candidate solutions to the feasible region. The influence of a BCHM on the optimisation process depends, at least, on how frequently it is applied and on how its way of action interferes with the variation operators of the search algorithm. The frequency of applying a BCHM depends on the number of infeasible candidate solutions, which is related, in its turn, to several factors: fitness landscape, location of the optima, problem dimensionality and size of the search neighbourhood defined by the variation operators. On the other hand, some BCHM mechanisms preserve the current population distribution and/or the search direction induced by the variation operators, while other BCHM mechanisms generate elements that, otherwise, could not be easily generated by the variation operators [18].

Due to all these entangled factors, it is not easy to decipher the reasons why a BCHM is effective or not in conjunction with some search algorithms for some optimisation problems. With some exceptions [1, 13], most studies are empirical, based on public benchmarking suites such as CEC [23] and BBOB [11]. Most analyses use CEC test functions, e.g. CEC2006 [15], CEC2017 [4, 16, 19], CEC2020 [16] and only a couple of them use BBOB [5, 18]. This might be explained by the fact that in the original BBOB test suite, for most of the function instances, the optimum is not near the boundary; thus the BCHM is activated only in the first stages of the search, thus having only a transient effect. The newly proposed

<sup>1</sup>BCHM is also referred to as a strategy of dealing with infeasible solutions (SDIS) [18].

SBOX-COST variant of BBOB<sup>2</sup> is characterised by the fact that it allows the generation of function instances with the global optimum arbitrarily close to the boundary, unlike the original BBOB where all functions except linear slope (*F5*) have a wide outside perimeter free of optima on all possible instances. However, it should be mentioned that such modification has not been implemented for functions *F4*, *F5*, *F8*, *F9*, *F19*, *F20*, *F24* in SBOX-COST - their optima locations are kept in  $[-4, 4]^n$  (as in BBOB). Domain of all functions in SBOX-COST is identical to BBOB,  $[-5, 5]^n$ .

The aim of this paper is to investigate the influence of several bound constraints handling methods on the search process specific to Differential Evolution. The empirical analysis is conducted on the SBOX-COST suite and addresses the following questions:

- Which SBOX-COST functions are characterised by a landscape that favours the search near the boundary, i.e. generation of infeasible solution candidates (Section 4)?
- Is it beneficial to preserve, by the bound constraint handling method, some DE search characteristics, e.g. population mean and variance, search direction (Section 5)?
- Is it beneficial to incorporate into the BCHM information from some feasible reference elements, e.g. best element in the population, mean of the population (Section 5)?
- How can a pool of BCHMs be constructed in the context of adaptive selection of the correction (Section 6)?

## 2 FRAMEWORK OF THE ANALYSIS

The analysis is carried out in the context of minimising a function  $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$ , with  $D = [a_1, b_1] \times [a_2, b_2] \times \dots \times [a_n, b_n]$ ,  $a_i < \infty$ ,  $b_i < \infty$ , under the assumption that all candidate solutions should belong to  $D$ . As the behaviour of a BCHM is correlated with the search metaheuristic, we will focus on only one metaheuristic, e.g. Differential Evolution.

### 2.1 Differential Evolution variants

There is an established jargon when it comes to describing the DE framework and its variants [22]. As in an evolutionary algorithm, in DE, a set of  $N$   $n$ -dimensional candidate solutions  $x^j$  ( $j = 1, 2, \dots, N$ ), called individuals, form a population. In the generational loop, for each  $j^{\text{th}}$  individual, called target, a trial individual,  $y^j$ , is constructed by mutation and crossover. Once all trial individuals are constructed, each of them will replace the corresponding target individual if it has a smaller value of the objective function than that of the target individual. Mutation consists in linearly combining existing individuals. Many implementations exist for this operator; see [8] for details. All established mutation operators require a scale factor  $F \in [0, 2]$  and are characterised by the notation  $A/B$ . Here,  $A$  can be the target individual (or current), the best so far solution, a randomly chosen individual from the population (i.e., rand), or a computed vector of the kind current-to-best, etc. To move  $A$  within the search space, a number  $B$  of ‘difference vectors’ (usually  $B = 1$  or  $B = 2$ ) is computed as the ( $F$ -) scaled difference between different randomly chosen individuals and added to  $A$ . Subsequently, a crossover operator  $C$  is fed with the target and mutant individuals to generate a trial. Note that the mutant might contain infeasible components (the mutation in DE is the only operator

that can generate infeasible solutions). As some of these can be transferred to trial during the crossover/recombination process, a BCHM is needed. Therefore, a complete DE variant can be indicated with the notation DE/A/B/C-BCHM. The most widely used crossover operators for DE are referred to as bin and exp strategies, which stand for binomial and exponential, respectively. Note that a prefixed value for the ‘crossover rate’ parameter  $CR \in [0, 1)$  is needed in both crossover strategies. However,  $CR$  assumes a significantly different meaning when  $C=\text{bin}$ , where it determines the probability of exchanging each component and when  $C=\text{exp}$ , where  $n$  and  $CR$  decide the length of a burst of consecutive components to be exchanged [7]. In the latter case, the probability of exchanging the next component follows a geometrical progression, thus decaying exponentially. We refer to [8, 22] for more details on crossover strategies.

The DE/rand/1/bin- $\star$  (with  $\star$  indicating a generic BCHM) configuration represents the original DE framework. Advances in DE have produced modern variants that feature new variation operators, additional operators, hybrid algorithmic components, and, most importantly, various self-adaptation mechanisms for the control parameters  $F$  and  $CR$ , as well as for adjusting the population size  $N$ . Achieving the optimal self-adaptive algorithm is still an open question in heuristic optimisation, and it is key in DE, given its sensitivity to the parameter setting (despite having a reasonably small number of parameters with respect to other heuristics).

The current state-of-the-art is represented by self-adaptive DE frameworks such as the Success-History based Adaptive DE (SHADE) [24], and its successor L-SHADE [25], which builds on the success of SHADE by adding a linear rule to reduce the size of the population during the optimisation process. In this light, while the first algorithm only self-tunes  $F$  and  $CR$ , the second algorithm dynamically changes  $N$  during the optimisation process from an initial suggested value of  $N = 18 \cdot n$ . Apart from this difference, these two advanced DE algorithms use the same ‘current-to-pbest/1’ mutation strategy proposed in [29] for the popular JADE algorithm (where the  $p$ best vector is selected from the  $p\%$  best individuals in the population), as well as a memory-based system to adapt the control parameters. The latter consists of storing both the weighted Lehmer average of successful  $F$  values and the weighted arithmetic average of successful  $CR$  values from previous generations. When needed, such values are randomly selected from the memory. Note that the value of  $p$  is not prefixed as in JADE, but is generated randomly. This mutation operator is usually followed by the binomial crossover. Therefore, the structure of these algorithms can be seen as a DE/current-to-pbest/1/bin algorithm with a memory-based adaptation system for control parameters and, in the case of L-SHADE, a population size reduction mechanism (which is shown to be beneficial in previous DE variants such as [6]). Despite these simple ideas, SHADE and L-SHADE have shown great performances in several benchmark problems compared to other DE algorithms.

### 2.2 Bound constraint handling methods

An infeasible trial individual,  $y$ , contains at least one out-of-bounds component (i.e.  $\exists i \in \{1, \dots, n\} : y_i \notin [a_i, b_i]$ ). A bound constraint handling method transforms all out-of-bounds components (in the case of *component-wise* methods) or all components, including the

<sup>2</sup>accessible through IOHProfiler [10]

**Table 1: Bound constraint handling methods** ( $c_L$  denotes correction for components violating the lower bound;  $c_U$  denotes correction for components violating the upper bound;  $c$  denotes a correction which does not depend on the violated bound).

Method	Corrected component/element	Application level	Type	Characteristics
Saturation	$c_L(y_i) = a_i$ $c_U(y_i) = b_i$	component	deterministic	set on the violated bound
Mirror	$c_L(y_i) = 2a_i - y_i$ $c_U(y_i) = 2b_i - y_i$	component	deterministic	a larger violation leads to a larger correction
Uniform	$c(y_i) \sim \mathcal{U}(a_i, b_i)$	component	stochastic	$c(y_i)$ is independent of the position of $y_i$
Beta	$c(y_i) \sim a_i + \text{Beta}(\alpha_i, \beta_i)(b_i - a_i)$	component	stochastic	aims to preserve population mean and variance
ExpC	Eq.(2)	component	stochastic	it uses a reference point
Vector	$c(y) = \alpha y + (1 - \alpha)R$ $\alpha \in (0, 1)$ as in Eq.(4)	vector	deterministic	it uses a reference point (when $R = x$ it preserves the search direction)

feasible ones (in the case of *vector-wise* methods) in such a way that the resulting individual becomes a feasible one.

There are various ways of transforming an infeasible individual into a feasible one. These are different with respect to what information about the infeasible individual they use (e.g. which bound is violated, how large is the violation of the bound) and with respect to the nature of the strategy of generating values inside the bounding box (e.g. deterministic or stochastic). There are more than fifteen BCHMs that have been included in various empirical studies. Unfortunately, the terminology is not standardized, thus the same BCHM is referred to by using different names (e.g. the BCHM replacing the infeasible component with the closest bound is mentioned as projection [4, 15, 19], clipping [16], saturation [17]).

The BCHMs included in the current analysis are summarized in Table 1, where besides some of the popular strategies (saturation/projection/ clipping, mirror/ reflection, uniform/ reinitialization) a stochastic method is included which is designed based on the particularities of DE population distribution (Beta) and variants of some previously proposed methods: exponentially confined [21] and a vector-wise ‘scaled mutant’ [19]. These new variants are described more in detail in the following.

**2.2.1 Correction based on Beta distribution.** Starting from the result presented in [1] which states that the distribution of the population of feasible DE trial elements (obtained by using DE/rand/1 mutation) can be approximated using a Beta distribution (with values in  $[0, 1]$ ), one can design a BCHM that generates feasible elements following this distribution. The main idea behind the Beta correction is to generate feasible components having the same mean and variance as the current population. To satisfy this condition, the parameters  $\alpha_i$  and  $\beta_i$  ( $i \in \{1, \dots, n\}$ ) of the Beta distribution are calculated based on the mean ( $\text{Mean}(X_i)$ ) and the variance ( $\text{Var}(X_i)$ ) of the current population (computed along the infeasible component), as described in Eq. 1, where  $m_i = (\text{Mean}(X_i) - a_i)/(b_i - a_i)$  and  $v_i = \text{Var}(X_i)/(b_i - a_i)^2$ . In the case where  $m_i = 0$  or  $m_i = 1$  (which could happen if the optimum is at the boundary), then  $m_i$  is replaced with  $\epsilon$  or  $1 - \epsilon$ , respectively ( $\epsilon > 0$  is a small value, for example,  $\epsilon = 0.1$ ).

$$\alpha_i = m_i \left( \frac{m_i(1 - m_i)}{v_i} - 1 \right), \quad \beta_i = \alpha_i \frac{1 - m_i}{m_i} \quad (1)$$

**2.2.2 Exponentially confined correction.** This BCHM was proposed in [21] as a stochastic correction for which the deviation with respect to the violated bound is computed using an exponential distribution. The calculation details are described in Eq. 2 where  $y_i$  denotes the infeasible component,  $r \sim \mathcal{U}[0, 1]$ ,  $r_i^L \sim \mathcal{U}[\exp(a_i - R_i), 1]$ ,  $r_i^U \sim \mathcal{U}[\exp(R_i - b_i), 1]$ , and  $R_i$  is the  $i^{\text{th}}$  component of a feasible reference vector  $R$ .

$$c(y_i) = \begin{cases} a_i - \ln \left( 1 + r(e^{a_i - R_i} - 1) \right) = a_i - \ln r_i^L & y_i < a_i \\ b_i + \ln \left( 1 + (1 - r)(e^{R_i - b_i} - 1) \right) = b_i + \ln r_i^U & y_i > b_i, \end{cases} \quad (2)$$

The variant proposed in [21] corresponds to the case where  $R$  is the target individual. In this paper, we also consider variants when the reference vector is the  $p$ best element (as defined in the SHADE algorithm) or is the midpoint of the current population (mean of the population elements). These three variants are referred to as ExpTarget, ExpBest, and ExpMidpoint, respectively.

**2.2.3 Vector-wise corrections.** In the context of DE, a correction acting on all components (including feasible ones) as illustrated in Eq.3 has been proposed in [19] under the assumption that  $R = 0$  and this null vector belongs to the feasible region.

$$c(y) = \alpha y + (1 - \alpha)R \quad (3)$$

In Eq.3,  $y$  is the infeasible trial individual (corresponding to the target individual,  $x$ ),  $R$  is a reference element from the feasible domain, and  $\alpha = \min_{i=1}^n \alpha_i \in [0, 1]$  with  $\alpha_i$  defined in Eq. 4.

$$\alpha_i = \begin{cases} (R_i - a_i)/(R_i - y_i) & \text{if } y_i < a_i \\ (b_i - R_i)/(y_i - R_i) & \text{if } y_i > b_i \\ 1 & \text{if } a_i \leq y_i \leq b_i \end{cases} \quad (4)$$

In the analysis presented in [4] a variant is used in which  $R$  is the middle point of the bounding box, i.e.  $R = (a + b)/2$ . However, the reference vector  $R$  can be anywhere in the feasible domain, as the corrected individual is on the point where the line that joins the infeasible mutant  $y$  and the reference point intersects the bounding box  $[a_1, b_1] \times \dots \times [a_n, b_n]$ . Once again, the reference vector can be the target individual (VectorTarget), the best (or  $p$ best) individual (VectorBest), the population mean (VectorMidpoint), or an arbitrary individual in the population (not considered in this study). It should be remarked that if  $R$  is the target individual,  $x$ ,

then the search direction is preserved through correction, since  $\cos(y - x, c(y) - x) = \cos(y - x, \alpha(y - x)) = 1$ .

### 3 RELATED WORK

Several authors analyzed the behaviour of different sets of BCHMs in conjunction with various DE variants leading to various conclusions. Some of these studies are shortly reviewed in the following. Arabas et al [2] compared the performance of several strategies, including saturation, toroidal, uniform, and mirroring, and found that the choice of the BCHM could affect the DE performance depending on the characteristics of the problem, such as the position of the optimum and the size of the problem. They observed that saturation and mirroring worked well when the optimum was close to the bounds. They also remarked that for small size problems, the DE performance was not significantly influenced by the used BCHM while for larger size problems, saturation and mirroring were more effective than uniform.

Padhye et al [21] investigated the effects of two groups of BCHMs, depending on whether the correction is performed vector-wise or component-wise, further divided into deterministic and non-deterministic methods on particle swarm optimisation, genetic algorithms, and differential evolution. They argue that deterministic methods such as saturation tend to reduce population diversity, while non-deterministic methods such as uniform may discard valuable information contained in the current population. On the other hand, when the optimum is located near the center of the feasible domain, no significant differences between different BCHMs were reported, while when it is located close to the boundary of the feasible domain, a vector-wise strategy, referred to as inverse-parabolic, performs the best.

The currently most extensive study on the influence of BCHMs on DE performance was presented in [4]. Biedrzycki et al analysed the impact of different correction methods on the dynamics of the population, convergence speed, and global optimisation efficiency. They found that the choice of the method could significantly influence the mean and variance of the mutant population distribution, especially for high-dimensional problems. They also observed that adaptive DE variants were less sensitive to the choice of strategy than non-adaptive ones. The best behaviour was induced by strategies midpoint (the corrected component is in the range defined by the corresponding component of the target/base vector and the violated limit) and mirroring. This motivated us to include in the analysis BCHM variants based on a reference vector identical to the target individual.

In [19], Kreisler et al recommend mirroring and projection to an interior point of the feasible region, followed by midpoint target as strategies which perform well when coupled with DE, in terms of the number of successful runs and the quality of the solutions obtained, in the case of CEC 2017 benchmark. Martinez et al [9] present results on nine different strategies (including midpoint target, mirroring, and projection) applied in the case of four real-world constrained optimisation problems. The main finding is that the effectiveness of the BCHMs depends on the problem, but, in general, projection on the bounds was the most effective.

In [5], Boks et al. investigated the effect of BCHMs on the performance of DE with a wide range of mutation operators and crossover

methods using the BBOB benchmark. They found that no single BCHM was optimal for all configurations and function groups, but recommended reinitialization at least in the case of binomial crossover. Resampling was found to be successful in the majority of the cases considered. This remark motivated us to include in the analysis a method which preserves characteristics of the population distribution, i.e. the proposed Beta correction (see Section 2.2.1).

Addressing the problem that none of the investigated BCHMs is able to deal with particularities of various landscapes and search strategies, in [15], Castillo et al proposed an interesting adaptive scheme for selecting the BCHM that is appropriate for a specific search landscape and search stage.

Unlike most of the previous works, which focus on the competition between the analysed methods, the current paper aims to also identify similarities between BCHMs as well as grouping patterns with respect to search behaviour.

## 4 BOUND CONSTRAINT VIOLATION PATTERNS

The aim of this section is to present an analysis of the similarities between functions belonging to the benchmarking test suites SBOX-COST and BBOB with respect to the number of cases where a BCHM is activated.

### 4.1 Quantification of bound constraint violation

The number of cases where a BCHM is activated and interferes with the search process is related to: (i) the ratio of infeasible trial individuals, i.e. the number of individuals obtained by DE mutation and crossover that contain at least one component which is outside the bounding box, divided by the total number of trial individuals; (ii) the ratio of mutated components (included in trial individuals) that are generated out of the bounding box; this ratio is considered in the following as an approximation of the bound violation probability. In the case of component-wise methods, the amount of BCHM activation depends on the bound violation probability, while in the case of vector-wise methods, it is related to the ratio of infeasible elements.

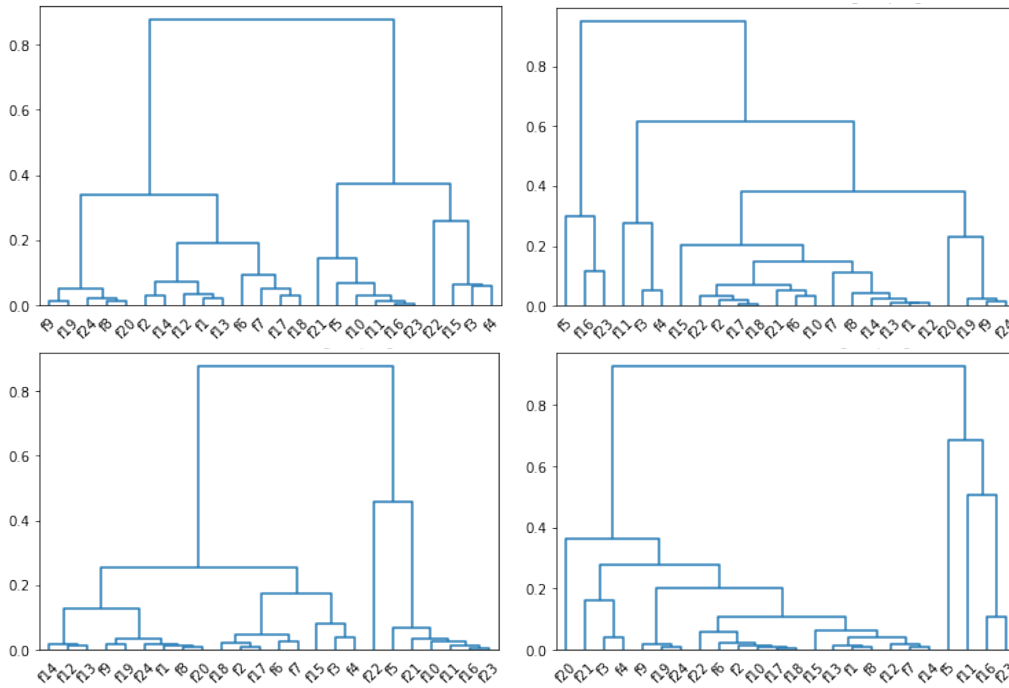
### 4.2 Experimental setup

To identify bound violation patterns and groups of test functions with similar characteristics, we used the following setup:

- *Test functions*: all 24 functions ( $n = 20$ ) from instances 1-5 and 101-110 of the SBOX-COST and BBOB test suites;
- *Methods*: DE/rand/1/bin-none ( $N = 50, F = 0.5, CR = 0.5$ ); L-SHADE-none ( $N_{init} = 18n, p \in [2/N_{crt}, 0.2]$  -  $p_{best}$  is selected between the first  $p\%$  top-ranked individuals, archive size of 6 [25]);
- *Stopping condition*:  $n_{fe} = 10000n$ .

The information collected during the experiments<sup>3</sup> is the ratio of infeasible components (estimation of bound violation probability, obtained as average on five independent runs). Since the purpose of the experiment is to analyse the characteristics of the test functions and not of the BCHMs, all infeasible elements have been discarded, i.e. similar to the death penalty method and the dismiss strategy

<sup>3</sup>The experimental data are available at <https://doi.org/10.5281/zenodo.7836831>



**Figure 1: Dendrograms corresponding to agglomerative clustering with complete linkage on cosine similarities between the bound violation probabilities corresponding to all test functions of the suite. Methods: DE/rand/1/bin (left column) and L-SHADE (right column) on SBOX-COST (top row) and BBOB (bottom row) benchmark suites. Instances: 1-5.**

in [7]. In the case of SBOX-COST this is implicitly ensured by the DE selection, as fitness value of infeasible elements is infinite.

### 4.3 Results

Figure 1 illustrates how the functions of the first 5 instances of the SBOX-COST and BBOB suites can be grouped with respect to the evolution of the probability of bound violation. The grouping is based on the application of agglomerative clustering on the cosine similarities between the vectors obtained by concatenating the bound violation probability values collected during the number of generations corresponding to the allocated  $n_{fe}$  budget. Functions that are typically clustered together, based on the L-SHADE behaviour (see Fig. 2), are: (i)  $\{F5, F16, F23\}$  - the ratio of infeasible components has a nonmonotonous evolution, being around 0.1 after 1000 generations; this would mean that around 10% of the trial components are transformed by a BCHM; (ii)  $\{F19, F20, F24\}$  - for these functions, the bound violation probability decreases quickly to values smaller than  $10^{-2}$ , then, during the remaining generations, an infeasible individual is occasionally generated; (iii)  $\{F3, F4\}$  - after a quick decrease, the bound violation probability increases, reaches a maximum in the range 0.03 – 0.06, then decreases to 0 by 500-th generation; (iv)  $\{F12, F13, F14\}, \{F17, F18\}$  - the bound violation probability is non-zero in the first 100-150 generations and after that no infeasible individuals are generated.

The grouping patterns identified for the bound violation probabilities can also be observed for the performance values (see the error values for all functions and all analysed BCHMs in Fig. 3).

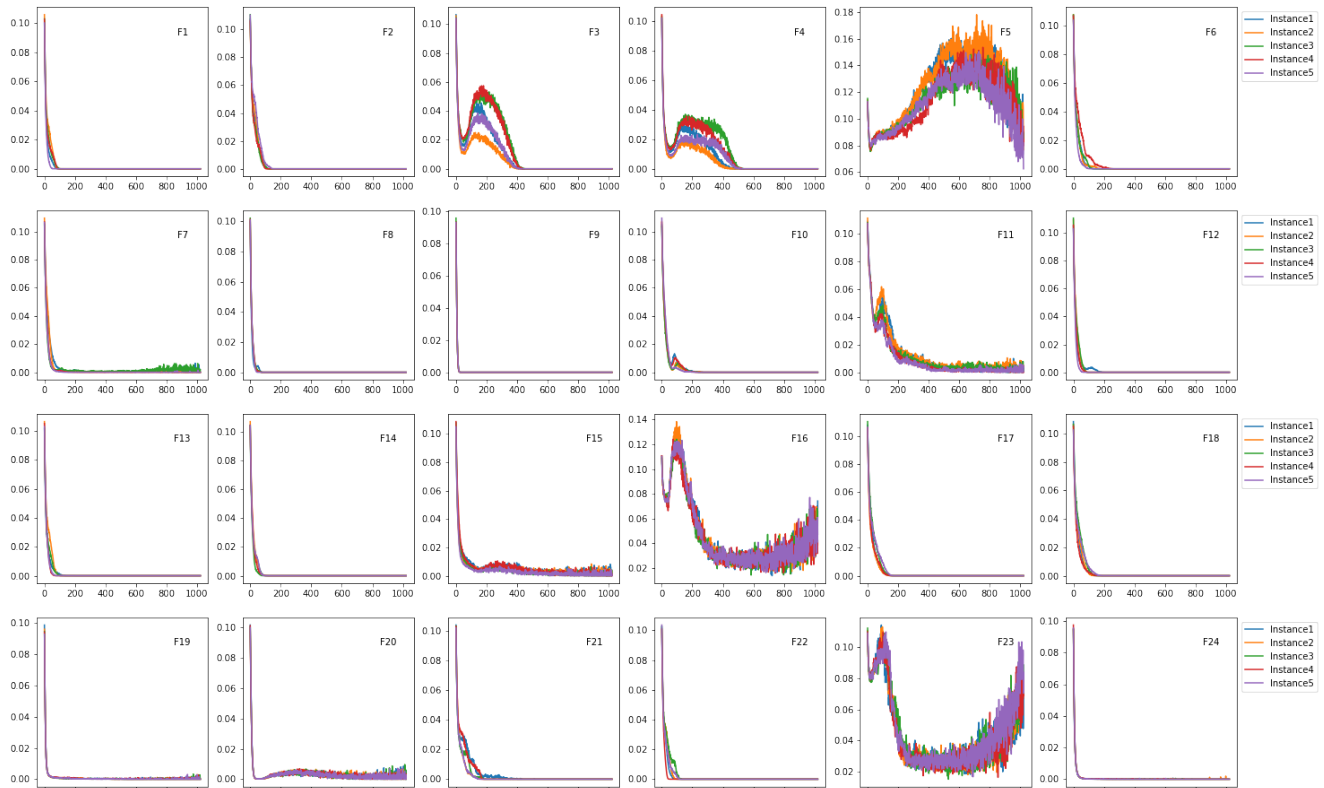
## 5 CONVERGENCE BEHAVIOUR PATTERNS

A desirable behaviour of DE is when the population convergence is synchronised with the convergence to the optimum, e.g. the population concentrates on the optimum. This means that when the population diversity becomes small enough, the best element in the population is close enough to the optimum. The case when the population lost all diversity before reaching the optimum corresponds to premature convergence, while the case when the solution is not localised even if the population is still diverse is caused by the inability of the search process to progress toward the solution (one of these behaviours corresponds to stagnation).

### 5.1 Experimental setup

To identify the type of behaviour induced by BCHMs when combined with L-SHADE (with the configuration specified in Section 4.2) on SBOX-COST functions, we collected results corresponding to the following cases (all reported results are averages over 5 independent runs):

- *Good behaviour (GB)*. The difference between the best element value in the population and the optimal value is smaller than  $10^{-6}$  and the population variance calculated per component is smaller than  $10^{-8}$ . This means that the population converged and the solution was localised.
- *Solution found (SF)*. The error is smaller than  $10^{-6}$ , but the population is not convergent (variance greater than  $10^{-8}$ ).
- *Premature convergence (PC)*. The variance is smaller than  $10^{-8}$  but the error is larger than  $10^{-6}$ .



**Figure 2: Empirical estimation of bound violation probabilities averaged over 30 runs of L-SHADE (with infeasible individuals discarded) in the case of SBOX-COST functions (instances 1-5).**

- *Bad behaviour (BB)*. Both the variance of the population and the error are greater than the thresholds.

## 5.2 Results

Figure 4 illustrates the behavioural differences between the BCHMs analysed together with the test functions. One remark is that the vector-wise BCHMs using the population mean or that using a  $p$ best element are more prone to premature convergence. On the other hand, for the first 14 functions (F1-F14), at least one BCHM leads to the solution, while for the last ten, except for F17, none of the cases is successful.

## 6 TOWARDS ADAPTIVE BCHM

The variability in the behaviour of BCHMs for different fitness landscapes makes their choice in the general setting rather difficult. As in the case of other algorithmic components or control parameters, the idea of adaptive selection seems natural. A first approach in this direction was proposed in [15] where for each infeasible individual, a method is selected from a pool of four BCHMs. The selection is stochastic, being based on a probability distribution adjusted every 25 generation by using the success ratio of each of the BCHMs in the pool. The design of an adaptive BCHM requires: (i) a diverse enough pool of methods; (ii) an adaptive selection strategy. Figure

5 illustrates the cosine similarities between different BCHMs computed using various characteristics (ratio of infeasible components, the evolution of the best-so-far values and the population variance). It can be seen that the results are in line with the particularities of the BCHMs. For example, with respect to the probability of bound violation, Sat is in the same group as vector-wise methods (all of which lead to corrected elements on the boundary). On the other hand Beta correction is in the same group as Dismiss both with respect to the bound violation probability and with respect to population variance (since both aim to preserve the current distribution of the population). As a first step toward the design of an adaptive BCHM, we selected from the groups generated using the best-so-far values (middle dendrogram in Fig. 5) the following methods: VectorBest (because of its atypical behaviour and the best performance for F8, F22, and F23), ExpBest (best performance for F4, F7, F9, F10, F14, and F24), Sat (best performance for F5, F12, F15), VectorTarget (best performance for F11, F17, F18), Beta (best performance for F19, F20, F21). The performance is assessed by ranking, for each test function, all BCHMs using Glicko2 rating [26] from the IOHAnalyzer platform [27] with 50 games. The adaptation of the probability distribution used for selection is based on that proposed in [15]. Preliminary results revealed an average performance of the adaptive variant (see Figure 4, last column in each subfigure), suggesting that more analysis is necessary to design an effective adaptive variant.

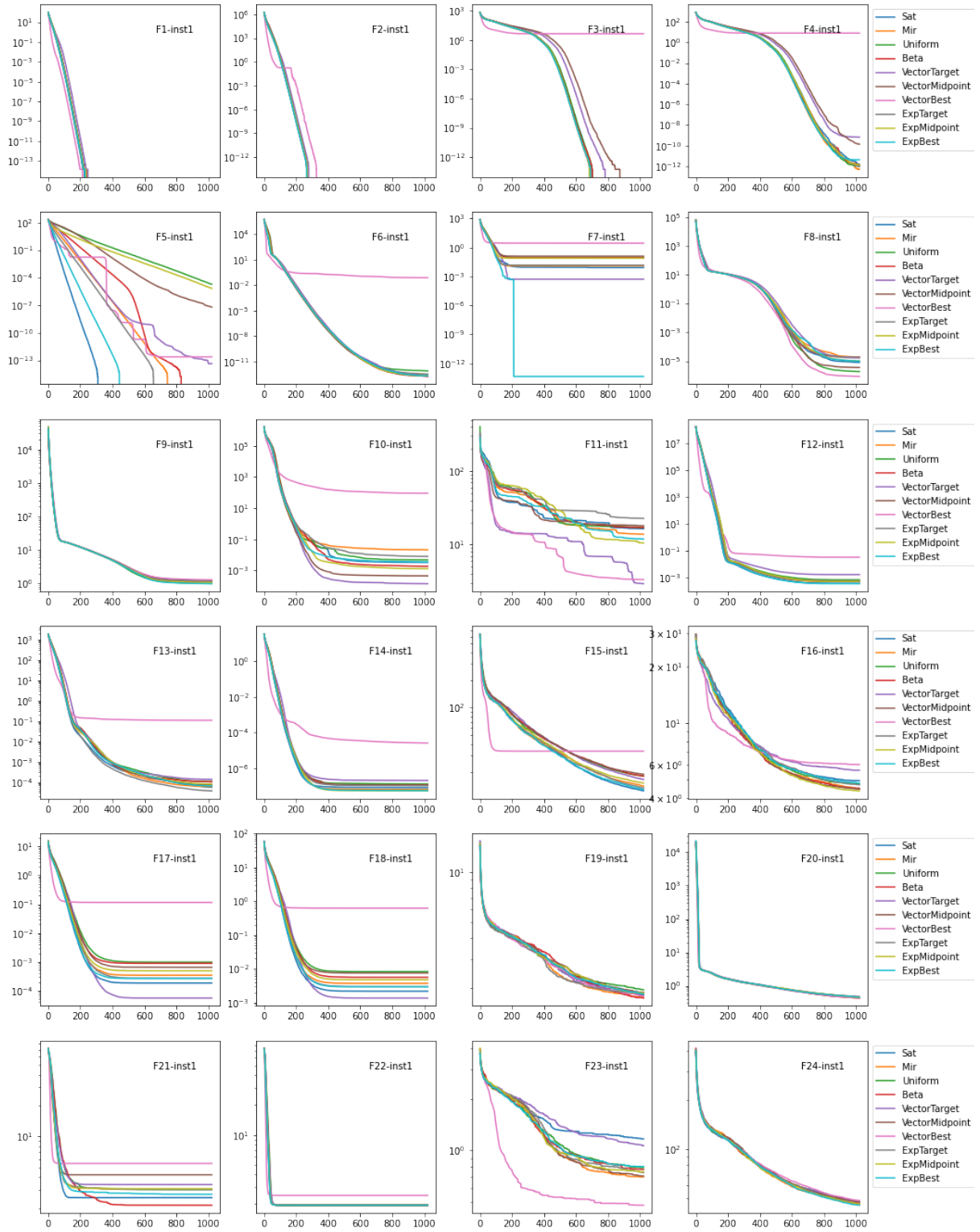
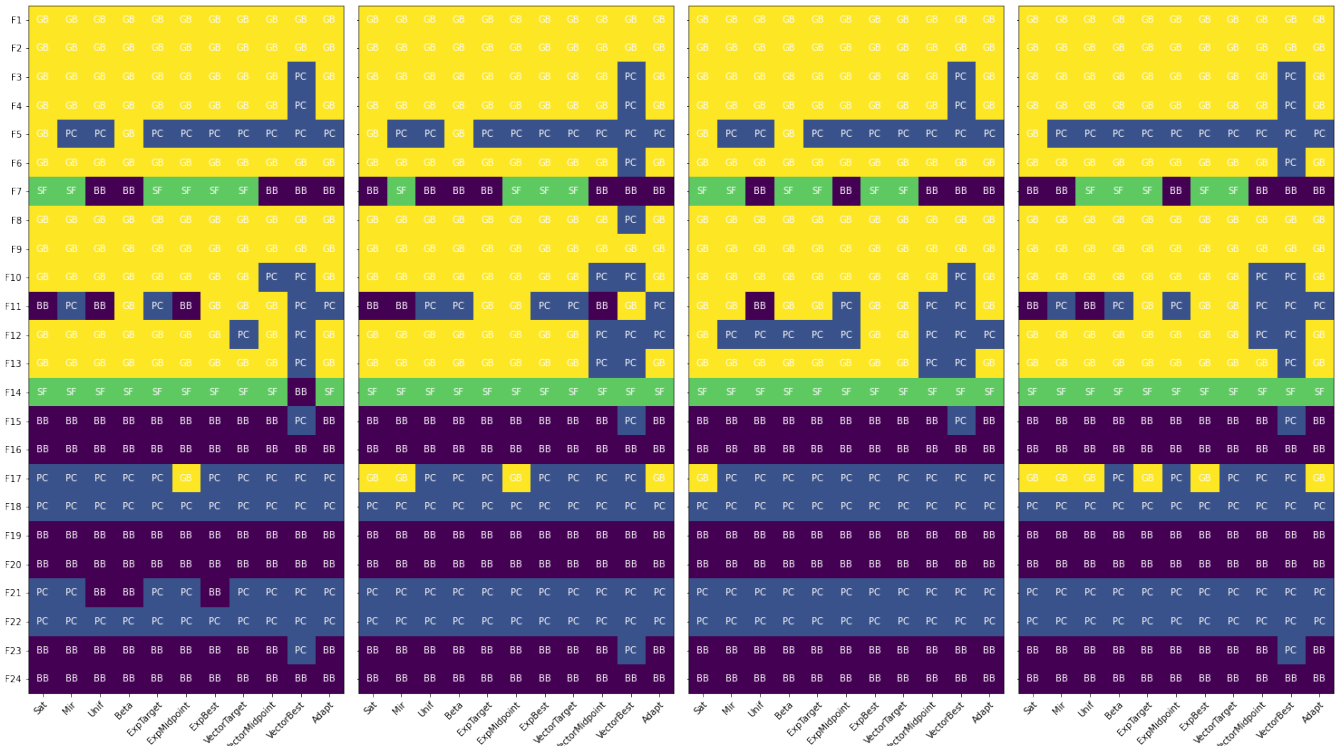
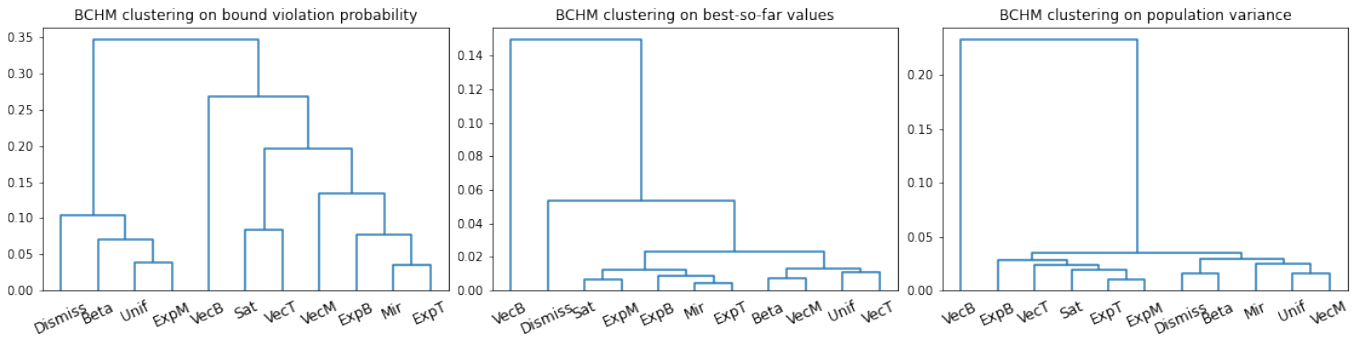


Figure 3: L-SHADE error values (log scale, averaged over 30 runs,  $n_{fe}=5000n$ ). SBOX-COST functions F1-F24, instance 1.



**Figure 4: Comparison of BCHMs with respect to the convergence behaviour of L-SHADE on instances 1-4 (left to right): GB (in yellow) means solution localized and convergent population; SF (in green) means solution localized and non-convergent population; PC (in blue) means convergent population but solution not localized, i.e. premature convergence; BB (in violet) means solution not localized and still diverse population.**



**Figure 5: Dendrograms corresponding to agglomerative clustering with complete linkage on cosine similarities between the vectors corresponding to the evolution of (i) estimated bound violation probability (ratio of infeasible components) (ii) best-so-far values; (iii) population variance. Method: L-SHADE; test suite: SBOX-COST (first 5 instances).**

## 7 CONCLUSIONS

The conducted analysis revealed the following remarks: (i) for most of the SBOX-COST functions, infeasible individuals are generated only in the first generations, leading to a negligible impact of BCHMs (e.g. *F9*); (ii) there are functions (e.g. *F16*, *F23*) with non-zero bound violation probability even after 1000 generations; (iii) the guiding role of the BCHM, ensured by using a reference

vector, is important (ExpBest, ExpTarget, VectBest, VectTarget led to best results in half of the SBOX-COST functions); (iv) preservation of the DE population or search characteristics (specific to Beta and VectTarget) proved to be useful for some functions (*F21*; *F11*, *F17*, *F18*). Finally, when considering BCHMs as algorithmic components of search metaheuristics, we believe that identifying



guidelines for the design of adaptive correction methods is a path to follow.

## REFERENCES

- [1] M. M. Ali and L. P. Fatti. 2006. A Differential Free Point Generation Scheme in the Differential Evolution Algorithm. *J. Glob. Optim.* 35, 4 (2006), 551–572. <https://doi.org/10.1007/s10898-005-3767-y>
- [2] Jaroslaw Arabas, Adam Szczepankiewicz, and Tomasz Wroniak. 2010. Experimental Comparison of Methods to Handle Boundary Constraints in Differential Evolution. 6239 (2010), 411–420. [https://doi.org/doi.org/10.1007/978-3-642-15871-1\\_42](https://doi.org/doi.org/10.1007/978-3-642-15871-1_42)
- [3] Rafał Biedrzycki. 2020. Handling bound constraints in CMA-ES: An experimental study. *Swarm and Evolutionary Computation* 52 (2020), 100627. <https://doi.org/10.1016/j.swevo.2019.100627>
- [4] Rafał Biedrzycki, Jaroslaw Arabas, and Dariusz Jagodzinski. 2019. Bound constraints handling in Differential Evolution: An experimental study. *Swarm Evol. Comput.* 50 (2019). <https://doi.org/10.1016/j.swevo.2018.10.004>
- [5] Rick Boks, Anna V. Kononova, and Hao Wang. 2021. Quantifying the impact of boundary constraint handling methods on differential evolution. In *GECCO '21: Genetic and Evolutionary Computation Conference, Companion Volume, Lille, France, July 10–14, 2021*, Krzysztof Krawiec (Ed.). ACM, 1199–1207. <https://doi.org/10.1145/3449726.3463214>
- [6] Janez Brest and Mirjam Sepesy Maučec. 2008. Population size reduction for the differential evolution algorithm. *Applied Intelligence* 29, 3 (2008), 228–247.
- [7] Fabio Caraffini, Anna V. Kononova, and David Corne. 2019. Infeasibility and structural bias in differential evolution. *Information Sciences* 496 (2019), 161–179. <https://doi.org/10.1016/j.ins.2019.05.019>
- [8] S. Das, Sankha Subhra Mullick, and P.N. Suganthan. 2016. Recent advances in differential evolution – An updated survey. *Swarm and Evolutionary Computation* 27 (2016), 1 – 30. <https://doi.org/10.1016/j.swevo.2016.01.004>
- [9] Sebastián-José de-la-Cruz-Martínez and Efrén Mezura-Montes. 2020. Boundary Constraint-Handling Methods in Differential Evolution for Mechanical Design Optimization. In *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1–8. <https://doi.org/10.1109/CEC48606.2020.9185495>
- [10] Carola Doerr, Hao Wang, Furong Ye, Sander van Rijn, and Thomas Bäck. 2018. IOHprofiler: A Benchmarking and Profiling Tool for Iterative Optimization Heuristics. *arXiv e-prints:1810.05281* (oct 2018). [arXiv:1810.05281](https://arxiv.org/abs/1810.05281)
- [11] Nikolaus Hansen, Anne Auger, Raymond Ros, Olaf Mersmann, Tea Tušar, and Dimo Brockhoff. 2021. COCO: a platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software* 36, 1 (2021), 114–144. <https://doi.org/10.1080/10556788.2020.1808977> [arXiv:https://doi.org/10.1080/10556788.2020.1808977](https://arxiv.org/abs/https://doi.org/10.1080/10556788.2020.1808977)
- [12] Sabine Helwig, Jürgen Branke, and Sanaz Mostaghim. 2013. Experimental Analysis of Bound Handling Techniques in Particle Swarm Optimization. *IEEE Transactions on Evolutionary Computation* 17, 2 (2013), 259–271. <https://doi.org/10.1109/TEVC.2012.2189404>
- [13] Sabine Helwig and Rolf Wanka. 2008. Theoretical Analysis of Initial Particle Swarm Behavior. In *Parallel Problem Solving from Nature – PPSN X*, Günter Rudolph, Thomas Jansen, Nicola Beume, Simon Lucas, and Carlo Poloni (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 889–898.
- [14] Efrén Juárez-Castillo, Héctor-Gabriel Acosta-Mesa, and Efrén Mezura-Montes. 2017. Empirical study of bound constraint-handling methods in Particle Swarm Optimization for constrained search spaces. In *2017 IEEE Congress on Evolutionary Computation*. IEEE, 604–611. <https://doi.org/10.1109/CEC.2017.7969366>
- [15] Efrén Juárez-Castillo, Héctor-Gabriel Acosta-Mesa, and Efrén Mezura-Montes. 2019. Adaptive boundary constraint-handling scheme for constrained optimization. *Soft Comput.* 23, 17 (2019), 8247–8280. <https://doi.org/10.1007/s00500-018-3459-4>
- [16] Tomas Kadavy, Adam Viktorin, Anezka Kazikova, Michal Pluhacek, and Roman Senkerik. 2022. Impact of Boundary Control Methods on Bound-Constrained Optimization Benchmarking. *IEEE Transactions on Evolutionary Computation* 26, 6 (2022), 1271–1280. <https://doi.org/10.1109/TEVC.2022.3204412>
- [17] Anna V. Kononova, Fabio Caraffini, and Thomas Bäck. 2021. Differential evolution outside the box. *Information Sciences* 581 (2021), 587–604. <https://doi.org/10.1016/j.ins.2021.09.058>
- [18] Anna V. Kononova, Diederick Vermetten, Fabio Caraffini, Madalina-A. Mitran, and Daniela Zaharie. 2022. The importance of being constrained: dealing with infeasible solutions in Differential Evolution and beyond. [arXiv:cs.NE/2203.03512](https://arxiv.org/abs/2203.03512)
- [19] Vinicius Kreischer, Thiago Tavares Magalhaes, HJ Barbosa, and Eduardo Krempser. 2017. Evaluation of bound constraints handling methods in differential evolution using the cec2017 benchmark. In *XIII Brazilian Congress on Computational Intelligence*.
- [20] Elre T. Oldewage, Andries P. Engelbrecht, and Christopher Wesley Cleghorn. 2018. Boundary Constraint Handling Techniques for Particle Swarm Optimization in High Dimensional Problem Spaces. In *Swarm Intelligence – 11th International Conference (Lecture Notes in Computer Science)*, Vol. 11172. 333–341. [https://doi.org/10.1007/978-3-030-00533-7\\_27](https://doi.org/10.1007/978-3-030-00533-7_27)
- [21] Nikhil Padhye, Pulkit Mittal, and Kalyanmoy Deb. 2015. Feasibility preserving constraint-handling strategies for real parameter evolutionary optimization. *Computational Optimization and Applications* 62 (2015), 851–890. <https://doi.org/doi.org/10.1007/s10589-015-9752-6>
- [22] Kenneth V. Price, Rainer Storn, and Jouni Lampinen. 2005. *Differential Evolution: A Practical Approach to Global Optimization*. Springer, Berlin, Heidelberg. <https://doi.org/10.1007/3-540-31306-0>
- [23] Ponnuthurai Nagarathnam Suganthan. [n. d.]. Benchmarks for Evaluation of Evolutionary Algorithms. [https://www3.ntu.edu.sg/home/epnsugan/index\\_files/cec-benchmarking.htm](https://www3.ntu.edu.sg/home/epnsugan/index_files/cec-benchmarking.htm). Accessed: 2023-04-10.
- [24] Ryoji Tanabe and Alex Fukunaga. 2013. Success-history based parameter adaptation for Differential Evolution. In *2013 IEEE Congress on Evolutionary Computation*. 71–78. <https://doi.org/10.1109/CEC.2013.6557555>
- [25] Ryoji Tanabe and Alex S Fukunaga. 2014. Improving the search performance of SHADE using linear population size reduction. In *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, IEEE, 1658–1665.
- [26] Niki Vecek, Marjan Mernik, and Matej Crepinsek. 2014. A chess rating system for evolutionary algorithms: A new method for the comparison and ranking of evolutionary algorithms. *Inf. Sci.* 277 (2014), 656–679. <https://doi.org/10.1016/j.ins.2014.02.154>
- [27] Hao Wang, Diederick Vermetten, Furong Ye, Carola Doerr, and Thomas Bäck. 2022. IOHanalyzer: Detailed Performance Analyses for Iterative Optimization Heuristics. *ACM Trans. Evol. Learn. Optim.* 2, 1, Article 3 (apr 2022), 29 pages. <https://doi.org/10.1145/3510426>
- [28] Simon Wessing. 2013. *Repair Methods for Box Constraints Revisited*. Lecture Notes in Computer Science, Vol. 7835. Springer, Berlin, Heidelberg, 469–478. [https://doi.org/10.1007/978-3-642-37192-9\\_47](https://doi.org/10.1007/978-3-642-37192-9_47)
- [29] Jingqiao Zhang and Arthur C. Sanderson. 2009. JADE: Adaptive Differential Evolution With Optional External Archive. *IEEE Transactions on Evolutionary Computation* 13, 5 (2009), 945–958. <https://doi.org/10.1109/TEVC.2009.2014613>