Boundary Constraint Violation in Evolutionary Algorithms

Ing. Tomáš Kadavý, Ph.D.

Doctoral Thesis Summary

Tomas Bata University in Zlín

Tomas Bata University in Zlín
Faculty of Applied Informatics

Doctoral Thesis Summary

Porušování Limitů Argumentů v Evolučních Algoritmech

Boundary Constraint Violation in Evolutionary Algorithms

Author:	**Ing. Tomas Kadavy, Ph.D.**
Degree programme:	Engineering Informatics / P3902
Degree course:	Engineering Informatics / 3902V023
Supervisor:	prof. Ing. Roman Šenkeřík, Ph.D.
Advisor:	doc. Ing. Michal Pluháček, Ph.D.
External examiners:	prof. Ing. Pavel Krömer, Ph.D.
	prof. Ing. Radomil Matoušek, Ph.D.
	doc. RNDr. Petr Bujok, Ph.D.

Zlín, August 2024

Published by **Tomas Bata Univesity in Zlín** in edition **Doctoral Thesis Summary.**
The publication was issued in the year 2024.

Klíčová slova: *Metody kontroly hranic, Metaheuristická optimalizace, Evoluční výpočetní techniky*

Key words: *Boundary Control Methods, Metaheuristic Optimization, Evolutionary Computational Techniques*

Full text of the doctoral thesis is available in the Library of TBU in Zlín.

ISBN 978-80-7678-279-2

ABSTRAKT

V posledních desetiletích se evoluční algoritmy (EA) staly populárními a uznávanými pro svou robustnost a efektivitu v řešení rozmanitých optimalizačních problémů. S narůstajícími výzvami v oblasti umělé inteligence (AI), zejména v kontextu aplikací strojového učení, dochází k nové vlně výzkumu EA. Klíčové aspekty pro další generaci těchto algoritmů zahrnují teoretické základy, analýzy běhu, správné benchmarkovací postupy a detailní zvládání kritických situací, což jsou základní stavební kameny pro dosahování nových úspěchů v AI. Jednou z klíčových výzev je i zvládnutí limitů parametrů optimalizované úlohy, které definují prostor přípustných řešení. I když se dostupné publikace, zabývající se metodami zabraňujících překročení těchto limitů, postupně zlepšují a jejich počet roste, stále je to mnohdy opomíjené téma, které má významný dopad na efektivitu evolučních algoritmů (EA). Tato dizertační práce se zaměřuje na vliv různých protiopatření na výkon evolučních algoritmů (EA). Výzkum začal analýzou základních variant EA, jako jsou PSO (Particle Swarm Optimization), FA (Firefly Algorithm) a SOMA (Self-Organizing Migrating Algorithm). Pozornost se poté přesunula na pokročilejší algoritmy (state-of-the-art), vybrané na základě benchmarkových sad. Studie identifikovala, že integrace účinných protiopatření do návrhu algoritmů může významně ovlivnit jejich pozici v benchmarkových testech. Závěry práce poukazují na významnou problematiku v replikovatelnosti algoritmů, způsobenou nekompletními popisy v publikacích. Tato situace naznačuje potřebu zlepšení v procesu návrhu algoritmů, aby se zvýšila jejich ověřitelnost a udržitelnost.

ABSTRACT

In recent decades, evolutionary algorithms (EA) have become popular and recognized for their robustness and effectiveness in solving a variety of optimization problems. With increasing challenges in the field of artificial intelligence (AI), especially in the context of machine learning applications, a new wave of EA research is emerging. Key aspects for the next generation of these algorithms include theoretical foundations, runtime analyses, proper benchmarking procedures, and detailed handling of critical situations, which are fundamental building blocks for achieving new successes in AI. One of the key challenges is mastering the limits of parameters of the optimized task, which define the space of permissible solutions. Although the available publications dealing with methods to prevent exceeding these limits are gradually improving and increasing in number, it is still a neglected topic that has a significant impact on the effectiveness of evolutionary algorithms (EA). This dissertation thesis focuses on the impact of various countermeasures on the performance of evolutionary algorithms (EA). The research began with an analysis of basic variants of EA, such as PSO (Particle Swarm Optimization), FA (Firefly Algorithm), and SOMA (Self-Organizing Migrating Algorithm). Attention then shifted to more advanced algorithms (state-of-the-art), selected based on benchmark sets. The study identified that integrating effective countermeasures into the design of algorithms could significantly influence their position in benchmark tests. The conclusions of the work point to a significant issue in the replicability of algorithms, caused by incomplete descriptions in publications. This situation indicates the need for improvement in the algorithm design process to enhance their verifiability and sustainability.

TABLE OF CONTENTS

1 INTRODUCTION

In recent decades, the landscape of optimization problem-solving has been significantly transformed by the advent and proliferation of metaheuristic algorithms, which have become indispensable tools for tackling tasks of varying complexity across both real and discrete domains. These methods, encompassing a diverse range of techniques from deterministic approaches [1] like the Newton [2] and gradient methods [3] to Evolutionary Algorithms (EAs) [4] and beyond, offer a robust framework for addressing challenges that are often intractable through traditional deterministic means due to current computational limitations. EAs, a prominent subgroup within the metaheuristic category [5, 6], stand out for their effectiveness in solving complex optimization problems, thereby highlighting the vast potential of Evolutionary Computing Techniques (ECTs).

This dissertation thesis delves into the crucial aspect of metaheuristics design and benchmarking: the Boundary Control Methods (BCM). Given the inherent variability and randomness of metaheuristic algorithms, there is always a possibility that trial solutions might fall outside the predefined parameter boundaries. This occurrence presents a significant challenge as these boundaries reflect the constraints of real-world optimization scenarios or follow the specification of benchmark functions. Such boundaries might be necessitated by practical considerations, such as ensuring the length of a screw remains within positive numerical limits, or by the theoretical constructs underlying benchmark functions, which dictate constraints to maintain mathematical validity in optimization research.

Boundary limits, therefore, form a core component of virtually every optimization task, necessitating the deployment of effective BCMs to handle instances where trial solutions fall outside these acceptable ranges. Over the years, a diverse array of strategies has been developed to address boundary violations, with some methods gaining prominence for their general applicability across a wide spectrum of algorithms, while others are finely tuned to the nuances of specific problems or algorithms.

By focusing on the intricacies of BCMs, this thesis endeavors to offer valuable insights and recommendations to researchers in the field of metaheuristics, potentially informing future directions in the profiling of benchmark testbeds. The ability to effectively manage boundary constraints is pivotal, not only for the integrity of the optimization process but also for the broader applicability and relevance of metaheuristic algorithms in tackling complex optimization tasks, whether they are rooted in theoretical challenges or practical applications. This exploration underscores the importance of BCMs as a critical factor in the design, evaluation, and enhancement of metaheuristic algorithms.

The thesis aims to highlight the significance of BCMs as crucial hyperparameters within metaheuristics. These methods should not only undergo thorough optimization and selection but also be clearly documented within the algorithm's description. Such transparency is essential for ensuring the reproducibility of benchmarking outcomes and enhancing our comprehension of metaheuristic population dynamics.

2 GOALS OF THE DISSERTATION THESIS

During my scientific activities, I discovered that only limited attention is given to the use of Boundary Control Methods (BCM). Therefore, I have set the following goal for this dissertation:

> To document and experimentally verify the impact of using various BCMs on the performance of state-of-the-art evolutionary algorithms. Based on the analysis of the test results, I formulated recommendations for good practices in benchmarking of evolutionary algorithms.

Therefore, the essential steps to goal fulfillment were as follows:

1. **Survey the current state** of Boundary Control Methods (BCMs) used in evolutionary algorithms.
2. **Investigate** influence of various BCMs on the performance of selected evolutionary algorithms.

3. **Conduct experiments** evaluating the impact of BCMs on the performance of state-of-the-art algorithms and results of competitive benchmarking.
4. **Based on the experimental results** draw conclusions and recommendations for good practices in benchmarking.

3 BOUNDARY CONTROL METHODS

The spectrum of optimization problems is vast, often originating from a wide array of real-world challenges. Typically, these problems are transferred to mathematical forms to facilitate easier analysis. With growing interest in metaheuristic optimization [7], there has been a notable increase in the number of benchmark functions and artificial problems created for testing purposes. Each of these benchmark functions used to evaluate metaheuristic optimizers is defined within a specific domain, such as real numbers, positive numbers, or integers, reflecting the varied nature of optimization scenarios. A common feature across both real-world and artificial optimization tasks is the presence of parameter bounds. These bounds may originate from practical limitations in real-world scenarios, such as physical constraints, cost factors, or time limitations.

Due to the inherent randomness in metaheuristic algorithms, there is always a possibility that trial solutions might fall outside the predefined parameter boundaries. This occurrence poses a challenge in effectively solving optimization problems. A typical solution involves checking each newly generated solution to ensure it remains within the acceptable parameter bounds. If a solution is found outside these bounds, an appropriate correction mechanism must be employed to bring it back into the feasible solution space.

3.1 Existing Boundary Control Methods

This subsection presents a consolidated overview of BCMs commonly utilized in research. The versatility of some techniques allows for their adaptation in various forms, while others are specifically tailored for particular optimization algorithms. Additionally, certain methods might be underrepresented in this summary due to their lesser popularity or because they are briefly mentioned in research focusing on different topics.

Clipping

The Clipping Method (also known as saturation) stands out for its simplicity and ease of implementation, often making it one of the first choices in BCMs. In this method, individual solutions $\boldsymbol{x}$ are prevented from crossing the defined boundaries in each dimension. Instead, they are "clipped" to remain within the parameter bounds.

Random

In cases where a trial solution violates the boundary in any dimension, the Random Method generates a new position for the respective dimension. This position is randomly determined between the lower and upper bounds, following a pseudo-random uniform distribution.

Reflection

The Reflection Method mirrors a solution back into the feasible space if it attempts to cross the defined borders. This technique is akin to the reflection behavior of a mirror. For each dimension that violates the boundary, the position of the individual is corrected in a way that reflects it back into the permissible range.

Periodic

The Periodic Method approaches boundary violations by considering an infinite solution space, effectively creating infinite copies of the optimized hyperspace. It employs a mapping technique that brings the individual back into the feasible space using a modulo function. This method ensures that solutions are cyclically repositioned within the acceptable range.

Halving the Distance
As suggested by its name, this method involves halving the distance between the original position and the crossed boundary. Unlike previous techniques, this approach requires tracking the starting position of an individual. It offers a more nuanced adjustment by averaging the boundary and the initial position.

Soft
The Soft Method is unique in that it imposes no immediate restrictions on individuals outside the feasible space, except that their objective function values are not updated until they re-enter the feasible area. Implementing this method can be challenging, as it does not guarantee finite iteration completion without specific algorithm tuning. This approach allows for greater flexibility but requires careful management to ensure algorithm convergence.

4 CHRONOLOGICAL CONTRIBUTIONS TO BOUNDARY CONTROL METHODS

The subsequent sections outline the empirical findings from a set of focused studies undertaken by the author of this doctoral thesis. Initial observations concentrated on the Particle Swarm Optimization (PSO) and its advanced iterations; further studies were conducted on the Self-Organizing Migrating Algorithm (SOMA) and the Firefly Algorithm (FA). These studies aimed to explore the effect of the BCMs on the overall performance of these metaheuristic algorithms.

An additional cornerstone of this exploration is presented in the journal article titled "Impact of Boundary Control Methods on Bound-Constrained Optimization Benchmarking" [8]. This paper focuses on the critical implementation of BCMs and their significant influence on the performance of leading algorithms, as demonstrated in the IEEE CEC competitions of 2017 and 2020.

A further contribution was focused on exploring the frequency of BCM activations. The study investigated how often BCMs were activated during the optimization process across various metaheuristic algorithms.

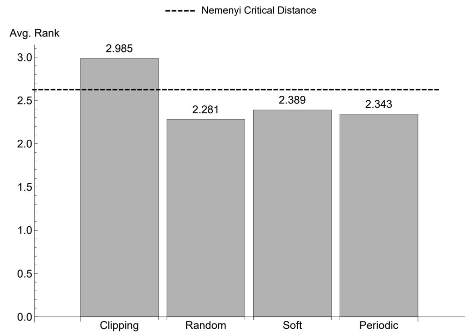
4.1 PSO

The study's results published in 2017 [9] confirmed that the selection of used BCM affects the algorithm's performance. The study compared *Clipping*, *Random*, *Periodic*, and *Soft* methods on the generic version of PSO and the variant called Diversity guided PSO (ARPSO) [10]. The ARPSO algorithm, developed by J. Riget and J. S. Vesterstrøm in 2002, specifically addresses the issue of premature convergence – a notable shortcoming of the traditional PSO algorithm they highlighted in their proposal.

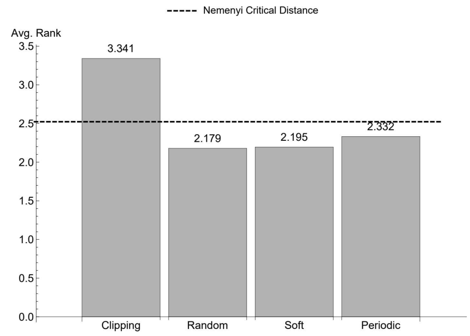
The experiments were conducted on the CEC 2015 benchmark set [11] for dimension sizes 10, 30, and 50. The benchmark encompasses 15 test functions, and every test function was repeated for 51 independent runs. The results were tested for statistical significance using the Friedman Rank test [12] with the significance level $\alpha = 0.05$, accompanied by Nemenyi critical distance (CD) [13].

4.1.1 Key findings

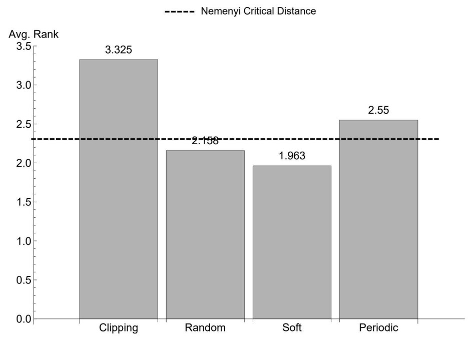
The Friedman rank tests (Fig. 4.1, Fig. 4.2) clearly show that the *Clipping* method negatively affected the overall performance of both PSO and ARPSO algorithms on all dimension sizes. The CD shows which BCMs do not alter the performance of the particular algorithm with statistical significance in comparison with the first-ranked method. These are *Random*, *Periodic*, and *Soft* methods for dimension sizes 10 and 30. For dimension size 50, only *Random* and *Soft* methods perform similarly.



(a) $D = 10$

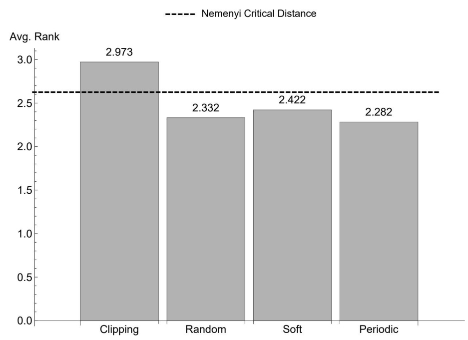


(b) $D = 30$

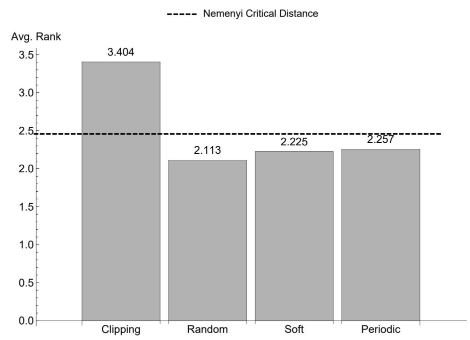


(c) $D = 50$

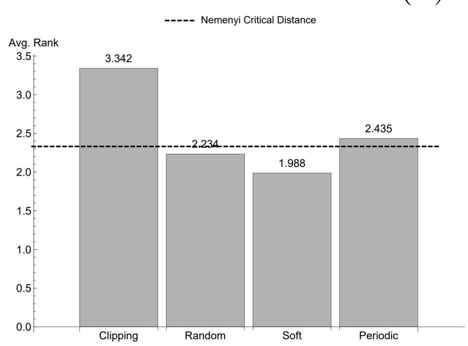
Fig. 4.1 Friedman rank comparison of the Clipping, Random, Periodic, and Soft methods on PSO on benchmark set CEC 2015. [9]



(a) $D = 10$



(b) $D = 30$



(c) $D = 50$

Fig. 4.2 Friedman rank comparison of the Clipping, Random, Periodic, and Soft methods on ARPSO on benchmark set CEC 2015. [9]

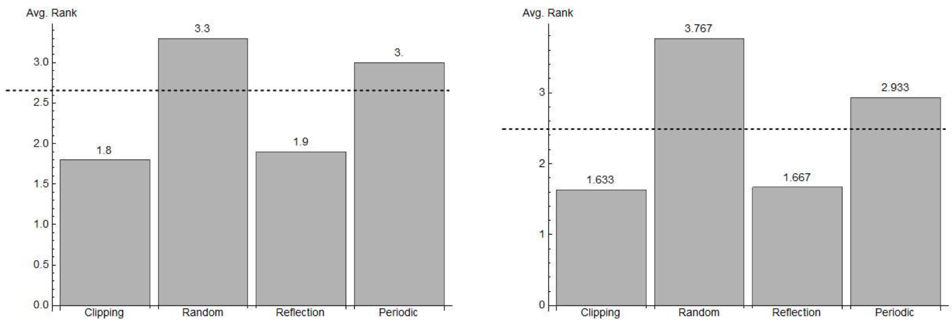
4.2 FA

The subsequent study from 2018 [14] compared *Clipping*, *Random*, *Reflection*, and *Periodic* BCMs on FireFly Algorithm (FA) and on a hybrid of FA and PSO, called Firefly Particle Swarm Optimization (FFPSO) [15]. This hybrid algorithm was introduced in late 2016 by Padmavathi Kora and K. Sri Rama Krishna. The basic idea behind such an approach, according to the authors, is that the new hybrid strategy can share advantages from both algorithms and hopefully eliminate their disadvantages. The main principle remains the same as in the standard FA, but the equation for firefly motion is slightly changed according to PSO movement.

The experiments were performed on the CEC 2017 benchmark set [16], encompassing 30 test functions, and every test function was repeated for 51 independent runs. The tested dimension sizes were 10 and 30. The results were tested for statistical significance using the Friedman Rank test [12] with the significance level $\alpha = 0.05$, accompanied by Nemenyi critical distance (CD) [13].

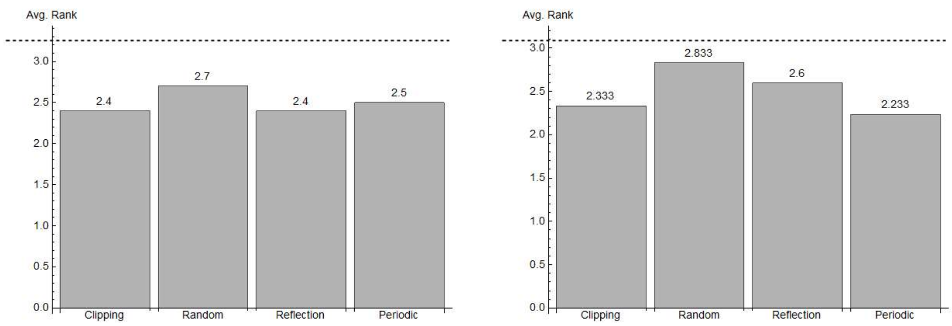
4.2.1 Key findings

Figures 4.3 and 4.4 illustrate the performance rankings for the FA and FFPSO algorithms, respectively, with lower ranks indicating better performance of the BCM used. The significant impact of the selected BCMs is predominantly observed in the canonical FA, while the hybrid FFPSO shows minimal or negligible effects. The Friedman ranks, accompanied by the Nemenyi critical distance (CD) depicted as a dashed line, highlight *clipping* and *reflection* methods as the most effective BCMs for dimension sizes 10 and 30.



(a) $D = 10$ (b) $D = 30$

Fig. 4.3 Friedman rank comparison of the Clipping, Random, Reflection, and Periodic methods on FA on benchmark set CEC 2017. [14]



(a) $D = 10$ (b) $D = 30$

Fig. 4.4 Friedman rank comparison of the Clipping, Random, Reflection, and Periodic methods on FFPSO on benchmark set CEC 2017. [14]

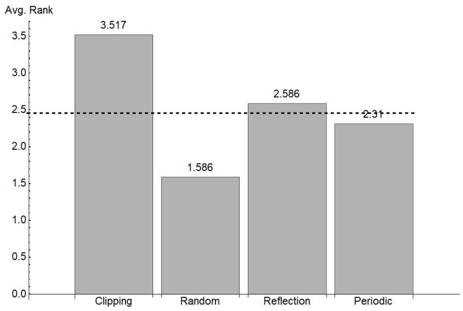
4.3 SOMA

The 2020 study [17] delved into the impact of BCMs on the Self-Organizing Migrating Algorithm (SOMA), specifically its All-To-One and All-To-All variants. SOMA is a metaheuristic optimization technique inspired by the social behavior of individuals within a population moving towards better positions or solutions. It is known for its efficacy in navigating complex optimization landscapes. This investigation was prompted by the notable gap in research concerning the interaction between BCMs and SOMA strategies, a gap that this study aimed to bridge.

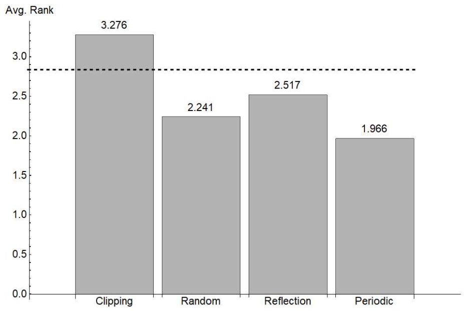
For the experiment, the CEC 2017 benchmark set [16] was chosen, encompassing 30 test functions categorized into unimodal, multimodal, hybrid, and composite groups. The experiment focused on dimension sizes of 10 and 30, adhering to the benchmark's stipulation of a maximum of 10,000 function evaluations per dimension. To ensure robustness, each test function underwent 51 independent trials, with the outcomes subjected to statistical analysis. The analysis utilized the Friedman Rank test [12] to assess statistical significance, rejecting the null hypothesis of equal means at a 5% significance level.

4.3.1 Key findings

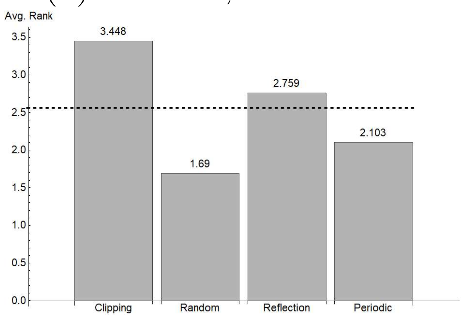
Figure 4.5 displays the Friedman ranking outcomes for the All-To-One and All-To-All SOMA strategies across dimension sizes of 10 and 30, with lower ranks indicating superior performance. The Nemenyi Critical Distance post-hoc test, visually represented by a dashed line from the highest-ranked method, identified the *Random* and *Periodic* methods as notably effective for both strategies, enhancing SOMA's ability to avoid local minima and maintain consistent movement patterns. Conversely, the *Clipping* method, restricting movement to the borders of the feasible space, was found to be the least effective, negatively impacting performance. The *Reflection* method also showed promise by facilitating effective navigation across the search space. Overall, the results highlight the effectiveness of *Random* and *Periodic* methods in introducing beneficial stochastic elements and preserving natural movement dynamics,



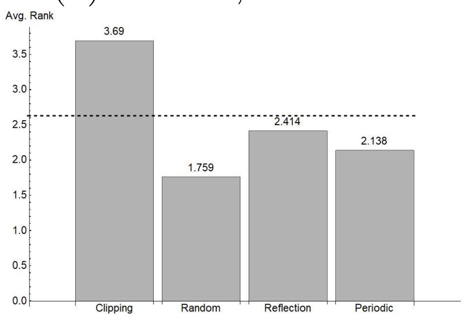
(a) $D = 10$; All-To-One



(b) $D = 10$; All-To-All



(c) $D = 30$; All-To-One



(d) $D = 30$; All-To-All

Fig. 4.5 Friedman rank comparison of the Clipping, Random, Reflection, and Periodic methods on SOMA on benchmark set CEC 2017. [17]

whereas the limitations of the *Clipping* method underscore its potential drawbacks in constrained environments.

4.4 Analyzing the Impact of Boundary Control Methods on Algorithmic Performance

The impulse for the performance analysis study comes from earlier investigations [9, 14, 17] that explored the impact of BCMs on basic metaheuristic algorithms, raising questions about their potential influence on more competitive algorithms, particularly CEC competition winners. This study investigates whether modifications to BCMs could enhance the performance of top algorithms from recent competitions, potentially altering their ranking outcomes.

The study was focused on the three top-ranking participants of two benchmark competitions: CEC17 [16], and CEC20 [18].

4.4.1 Top 3 best-performing algorithms for CEC17

The testbed CEC17 published in 2016 [16] encompasses 30 test functions for dimension sizes of 10, 30, 50, and 100. The following subsections briefly describe the top three performing algorithms according to the official results [19].

EBOwithCMAR was initially proposed for the CEC17 benchmark and successfully obtained the first position among 11 competitors. The hybrid algorithm is based on the Effective Butterfly Optimizer (EBO) and Covariance Matrix Adapted Retreat Phase (CMAR), which improves the local search capability of EBO. The paper [20] does not specify any used BCM; however, upon the analysis of the algorithm code showed that EBOwithCMAR uses two BCMs, *Halving* for EBO and *Clipping* for CMAR.

jSO [21] represents an improved variant of the iL-SHADE algorithm [22] and ranked in second place. The improvement lies predominantly in the new version of the mutation strategy. The jSO uses *Halving* BCM, which is referred in the paper as a “repeat mechanism” without any detailed description or citation.

LSHADE-cnEpSin is the third algorithm used in this study and represents an extension to the LSHADE-EpSin [23], which was ranked as the joint winner in the competition IEEE CEC 2016. The enhancement lies in the ensemble of sinusoidal approaches and covariance matrix learning for the crossover operator. The LSHADE-cnEpSin [24] ranked third in the CEC17 competition and uses *Halving* BCM, which is unfortunately not mentioned in the paper by the authors.

4.4.2 Top 3 best-performing algorithms for CEC20

The CEC20 benchmark [18] was introduced in 2019 and includes 10 test functions for dimension sizes of 5, 10, 15, and 20. Again, the following subsections briefly describe the top three performing algorithms according to the official results [25].

IMODE is a Differential Evolution (DE) based algorithm ranked as the winner in the CEC20 competition. IMODE [26] benefits from multiple differential evolution operators, with more emphasis placed on the best-performing operator. The algorithm employs two BCMs: *Clipping* and *Halving*, selected randomly each time it is used. Unfortunately, BCMs are not mentioned in the paper.

AGSK is the second-best performing algorithm in the CEC20 competition. AGSK [27] is the enhanced version of the Gaining Sharing Knowledge-based algorithm (GSK) [28], which uses adaptive settings to its control parameters. The algorithm utilizes *Halving* BCM; however, the authors did not specify this.

j2020 [29] ranked third place in the competition, and it is based on the two self-adaptive DE algorithms: jDE [30] and jDE100 [31]. The used BCM is *Periodic*, which is described in the paper.

4.4.3 Statistical Evaluation

Friedman rank test was a first step. Each algorithm was tested and evaluated while using different BCMs. The evaluation was performed by Friedman rank test [12], and the results are presented in Table 4.1 for CEC17 and Table 4.2 for CEC20.

Table 4.1 contains more results with statistically significant differences than Table 4.2. The likely reasons are that the CEC20 benchmark contains only 10 test functions and lower dimensionality might cause a lower number of BCM use (see [32] – low dimensionality leads to a lower probability of creation of an infeasible trial solution), and the top ranking algorithms in the competition are robust and similar in the performance.

4.4.4 Competition Scoring System

The motivation for this second test, using CEC scoring system, was to determine if a change of BCM may cause a change in the order of the algorithms.

The CEC scoring system is in detail provided in the technical reports accompanying the CEC benchmarks [16, 18].

4.4.5 Results

This subsection presents the results of both experiments performed for benchmarks CEC17 and CEC20. Each test scenario used a different number of independent runs as defined by the used benchmark. CEC 17 testbed defines 51 independent runs, while CEC20 testbed requires 30 independent runs.

The three approaches were used to analyze and represent the results: Friedman rank test, CEC scoring system, and selection of the best performing BCM variant for the algorithms.

Selection of the BCM was the third and the last step to implement the best-performing BCM variant for the algorithms and check if the final order of the competition would be different. For the CEC17, tables 4.3, 4.4, and 4.5 represent the situations when only one algorithm selects its best variant of the BCM. If the rank is changed, the original rank is shown in parentheses. The most noticeable difference is in Table 4.5, where the LSHADE-cnEpSin obtained the first rank. Table 4.9 then contains the ranks accomplished if all three algorithms had used the best-performing variant of the BCM, and again, the LSHADE-cnEpSin would have achieved the first position.

For the CEC20, the process is the same as for CEC17. The results are presented in Table 4.6 - 4.8 and no change in the algorithms order was observed.

4.4.6 Key findings

The motivation behind the contribution was to establish if the BCM can influence the algorithm performance from the competition results point of view. Thus, raise awareness about the need for careful selection of the BCM, similar to other hyperparameters of the metaheuristic algorithms. The presented results confirm that ill-selected BCM can negatively influence the algorithm's

*Tab. 4.1 Friedman ranks for CEC17. The values in each BCM column represent the Friedman rank in a particular row; the lower the value, the better rank of the algorithm. The p-values are accompanied by the symbol representing different significance levels: * = 0.1, †= 0.05, ** = 0.01, *** = 0.001. The last column CD stands for Nemenyi Critical Difference – if two BCM ranks differ more than CD value, they are significantly different.*

		Default	Clipping	Random	Periodic	Reflection	Halving	p-value	CD
EBOwithCMAR	10D	3.48E+00	3.67E+00	3.41E+00	3.12E+00	3.43E+00	3.88E+00	6.17E–01	1.40E+00
	30D	3.43E+00	3.64E+00	3.52E+00	3.47E+00	3.71E+00	3.24E+00	9.39E–01	
	50D	3.45E+00	3.55E+00	3.62E+00	3.38E+00	3.55E+00	3.45E+00	9.97E–01	
	100D	3.64E+00	4.09E+00	3.43E+00	4.16E+00	2.53E+00	3.16E+00	5.83E–03**	
	Mean	**3.50E+00**	**3.74E+00**	**3.50E+00**	**3.53E+00**	**3.31E+00**	**3.43E+00**		
jSO	10D	Halving	3.64E+00	2.81E+00	2.88E+00	2.62E+00	3.05E+00	4.72E–02†	1.13E+00
	30D		3.22E+00	2.84E+00	3.16E+00	2.97E+00	2.81E+00	7.73E–01	
	50D		3.31E+00	2.66E+00	3.38E+00	2.76E+00	2.90E+00	2.18E–01	
	100D		3.07E+00	2.83E+00	3.69E+00	2.31E+00	3.10E+00	1.52E–02†	
	Mean		**3.31E+00**	**2.78E+00**	**3.28E+00**	**2.66E+00**	**2.97E+00**		
LSHADE-cnEpSin	10D	Halving	3.83E+00	2.38E+00	2.17E+00	3.28E+00	3.34E+00	9.14E–07***	1.13E+00
	30D		3.48E+00	2.59E+00	2.69E+00	3.21E+00	3.03E+00	1.17E–01	
	50D		3.66E+00	2.62E+00	2.55E+00	3.10E+00	3.07E+00	3.40E–02†	
	100D		3.41E+00	2.62E+00	2.97E+00	2.97E+00	3.03E+00	3.94E–01	
	Mean		**3.60E+00**	**2.55E+00**	**2.59E+00**	**3.14E+00**	**3.12E+00**		

*Tab. 4.2 Friedman ranks for CEC20. The values in each BCM column represent the Friedman rank in a particular row; the lower the value, the better rank of the algorithm. The p-values are accompanied by the symbol representing different significance levels: * = 0.1, †= 0.05, ** = 0.01, *** = 0.001. The last column CD stands for Nemenyi Critical Difference – if two BCM ranks differ more than CD value, they are significantly different.*

		Default	Clipping	Random	Periodic	Reflection	Halving	p-value	CD
IMODE	5D	3.25E+00	4.05E+00	3.45E+00	3.75E+00	3.40E+00	3.10E+00	4.60E–01	2.38E+00
	10D	3.40E+00	3.20E+00	3.60E+00	3.70E+00	3.10E+00	4.00E+00	8.63E–01	
	15D	3.95E+00	3.65E+00	2.85E+00	3.35E+00	3.55E+00	3.65E+00	7.44E–01	
	20D	3.90E+00	3.60E+00	3.40E+00	3.90E+00	2.80E+00	3.40E+00	7.19E–01	
	Mean	**3.63E+00**	**3.62E+00**	**3.32E+00**	**3.68E+00**	**3.21E+00**	**3.54E+00**		
AGSK	5D	Halving	3.60E+00	3.10E+00	3.00E+00	3.15E+00	2.15E+00	9.27E–02*	1.93E+00
	10D		3.90E+00	2.30E+00	2.80E+00	2.70E+00	3.30E+00	1.45E–01	
	15D		3.45E+00	2.55E+00	3.25E+00	2.90E+00	2.85E+00	4.84E–01	
	20D		3.80E+00	2.45E+00	3.35E+00	2.55E+00	2.85E+00	1.27E–01	
	Mean		**3.69E+00**	**2.60E+00**	**3.10E+00**	**2.82E+00**	**2.79E+00**		
j2020	5D	Periodic	3.70E+00	2.70E+00	2.90E+00	3.00E+00	2.70E+00	5.11E–01	1.93E+00
	10D		3.70E+00	1.70E+00	3.30E+00	2.90E+00	3.40E+00	1.98E–02†	
	15D		3.80E+00	2.20E+00	3.45E+00	2.65E+00	2.90E+00	9.56E–02*	
	20D		3.70E+00	2.80E+00	3.35E+00	2.70E+00	2.45E+00	3.32E–01	
	Mean		**3.73E+00**	**2.35E+00**	**3.25E+00**	**2.81E+00**	**2.86E+00**		

Tab. 4.3 CEC17 – Score – EBOwithCMAR

Rank	Algorithm	Score 1	Score 2	Score
1 (1)	EBOwithCMAR	5.00E+01	5.00E+01	1.00E+02
2 (2)	jSO	4.96E+01	4.55E+01	9.51E+01
3 (4)	LSHADE_SPACMA	4.66E+01	4.83E+01	9.49E+01
4 (3)	LSHADE-cnEpSin	4.78E+01	4.25E+01	9.03E+01
5 (5)	DES	4.61E+01	4.23E+01	8.84E+01
6 (6)	MM_OED	4.62E+01	3.74E+01	8.35E+01
7 (7)	IDEbestNsize	3.00E+01	2.69E+01	5.69E+01
8 (9)	RB-IPOP-CMA-ES	3.81E+00	3.32E+01	3.70E+01
9 (8)	MOS-CEC2013	1.90E+01	1.78E+01	3.68E+01
10 (10)	MOS-SOCO2011	1.11E+01	1.97E+01	3.08E+01
11 (11)	PPSO	3.94E+00	1.77E+01	2.17E+01
12 (12)	DYYPO	5.96E–01	1.76E+01	1.82E+01
13 (13)	TLBO-FL	2.89E–02	1.68E+01	1.68E+01

Tab. 4.4 CEC17 – Score – jSO

Rank	Algorithm	Score 1	Score 2	Score
1 (1)	EBOwithCMAR	4.89E+01	5.00E+01	9.89E+01
2 (2)	jSO	5.00E+01	4.76E+01	9.76E+01
3 (4)	LSHADE_SPACMA	4.60E+01	4.87E+01	9.47E+01
4 (3)	LSHADE-cnEpSin	4.71E+01	4.35E+01	9.06E+01
5 (5)	DES	4.55E+01	4.33E+01	8.88E+01
6 (6)	MM_OED	4.55E+01	3.81E+01	8.36E+01
7 (7)	IDEbestNsize	2.96E+01	2.76E+01	5.72E+01
8 (9)	RB-IPOP-CMA-ES	3.76E+00	3.38E+01	3.76E+01
9 (8)	MOS-CEC2013	1.88E+01	1.82E+01	3.70E+01
10 (10)	MOS-SOCO2011	1.10E+01	2.01E+01	3.11E+01
11 (11)	PPSO	3.89E+00	1.81E+01	2.20E+01
12 (12)	DYYPO	5.88E–01	1.79E+01	1.85E+01
13 (13)	TLBO-FL	2.84E–02	1.71E+01	1.72E+01

Tab. 4.5 CEC17 – Score – LSHADE-cnEpSin

Rank	Algorithm	Score 1	Score 2	Score
1 (3)	LSHADE-cnEpSin	5.00E+01	5.00E+01	1.00E+02
2 (1)	EBOwithCMAR	4.73E+01	4.92E+01	9.66E+01
3 (2)	jSO	4.73E+01	4.62E+01	9.35E+01
4 (4)	LSHADE_SPACMA	4.45E+01	4.81E+01	9.26E+01
5 (5)	DES	4.40E+01	4.34E+01	8.74E+01
6 (6)	MM_OED	4.40E+01	3.79E+01	8.19E+01
7 (7)	IDEbestNsize	2.86E+01	2.75E+01	5.61E+01
8 (9)	RB-IPOP-CMA-ES	3.63E+00	3.37E+01	3.73E+01
9 (8)	MOS-CEC2013	1.81E+01	1.83E+01	3.64E+01
10 (10)	MOS-SOCO2011	1.06E+01	2.02E+01	3.09E+01
11 (11)	PPSO	3.76E+00	1.82E+01	2.20E+01
12 (12)	DYYPO	5.68E–01	1.80E+01	1.86E+01
13 (13)	TLBO-FL	2.75E–02	1.72E+01	1.73E+01

Tab. 4.6 CEC20 – Score – IMODE

Rank	Algorithm	Score 1	Score 2	Score
1 (1)	j2020	5.00E+01	4.51E+01	9.51E+01
2 (2)	IMODE	2.34E+01	5.00E+01	7.34E+01
3 (3)	AGSK	2.30E+01	4.47E+01	6.77E+01

Tab. 4.7 CEC20 – Score – AGSK

Rank	Algorithm	Score 1	Score 2	Score
1 (1)	j2020	5.00E+01	4.52E+01	9.52E+01
2 (2)	IMODE	2.14E+01	5.00E+01	7.14E+01
3 (3)	AGSK	2.25E+01	4.43E+01	6.68E+01

Tab. 4.8 CEC20 – Score – j2020

Rank	Algorithm	Score 1	Score 2	Score
1 (1)	j2020	5.00E+01	4.36E+01	9.36E+01
2 (2)	IMODE	2.86E+01	5.00E+01	7.86E+01
3 (3)	AGSK	3.06E+01	4.45E+01	7.51E+01

Tab. 4.9 CEC17 – Score

Rank	Algorithm	Score 1	Score 2	Score
1 (3)	LSHADE-cnEpSin	5.00E+01	5.00E+01	1.00E+02
2 (1)	EBOwithCMAR	4.77E+01	5.00E+01	9.76E+01
3 (2)	jSO	4.84E+01	4.58E+01	9.42E+01
4 (4)	LSHADE_SPACMA	4.45E+01	4.78E+01	9.23E+01
5 (5)	DES	4.40E+01	4.30E+01	8.70E+01
6 (6)	MM_OED	4.40E+01	3.81E+01	8.21E+01
7 (7)	IDEbestNsize	2.86E+01	2.76E+01	5.62E+01
8 (9)	RB-IPOP-CMA-ES	3.63E+00	3.37E+01	3.73E+01
9 (8)	MOS-CEC2013	1.81E+01	1.82E+01	3.64E+01
10 (10)	MOS-SOCO2011	1.06E+01	2.02E+01	3.08E+01
11 (11)	PPSO	3.76E+00	1.82E+01	2.20E+01
12 (12)	DYYPO	5.68E–01	1.80E+01	1.86E+01
13 (13)	TLBO-FL	2.75E–02	1.72E+01	1.73E+01

Tab. 4.10 CEC20 – Score

Rank	Algorithm	Score 1	Score 2	Score
1 (1)	j2020	5.00E+01	4.19E+01	9.19E+01
2 (2)	IMODE	3.14E+01	5.00E+01	8.14E+01
3 (3)	AGSK	2.96E+01	4.53E+01	7.49E+01

overall performance.

While the boundary control methods (BCM) are often an overlooked part of the experiment design in metaheuristics benchmarking, the paper aimed to highlight the importance of understanding the BCM as a necessary input for results reproducibility.

To conclude, the findings highlight a significant gap in the reproducibility of results among competition entries, primarily due to the omission of information about the utilized BCM. This oversight not only hampers the reproducibility of results but also overlooks potential performance enhancements that could be achieved by focusing on BCMs. Moreover, it has been observed that algorithms, particularly from the Differential Evolution (DE) family, implemented in Matlab, often rely on the same or similar libraries for BCM. These libraries commonly include the implementation of the *halving* BCM, likely influencing researchers' preference for its use due to its ready availability.

4.5 Exploring the Frequency of BCMs Activation

This study delves into the relationship between the frequency of BCM activation and various problem characteristics, such as dimensionality and fitness landscape, analyzing each dimension separately. The focus was on evaluating the top three algorithms from the CEC20 competition (AGSK, IMODE, and j2020) using the competition's benchmark set.

The activation frequency of BCMs was assessed for each function in the benchmark set across the top three performing algorithms. The number of function evaluations (FEs) and the population size were standardized according to competition rules to align with original benchmarking conditions. Each experimental setting was conducted 30 times to ensure the reliability of the results, and the average number of BCM activations was calculated for each problem dimension. Dimension sizes of 5 and 10 were specifically examined.

To offer a comprehensive understanding of BCM activation patterns, activations were observed in each problem dimension separately. Through this analytical approach, insights into the intricacies of BCM activation frequency and its relationship with the problem's dimensionality were gained. Furthermore, it was examined whether the BCM activation frequency differed significantly across the functions in the benchmark set, yielding valuable information for algorithm designers and researchers.

The results are presented using stacked graphs, a highly effective visualization technique that provides several advantages for displaying and interpreting data. Stacked graphs are utilized for a clear and concise representation of multiple datasets within a single, unified plot. In the context of this research, stacked graphs are employed to effectively illustrate the activation frequency of various BCMs in relation to problem dimensionality and fitness landscape. Additionally, the same technique is used to visualize differences in BCM activation rate among the distinct problem dimensions.

Figures Fig. 4.6 display stacked graphs of the average number of BCM activations for three different algorithms (IMODE, AGSK, and j2020) across six different BCMs. In each figure, each column represents an algorithm, with BCMs stacked on top of each other to form a bar chart. The x-axis displays the six different BCMs, and the y-axis indicates the average number of activations (over 30 runs), for each algorithm-dimension combination. It is important to note that the values for each algorithm in the stacked graph are represented as the sum of BCM activations in different dimensions. Different colors in each column represent the different algorithms.

Figures Fig. 4.7 show different stacked graphs; these depict a particular BCM activation on a selected test function, dimension size, and algorithm. The x-axis represents the function evaluations (FEs) of the algorithm, and the y-axis shows the average number of activations (over 30 runs) for each problem dimension.

From the analysis of the first group of graphs (Fig. 4.6), it is concluded that the AGSK algorithm contributes to BCM activation the most across all test functions. It is also concluded that the particular BCM affects the AGSK algorithm the most. Moreover, the number of BCM activations for algorithms j2020 and IMODE is observed to remain fairly consistent across all BCM variants.

All figures, codes of the examined algorithms, and their results are accessible in high resolution from the designated webpage [1].

4.5.1 Key findings

Based on the findings that reveal variations in BCM activation rates across different algorithms, test problems (fitness landscapes), and problem dimensions, as well as differences in activation rates for each BCM on the same problem, several important conclusions are drawn:

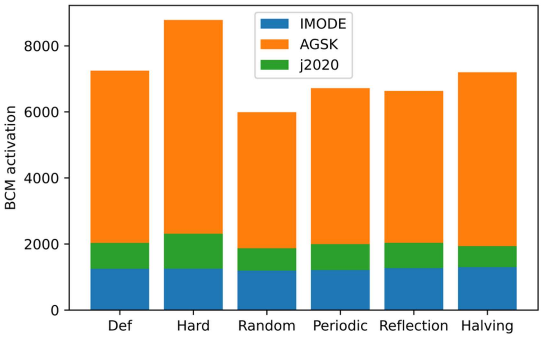
Algorithm-specific characteristics: Differences in BCM activation rates among different algorithms are observed, indicating unique characteristics for each algorithm that influence how boundary constraints are handled. This underscores the importance of selecting appropriate algorithms for specific problem types and suggests potential improvements through a better understanding of BCM behavior.

Problem-dependent activation rates: Variations in BCM activation rates across different fitness landscapes suggest that the effectiveness of BCMs is strongly dependent on the characteristics of the test problems. This necessitates careful consideration of problem-specific properties when designing or selecting BCMs to ensure optimal performance.

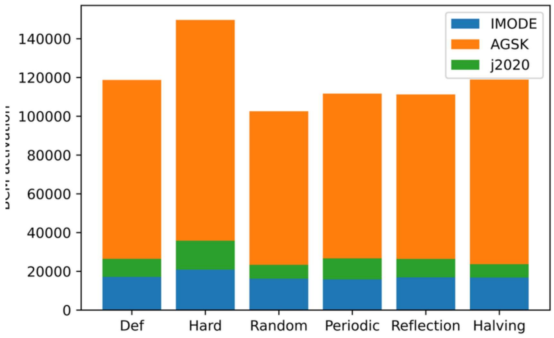
Dimensionality impact: The variability of BCM activation rates among dimensions of the same problem highlights the influence of problem dimensionality on the complexity of boundary constraints. This emphasizes the need to consider dimensionality's impact on BCM activation patterns when designing or adapting algorithms for high-dimensional optimization challenges.

Tailoring BCMs for improved performance: Observed differences in activation rates for each BCM on the same problem suggest that a universal solution for boundary control does not exist. By understanding these varia-

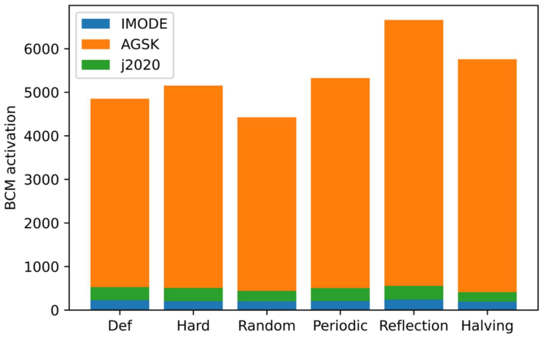
[1] https://go.fai.utb.cz/2023workshop



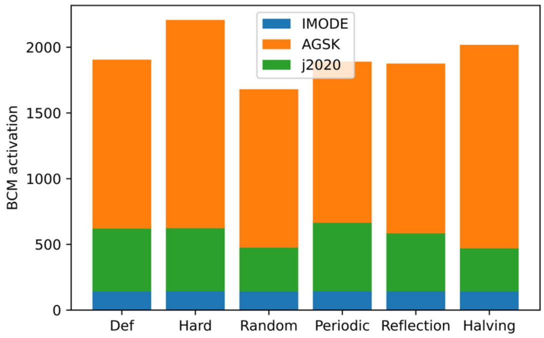
(a) f_1



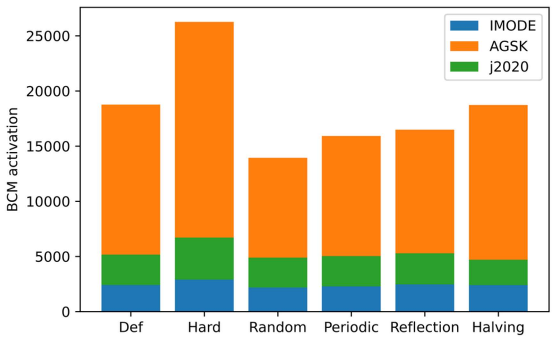
(b) f_2



(c) f_3

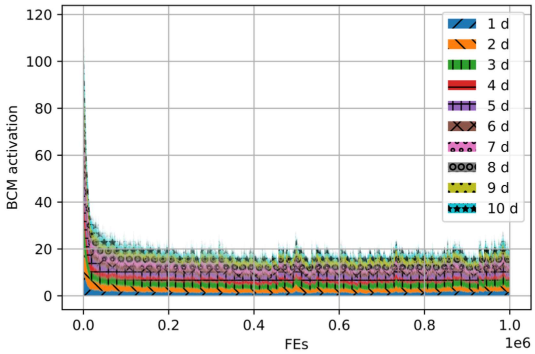


(d) f_4

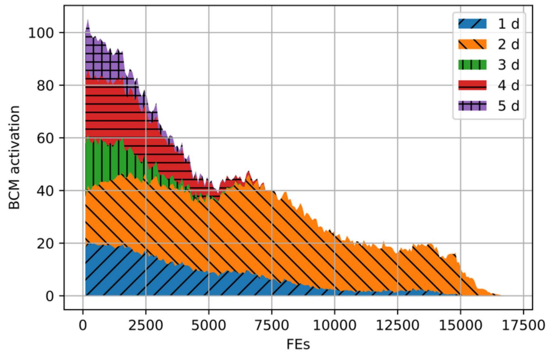


(e) f_5

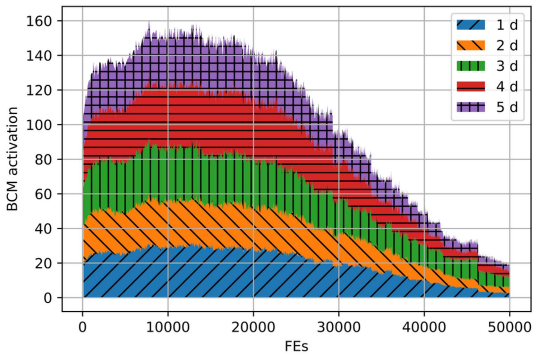
Fig. 4.6 Total BCM activation for $dim = 5$*. [33]*



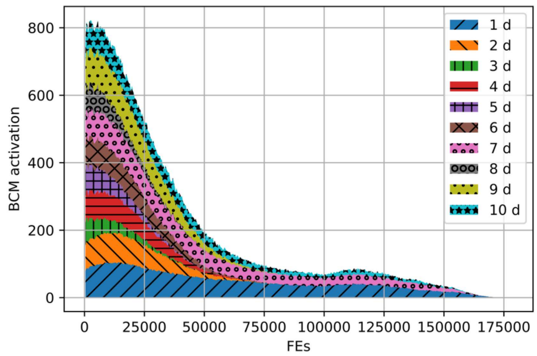
(a) j2020; *Periodic*; f_2; $dim = 10$



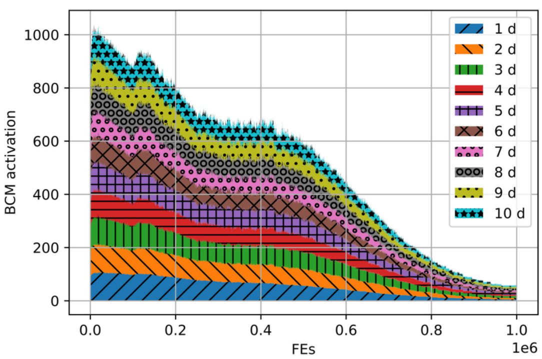
(b) AGSK; *Clipping*; f_1; $dim = 5$



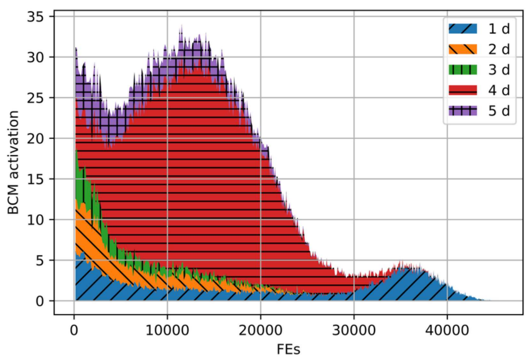
(c) AGSK; *Clipping*; f_2; $dim = 5$



(d) AGSK; *Clipping*; f_1; $dim = 10$



(e) AGSK; *Clipping*; f_2; $dim = 10$



(f) IMODE; *Clipping*; f_9; $dim = 5$

Fig. 4.7 BCM activation over the run of the algorithm. [33]

tions and identifying the most effective BCM for a given problem or algorithm, researchers and practitioners can tailor BCM implementation to enhance performance in optimization tasks.

5 THE CONTRIBUTION TO SCIENCE AND PRACTICE

Metaheuristic-based optimization methods are currently enjoying immense popularity. Alongside this growing popularity, the volume of research articles on this subject is also expanding, with continuous development and modification of new and existing algorithms respectively. A crucial aspect of this development process is the meticulous control of parameters that govern the behavior of the algorithm. Given that the primary focus is often on algorithm performance, selecting optimal control parameters is critical. The term "control parameter" encompasses a range of variables, including information storage systems and the selection of methods for managing critical states.

The BCMs are categorized as procedures for handling these critical states, as they are applied when a trial solution falls into an infeasible space. Preliminary studies indicate that each BCM may affect algorithm performance differently, necessitating careful consideration during the selection process. An ill-advised selection of BCMs can degrade the performance of the metaheuristic algorithm or alter other behaviors, potentially compromising specific desired characteristics of the algorithm.

Research articles that propose new metaheuristic algorithms or modifications to existing ones often overlook BCMs. Without specific details on the BCMs used by the original authors, subsequent implementations of such algorithms may be imprecise, leading to variations in effectiveness when solving specific tasks.

Addressing the research gap described above, which focuses on the often overlooked BCMs, constitutes the main part of this thesis. The first goal of this work is to raise awareness within the scientific community about the importance of BCMs, demonstrated by the presented results which prove their real impact on algorithm performance. This impact is evident not only in basic metaheuristic algorithms but also in state-of-the-art variants that have participated in benchmark competitions.

The second goal targets algorithm designers, who are urged to pay careful attention to providing a detailed description of the algorithm and its setup. This is crucial for the reproducibility of results and the effective evaluation of algorithm performance across different implementations.

The work highlights the significance of BCMs in the development and benchmarking of metaheuristic algorithms, and BCMs should also be important components in the automatic design or configuration of algorithms. It is imperative that these components are not merely mentioned as afterthoughts but are integrated into the core design and reporting of algorithmic research to ensure accuracy and replicability in scientific studies.

6 GOAL FULFILLMENT

This section outlines the steps implemented to achieve the dissertation goal, which were established as follows:

- ✓ **Survey the current state of boundary control methods (BCMs) used in evolutionary algorithms:** The current state of BCMs was extensively reviewed and a comprehensive survey of the literature was conducted not only to establish a foundational understanding of BCM applications in evolutionary algorithms but also to investigate which BCMs are being utilized. Additionally, this review explored whether other researchers, particularly in the context of algorithm design and benchmarking, are addressing these issues.
- ✓ **Investigate the influence of various BCMs on the performance of selected evolutionary algorithms:** Initial studies investigated the impact of BCMs on both basic and more advanced variants of various

algorithms such as PSO, FA, and SOMA. These investigations laid the groundwork for subsequent experiments aimed at assessing the influence of BCMs on state-of-the-art algorithms.

- ✓ **Conduct experiments evaluating the impact of BCMs on the performance of state-of-the-art algorithms and results of competitive benchmarking:** The experiments were carried out as described in Section 4, which tested the efficacy and the influence of BCMs on state-of-the-art algorithms using modern competitive benchmarks. The results from these experiments corroborated the preliminary findings from the initial studies, confirming the significant impact of BCMs on algorithm performance.
- ✓ **Based on the experimental results, draw conclusions and recommendations for good practices in benchmarking:** The conclusions and recommendations were presented, underscoring the significant influence of BCMs on the performance of metaheuristic algorithms. As evidenced by the experimental results in Section 4, appropriately chosen BCMs significantly altered the competitive ranking of the algorithms. The analysis of competition algorithms and the survey of state-of-the-art literature suggest that BCMs should be considered an integral part of the hyperparameter analysis of any algorithm design. Authors should always specify which BCM implementation was chosen to ensure fair comparison and reproducibility.

These findings underscore the importance of BCMs in the design and analysis of metaheuristic algorithms, advocating for their consistent inclusion in algorithmic research and documentation.

7 CONCLUSION

This dissertation provides a comprehensive treatise on the use of Boundary Control Methods (BCMs) in metaheuristic algorithms. The initial sections introduce the fundamental concepts and relationships among these fields, with a specific focus on the role and impact of BCMs in contemporary EA trends.

Following the introduction, the dissertation delineates the proposed goals and the methodologies employed to achieve them. A detailed examination of BCMs and a state-of-the-art overview are presented, where five representative BCMs are selected based on the literature review. The methodology and preliminary results, previously published at international conferences by the Author, are described in subsequent sections. After these preliminary results, further experiments were conducted to evaluate the impact of BCMs on the performance of state-of-the-art algorithms and the results of competitive benchmarking. Additionally, an experiment was included to investigate the frequency of BCM usage among state-of-the-art algorithms in the CEC20 benchmark.

While BCMs are often an overlooked part of experiment design in metaheuristics benchmarking, this dissertation highlights the importance of understanding BCMs as a necessary input for results reproducibility and potential performance improvement. The experimental findings from the CEC17 benchmark participants clearly demonstrate this. Notably, the LSHADE-cnEpSin algorithm could have won the CEC17 competition if it had employed the *random* BCM. Additionally, it was observed that none of the three tested algorithms achieved the best results with their original BCMs, indicating that performance improvements were possible through alternative BCMs. However, only five of the 12 CEC17 participants provided details on their BCM practices, with two of these reports being incomplete, which significantly impairs the reproducibility of their experiments. These findings underscore the significant influence of BCMs on the performance of metaheuristic algorithms, as evidenced in Section 5.4, where appropriately chosen BCMs significantly altered the competitive ranking of the algorithms. The analysis of competition algorithms and the survey of state-of-the-art literature suggest that BCMs should be considered an integral part of the hyperparameter analysis of any algorithm design. Authors should always specify which BCM implementation was chosen to ensure fair comparison and reproducibility, advocating for their consistent inclusion in algorithmic research and documentation.

This dissertation has conclusively demonstrated that BCMs are not merely supplementary components but are integral to the effective design, analysis, and application of metaheuristic algorithms. Their role should be carefully considered and integrated into future research and practice in the field of optimization. The importance of future research on BCMs lies in their universal applicability and profound impact across the entire field of metaheuristic optimizers, particularly in bound-constrained scenarios. Such research is crucial for advancing our understanding and implementation of these methods, ensuring they contribute significantly to the robustness and efficacy of optimization solutions.

8 CURRICULUM VITAE

Personal Information:	
Name	Tomas Kadavy
Date of birth	08 September 1990
Nationality	Czech
Contact	phone: +420 731 861 391 email: kadavy@utb.cz
Education:	
2010 - 2014	Tomas Bata University in Zlín, Faculty of Applied Informatics, Information and Control Technologies (Bc.)
2014 - 2016	Tomas Bata University in Zlín, Faculty of Applied Informatics, Information Technologies (Ing.)
2016 - present	Tomas Bata University in Zlín, Faculty of Applied Informatics, Engineering Informatics (Ph.D.)
Language knowledge:	
Czech	Native
English	Intermediate (Doctoral exam)
Professional achievements:	
Others	Nominee for Joseph Fourier Prize 2019 Erasmus+ - Univeristy of Twente, Netherlands, 2018 (2 months)
Grant activities	Grant Agency of the Czech Republic, GACR 15-06700S (member of the team) European Cooperation in Science and Technology (COST), IC1406 (member of the team) MSM - Mobility, 8J22AT006, Use of Evolutionary Algorithms for the Design and Optimization of 3D Antennas MPO - OPPIK, CZ.01.1.02/0.0/0.0/20_321/0023870 (Junior researcher) European Cooperation in Science and Technology (COST), CA22137 (member of the team)

REFERENCES

[1] LIN, M.-H., TSAI, J.-F. and YU, C.-S. A review of deterministic optimization methods in engineering and management. *Mathematical Problems in Engineering.* 2012, vol. 2012.

[2] MANGASARIAN, O. A Newton method for linear programming. *Journal of Optimization Theory and Applications.* 2004, vol. 121, no. 1, pp. 1–18.

[3] HASDORFF, L. Gradient optimization and nonlinear control, 1976.

[4] BACK, T. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms.* Oxford university press, 1996.

[5] GENDREAU, M., POTVIN, J.-Y. and OTHERS. *Handbook of metaheuristics.* vol. 2. Springer, 2010.

[6] WONG, W. and MING, C. I. A review on metaheuristic algorithms: recent trends, benchmarking and applications. In *2019 7th International Conference on Smart Computing & Communications (ICSCC)*, pp. 1–5. IEEE, 2019.

[7] SÖRENSEN, K. Metaheuristics—the metaphor exposed. *International Transactions in Operational Research.* 2015, vol. 22, no. 1, pp. 3–18.

[8] KADAVY, T., VIKTORIN, A., KAZIKOVA, A., PLUHACEK, M. and SENKERIK, R. Impact of boundary control methods on bound-constrained optimization benchmarking. *IEEE Transactions on Evolutionary Computation.* 2022, vol. 26, no. 6, pp. 1271–1280.

[9] KADAVY, T., PLUHACEK, M., VIKTORIN, A. and SENKERIK, R. Comparing strategies for search space boundaries violation in PSO. In *International Conference on Artificial Intelligence and Soft Computing*, pp. 655–664. Springer, 2017.

[10] RIGET, J. and VESTERSTRØM, J. S. A diversity-guided particle swarm optimizer-the ARPSO. *Dept. Comput. Sci., Univ. of Aarhus, Aarhus, Denmark, Tech. Rep.* 2002, vol. 2, pp. 2002.

[11] CHEN, Q., LIU, B., ZHANG, Q., LIANG, J., SUGANTHAN, P. and QU, B. Problem definitions and evaluation criteria for CEC 2015 special session on bound constrained single-objective computationally expensive numerical optimization. Technical report, Nanyang Technological University, 2014.

[12] FRIEDMAN, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association.* 1937, vol. 32, no. 200, pp. 675–701.

[13] DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research.* 2006, vol. 7, no. Jan, pp. 1–30.

[14] KADAVY, T., PLUHACEK, M., VIKTORIN, A. and SENKERIK, R. Boundary Strategies For Firefly Algorithm Analysed Using CEC'17 Benchmark. In *ECMS*, pp. 170–175, 2018.

[15] KORA, P. and KRISHNA, K. S. R. Hybrid firefly and particle swarm optimization algorithm for the detection of bundle branch block. *International Journal of the Cardiovascular Academy.* 2016, vol. 2, no. 1, pp. 44–48.

[16] Awad, N., Ali, M., Liang, J., Qu, B. and Suganthan, P. Problem definitions and evaluation criteria for the cec 2017 special sessionand competition on single objective real-parameter numerical optimization. Nanyang Technologial University, Jordan University of Science and Technology and Zhengzhou University, Singapore and Zhenzhou. *Nanyang Technological University, Jordan University of Science and Technology and Zhengzhou University, Singapore and Zhenzhou, China, Tech. Rep.* 2016, vol. 201611.

[17] Kadavy, T., Pluhacek, M., Senkerik, R. and Viktorin, A. Boundary Strategies for Self-organizing Migrating Algorithm Analyzed Using CEC'17 Benchmark. In *Swarm, Evolutionary, and Memetic Computing and Fuzzy and Neural Computing*, pp. 58–69. Springer, 2019.

[18] Yue, C. T., Price, K. V., Suganthan, P. N., Liang, J. J., Ali, M. Z., Qu, B. Y., Awad, N. H. and Biswas, P. P. Problem Definitions and Evaluation Criteria for the CEC 2020 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization. Technical report, Nanyang Technological University, 2019.

[19] Awad, N. H., Ali, M. Z., Liang, J. J., Qu, B. Y. and Suganthan, P. N. CEC 2017 Special Session on Single Objective Numerical Optimization Single Bound Constrained Real-Parameter Numerical Optimization, Jul 2019.

[20] Kumar, A., Misra, R. K. and Singh, D. Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase. In *2017 IEEE congress on evolutionary computation (CEC)*, pp. 1835–1842. IEEE, 2017.

[21] Brest, J., Maučec, M. S. and Bošković, B. Single objective real-parameter optimization: Algorithm jSO. In *2017 IEEE congress on evolutionary computation (CEC)*, pp. 1311–1318. IEEE, 2017.

[22] Tanabe, R. and Fukunaga, A. S. Improving the search performance of SHADE using linear population size reduction. In *2014 IEEE congress on evolutionary computation (CEC)*, pp. 1658–1665. IEEE, 2014.

[23] Awad, N. H., Ali, M. Z., Suganthan, P. N. and Reynolds, R. G. An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems. In *2016 IEEE congress on evolutionary computation (CEC)*, pp. 2958–2965. IEEE, 2016.

[24] Awad, N. H., Ali, M. Z. and Suganthan, P. N. Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 372–379. IEEE, 2017.

[25] Yue, C. T., Price, K. V., Suganthan, P. N., Liang, J. J., Ali, M. Z., Qu, B. Y., Awad, N. H. and Biswas, P. P. Competition on Single Objective Bound Constrained Numerical Optimization, Sep 2020.

[26] Sallam, K. M., Elsayed, S. M., Chakrabortty, R. K. and Ryan, M. J. Improved multi-operator differential evolution algorithm for solving unconstrained problems. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8. IEEE, 2020.

[27] Mohamed, A. W., Hadi, A. A., Mohamed, A. K. and Awad, N. H. Evaluating the performance of adaptive GainingSharing knowledge based algorithm on CEC 2020 benchmark problems. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8. IEEE, 2020.

[28] Mohamed, A. W., Hadi, A. A. and Mohamed, A. K. Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm. *International Journal of Machine Learning and Cybernetics*. 2020, vol. 11, no. 7, pp. 1501–1529.

[29] Brest, J., Maučec, M. S. and Bošković, B. Differential evolution algorithm for single objective bound-constrained optimization: Algorithm j2020. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8. IEEE, 2020.

[30] Brest, J., Greiner, S., Boskovic, B., Mernik, M. and Zumer, V. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE transactions on evolutionary computation*. 2006, vol. 10, no. 6, pp. 646–657.

[31] Brest, J., Maučec, M. S. and Bošković, B. The 100-digit challenge: Algorithm jde100. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 19–26. IEEE, 2019.

[32] Helwig, S., Branke, J. and Mostaghim, S. Experimental analysis of bound handling techniques in particle swarm optimization. *IEEE Transactions on Evolutionary computation*. 2012, vol. 17, no. 2, pp. 259–271.

[33] Kadavy, T., Pluhacek, M., Viktorin, A. and Senkerik, R. Exploring the Frequency of Boundary Control Methods Activation in Metaheuristic Algorithms. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, pp. 2330–2336, 2023.

PUBLICATIONS OF THE AUTHOR

Journals with IF

[1] KADAVY, Tomas, VIKTORIN, Adam, KAZIKOVA, Anezka, PLUHACEK, Michal, SENKERIK, Roman. Impact of Boundary Control Methods on Bound-constrained Optimization Benchmarking. IEEE Transactions on Evolutionary Computation, 2022, vol. 26, no. 6, pp. 1271-1280. ISSN 1089-778X.

[2] KADAVY, Tomas, SENKERIK, Roman, PLUHACEK, Michal, VIKTORIN, Adam. Orthogonal Learning Firefly Algorithm. LOGIC JOURNAL OF THE IGPL, 2021, vol. 29, no. 2, pp. 167-179. ISSN 1367-0751.

[3] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas, ZAMUDA, Ales. Distance based parameter adaptation for Success-History based Differential Evolution. Swarm and Evolutionary Computation, 2019, vol. 2019, no. 50, pp. 1-17. ISSN 2210-6502.

[4] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas. Analysing knowledge transfer in SHADE via complex network. LOGIC JOURNAL OF THE IGPL, 2020, vol. 28, no. 2, pp. 153-170. ISSN 1367-0751.

[5] PLUHACEK, Michal, VIKTORIN, Adam, SENKERIK, Roman, KADAVY, Tomas, ZELINKA, Ivan. Extended experimental study on PSO with partial population restart based on complex network analysis. LOGIC JOURNAL OF THE IGPL, 2020, vol. 28, no. 2, pp. 211-225. ISSN 1367-0751.

[6] PLUHACEK, Michal, KAZIKOVA, Anezka, VIKTORIN, Adam, KADAVY, Tomas, SENKERIK, Roman. Chaos in popular metaheuristic optimizers - a bibliographic analysis. Journal of Difference Equations and Applications, 2023, vol. 29, no. 9-12, pp. 1228-1243. ISSN 1023-6198.

[7] MAIR, Dominik, RENZLER, MICHAEL, KOVAR, Stanislav, MARTINEK, Tomas, KADAVY, Tomas, BERGMUELLER, SIMON, HORN, ANDRADA, BRAUN, JAKOB, KASERER, LUKAS. Evolutionary Optimized 3D WiFi Antennas Manufactured via Laser Powder Bed Fusion. IEEE Access, 2023, no. 11, pp. 121914 - 121923. ISSN 2169-3536.

Journals in Scopus

[8] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas. Modified progressive random walk with chaotic PRNG. International Journal of Parallel, Emergent and Distributed Systems, 2017, pp. 1-10. ISSN 1744-5760.

[9] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas. Clustering analysis of the population in Db_SHADE algorithm. Mendel, 2018, vol. 24, no. 1, pp. 9-16. ISSN 1803-3814.

[10] SENKERIK, Roman, VIKTORIN, Adam, ZELINKA, Ivan, PLUHACEK, Michal, KADAVY, Tomas, KOMINKOVA OPLATKOVA, Zuzana, BHATEJA, Vikrant, CHANDRA SATAPATHY, Suresh. Differential Evolution And Deterministic Chaotic Series: A Detailed Study. Mendel, 2018, vol. 24, no. 2, pp. 61-68. ISSN 1803-3814.

[11] PLUHACEK, Michal, ZELINKA, Ivan, SENKERIK, Roman, VIKTORIN, Adam, KADAVY, Tomas. Particle swarm optimization with distance based repulsivity. Mendel, 2018, vol. 2018, 24, no. 2, pp. 81-86. ISSN 1803-3814.

[12] PLUHACEK, Michal, SENKERIK, Roman, VIKTORIN, Adam, KADAVY, Tomas. Chaos-enhanced multiple-choice strategy for particle swarm optimisation. International Journal of Parallel, Emergent and Distributed Systems, 2020, vol. 35, no. 6, pp. 603-616. ISSN 1744-5760.

[13] PLUHACEK, Michal, KAZIKOVA, Anezka, KADAVY, Tomas, VIKTORIN, Adam, SENKERIK, Roman. Relation of neighborhood size

and diversity loss rate in particle swarm optimization with ring topology. Mendel, 2021, vol. 27, no. 2, pp. 74-79. ISSN 1803-3814.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AGSK	Adaptive Gaining-sharing Knowledge
AI	Artificial Intelligence
ARPSO	Attractive and Repulsive Particle Swarm Optimization
BCM	Boundary Control Method
BBOB	Black-box Optimization Benchmarking
CD	Nemenyi Critical Distance
CEC	Congress on Evolutionary Computation
CMA	Covariance Matrix Adaptation
CMA–ES	Covariance Matrix Adaptation Evolution Strategy
COCO	COmparing Continuous Optimizers
CR	Crossover Rate
CS	Cuckoo Search
DE	Differential Evolution
DES	Differential Evolution Strategy
DISH	Distance based parameter adaptation for success-history based differential evolution
DYYPO	Dynamic Yin-Yang Pair Optimization
EA	Evolutionary Algorithms
EBOwithCMAR	Effective Butterfly Optimizer using Covariance Matrix Adapted Retreat Phase
ECT	Evolutionary Computing Technique
FA	Firefly Algorithm
FFPSO	Firefly and Particle Swarm Optimization
GA	Genetic Algorithm
IDEbestNsize	Enhanced individual-dependent differential evolution with population size adaptation
IEEE	Institute of Electrical and Electronics Engineers
iL–SHADE	Improved L–SHADE
IMODE	Improving Multi-objective Differential Evolutionary
jDE100e	Eigenvector Crossover jDE100
LSHADE	SHADE with Linear decrease of the population size
LSHADE–cnEpSin	Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood
LSHADE_SPACMA	LSHADE with semi-parameter adaptation hybrid with CMA-ES
MM_OED	Multi-method based orthogonal experimental design algorithm
MOS	Multiple Offspring Sampling
MP–EEH	Multi-population exploration-only exploitation-only hybrid
mpmL–SHADE	Multi-population Modified L-SHADE
NP	Population size
PPSO	Proactive particles in swarm optimization
PSO	Particle Swarm Optimization
RASP–SHADE	Ranked Archive Differential Evolution with Selective Pressure
RB–IPOP–CMA–ES	A restart CMA evolution strategy with increasing population size with midpoint
SHADE	Success–History based Adaptive DE
SOMA	Self-Organizing Migrating Algorithm
SOMA–ATO	SOMA All To One
SOMA–ATA	SOMA All To All
SOMA–CL	SOMA with Clustering-Aided Migration
TLBO–FL	Teaching learning based optimization with focused learning

Tomas Kadavy

Porušování Limitů Argumentů v Evolučních Algoritmech

Boundary Constraint Violation in Evolutionary Algorithms

Doctoral Thesis Summary

Published by: Tomas Bata Univesity in Zlín

nám. T. G. Masaryka 5555, 760 01 Zlín, the Czech Republic

Edition: published electronically

Typesetting by: Ing. Tomas Kadavy, Ph.D.

This publication has not undergone any proofreading or editorial review.

First Edition

Publication year: 2024

ISBN 978-80-7678-279-2