Tomas Bata University in Zlín
Faculty of Applied Informatics

Doctoral Thesis

Boundary Constraint Violation in Evolutionary Algorithms

Porušování Limitů Argumentů v Evolučních Algoritmech

Author:	**Ing. Tomas Kadavy**
Branch of study:	Engineering Informatics
Supervisor:	prof. Ing. Roman Šenkeřík, Ph.D.
Advisor:	doc. Ing. Michal Pluháček, Ph.D.

Zlín, 2024

At this place, I would like to thank my colleagues and friends from the office, Ing. Alzbeta Tureckova, Ph.D. and Ing. Adam Viktorin, Ph.D.. Together, we have managed not to lose our minds amid the trials of life as young academic workers. Their camaraderie and support have been invaluable throughout this journey. I must express my profound gratitude to my supervisor and consultant, prof. Ing. Roman Senkerik, Ph.D. and doc. Ing. Michal Pluhacek, Ph.D.. Without their expert guidance, this work would not have been half as excellent. Their insights and dedication have been pivotal to my research. I am also grateful to all other colleagues and friends from the faculty and beyond, who have kept the spirit of great fun and discussion alive during our coffee breaks. These moments have provided much-needed relief and inspiration. In addition, I must extend my thanks to OpenAI, whose most famous product, ChatGPT, has been instrumental in sculpting and improving most of the text in this thesis. Its assistance has been invaluable in refining my arguments and clarifying my thoughts. Finally, I would like to thank my family, who has endured my presence and supported me through this intense period of my academic career. Their unwavering support and encouragement have been my foundation. Thank you all for your indispensable contributions to my journey and this dissertation.

"Live slow, live well." – Vincent Van Sloth

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.

SUMMARY

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

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

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, 5, 6] 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 [7, 8, 9], stand out for their effectiveness in solving complex optimization problems, thereby highlighting the vast potential of Evolutionary Computing Techniques (ECTs).

The evolution of metaheuristic algorithms involves expanding theoretical foundations, enhancing strategies through hybridization [10], hyperparameter fine-tuning [11], and the integration of learning and adaptive mechanisms. Recent developments also focus on the automatic configuration and selection of algorithms [12], merging machine learning with evolutionary computation for improved performance prediction [13] and explainability [14], highlighting the growing importance of advanced ML techniques in algorithm optimization.

The significance of these advancements is often validated through rigorous benchmarking [15] on well-recognized platforms like the IEEE CEC benchmarks [16] and the COCO platform [17] BBOB testbed, which play a pivotal role in shaping the direction of algorithm design, comparison, and refinement.

Amidst this progress, the metaheuristic research community has increasingly focused on establishing best practices for benchmarking [18] to ensure fair, comprehensive comparisons and to gather deeper insights into algorithmic performance across diverse problem sets [19]. The push towards more transparent and standardized benchmarking protocols reflects a broader recognition of the need to enhance the reproducibility and clarity of results, particularly as metaheuristic algorithms are tasked with solving ever-more complex optimization problems. This initiative, supported by both independent researchers and professional organizations like IEEE, underscores the critical importance of a thorough understanding of optimization processes and methodologies, not only to advance the current state of knowledge but also to facilitate the transition of these insights into other advanced Artificial Intelligence (AI) methods.

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.

The subsequent sections will provide concise overviews of various basic metaheuristic algorithms selected for examination and comparison regarding their response to BCMs. These algorithms have been specifically chosen to cover a broad spectrum of strategies, some of which draw inspiration from natural phenomena, swarm behaviors, or self-organization processes [20, 21, 22]. This focus on a diverse set of metaheuristics is intentional, addressing a gap in existing research, which typically concentrates on a single algorithm or a closely related family of algorithms. Hence, this effort is directed towards offering a more holistic analysis of boundary constraint methods across the field of evolutionary algorithms, aiming to contribute to a comprehensive understanding of how different BCMs impact a wide range of metaheuristic approaches.

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 EVOLUTIONARY ALGORITHMS

The motivation behind Evolutionary Algorithms (EAs) [4, 5, 6] began with a relatively simple idea: nature, over millions of years, has evolved systems and organisms that are remarkably effective at solving complex problems. Examples include ants that intuitively find the shortest paths [23], bees optimizing their foraging strategies [24], the optimal growth patterns of certain plant species [25], and various species adapting to environmental changes through natural selection. The foundation of EAs is built upon Charles Darwin's theory of natural evolution [26], Gregor Mendel's research on species heredity [27], and studies on collective behavior and patterns [28].

Thus, a straightforward approach is to apply the vast knowledge of natural systems to real-life optimization challenges. In the context of EAs, this means transforming these natural experiences into metaheuristic algorithms. The general process often involves taking inspiration from a natural process, which is then translated into the pseudo-code of a new algorithm. However, this translation often involves simplification and metaphorization, leading to algorithms that may not always be as robust as their natural counterparts. Moreover, despite the wide array of algorithms within the EA field, only a few have achieved widespread popularity and use. The rapid proliferation of metaheuristic algorithms has drawn criticism [29, 30], particularly regarding the development of new algorithms based on unjustified concepts drawn from nature, created primarily for rapid publication.

Growing consensus indicates that the creation of these algorithms, despite their innovative surface, often lacks substantial justification in terms of the natural concepts they are based on. Many in the field express concern that such algorithms are more about achieving publication than meaningfully contributing to the field of optimization. This practice not only dilutes the impact of genuinely innovative algorithms but also raises questions about the effectiveness and robustness of these rapidly produced methods. Therefore, there is a pressing need for more rigorous standards in the development and evaluation of new EAs.

These standards should prioritize not just the novelty of the natural concept but also the practical applicability and improvement in optimization effectiveness, ensuring that contributions to the field are both scientifically sound and technically valuable, aligning more closely with the principles of meaningful scientific advancement rather than the pursuit of publication metrics.

3.1 Principles and Mechanisms of Evolutionary Algorithms

In the realm of computational problem-solving, optimization plays a pivotal role, particularly within the context of Evolutionary Algorithms (EAs) [31]. Optimization fundamentally involves selecting the most effective combination of parameters. This combination is then evaluated based on its overall performance, typically measured by an objective function. The mathematical goal is to find a solution $\boldsymbol{x} = \{x_1, x_2, \ldots, x_D\}$ that optimizes (minimizes or maximizes) the objective function $f(\boldsymbol{x})$. This optimization process can be subject to various constraints, either defined by functions $g_k(\boldsymbol{x}) \leq 0;\ k = 1, \ldots, n$ or by parameter bounds $l_j \leq x_j \leq h_j;\ j = 1, \ldots, D$, where D represents the dimensionality of the optimization problem, indicating the number of parameters to be optimized.

EAs approach this optimization task by simulating the process of natural evolution. They operate on a population of potential solutions, applying mechanisms akin to genetic inheritance, mutation, recombination, and selection. Through iterative processes, these algorithms evolve the population, aiming to improve the quality of solutions with respect to the objective function. Each member of the population, often referred to as an individual or a chromosome, represents a potential solution to the optimization problem, encoded in a structure analogous to the genetic makeup of living organisms.

The effectiveness of EAs in navigating complex, multidimensional search spaces lies in their ability to balance exploration and exploitation [32]. Exploration involves searching through new, untested areas of the solution space, while exploitation focuses on refining the already discovered promising areas. This balance is critical in avoiding local optima traps and steering the algorithm towards the global optimum.

In summary, the essence of optimization in the context of EAs is rooted in the careful selection and iterative improvement of solutions, guided by the principles of natural evolution. This approach is particularly effective in dealing with complex problems where traditional optimization techniques may struggle. Traditional methods often face challenges with complex problems due to their tendency to become trapped in local optima, their sensitivity to the dimensionality of the search space, and their inability to handle discontinuous or non-differentiable functions effectively. Unlike these methods, EAs are more robust in exploring diverse solution spaces, as they use populations of solutions and stochastic processes, which help avoid premature convergence on sub-optimal solutions and provide a broader exploration of the problem space.

3.2 Genetic Algorithm

The Genetic Algorithm (GA), a renowned member of the EAs family, was pioneered in the 1960s by John Holland [33]. A hallmark of the classical GA is its unique approach to representing individual parameters encoded as a binary string. These parameters, termed genes, collectively form a chromosome, representing one individual's solution. An entire set of these chromosomes constitutes the population in GA. The main functional components of GA are segmented into three distinct phases.

Selection: This initial phase involves choosing parent chromosomes from the population for reproduction, based on their performance as measured by the objective function. Common methods employed for selection include the roulette rule, rank selection, and tournament selection. Each method influences the rate of convergence and the overall quality of the resulting solution. Typically, individuals with higher solution quality have a greater probability of being selected to pass their genetic information to the subsequent generation.

Crossover: This phase is central to the genetic algorithm, epitomizing the core concept of genetic recombination. The crossover process entails combining genetic information from two parent chromosomes to produce offspring. Various crossover methods exist, but the focus here is on the single-point crossover. As the name implies, single-point crossover involves dividing each parent's chromosome at a specific point and exchanging the segments, resulting in two offspring that inherit a blend of genetic material from both parents.

Mutation: The final phase is mutation, where, with a low probability, certain genes in the offspring's chromosomes may be inverted. The primary objective of mutation is to introduce genetic diversity into the population and to avert premature convergence on suboptimal solutions. The offspring, post-mutation, are then included in the new generation for the next cycle of the algorithm.

Upon completion of the mutation phase, the selection process recommences, marking the start of a new iteration. This cycle continues until a new generation of the population is formed. The description here encapsulates one of the most basic variants of GA. Over the years, numerous adaptations of GA have been developed, many tailored to address specific optimization problems with enhanced efficacy.

The pseudo-code of the described basic GA is shown in Algorithm 1.

Algorithm 1 GA

```
GA initialization
while Stopping criterion not met do
    for i = 1 to NP with i+ = 2 do
        Selection
        Crossover
        Mutation
        Evaluation
    end for
    record the best solution
end while
```

3.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) stands as a notable example of so-called swarm-based algorithms, first introduced by Eberhart and Kennedy in 1995 [34]. The PSO algorithm is inspired by the social behavior and movement patterns observed in natural swarms, such as bird flocking. According to the authors of the proposed modifications, modern adaptations of PSO primarily focus on addressing its known weakness: the tendency for premature convergence to local optima [35, 36].

In PSO, each member of the swarm, denoted as $\boldsymbol{x}$, navigates the solution space of the optimization problem. This movement is governed by two key factors: the individual's current position $\boldsymbol{x}$ and its velocity $\boldsymbol{v}$. The position of each swarm member is updated according to the equation:

$$x_{i,j}^{k+1} = x_{i,j}^{k} + v_{i,j}^{k} \tag{3.1}$$

where $x_{i,j}^{k+1}$ represents the position of the i-th individual in the j-th dimension during the $(k+1)$-th iteration of the algorithm.

Each individual in the swarm maintains a record of its personal best position, $\boldsymbol{pBest}$, while the collective swarm retains information about the overall best solution found, $\boldsymbol{gBest}$. These memories, $\boldsymbol{pBest}$ and $\boldsymbol{gBest}$, are integral to the calculation of the velocity $\boldsymbol{v}$, as described in the following equation:

$$v_{i,j}^{k+1} = w \cdot v_{i,j}^{k} + c_1 \cdot r_1 \cdot (pBest_{i,j} - x_{i,j}^{k}) + c_2 \cdot r_2 \cdot (gBest_j - x_{i,j}^{k}) \tag{3.2}$$

Here, w denotes the inertia weight, c_1 and c_2 are cognitive and social factors, and r_1 and r_2 are random numbers drawn from an unimodal distribution within the range $< 0, 1 >$. The pseudo-code for PSO is detailed in Algorithm 2.

Algorithm 2 PSO

```
PSO initialization
while Stopping criterion not met do
    for i = 1 to NP do
        calculate the velocity v (3.2)
        calculate the new position x (3.1)
        evaluate the x
        update the positions pBest and gBest
    end for
    record the best solution
end while
```

3.4 Differential Evolution

Differential Evolution (DE), a notable algorithm introduced by Storn and Price in [37], forms the foundation of a family of algorithms widely recognized for their effectiveness in continuous optimization. The majority of DE variants adhere to the foundational principles outlined in the original 1995 publication [38].

At its core, DE evolves from a randomly generated population of individuals, where each individual $\boldsymbol{x}$ signifies a potential solution to the optimization problem. These individuals are represented as vectors of parameters, with each component of the vector corresponding to a distinct attribute of the problem. The evolution of these individuals is facilitated through a combination of mutation, crossover, and selection processes, aiming to produce superior offspring in subsequent iterations. This evolutionary cycle continues until a predefined stopping criterion is met, ensuring that each new iteration comprises solutions that are either better or equivalent to those in the preceding iteration. The DE algorithm is fundamentally structured into three key steps.

Mutation: This step involves the formation of a mutated vector $\boldsymbol{v}$ through the combination of vectors from selected individuals. This process is governed by a scaling factor F, as illustrated in the equation:

$$v_{i,j}^k = x_{r_1,j}^k + F \cdot (x_{r_2,j}^k - x_{r_3,j}^k) \tag{3.3}$$

Here, $v_{i,j}^{k}$ represents the mutated vector in the j-th dimension for the i-th individual at iteration k. The indices r_1, r_2, r_3 are randomly selected from the population and are distinct from each other and also distinct from i-th individual, with NP denoting the total population size.

Crossover: The creation of the trial vector $\boldsymbol{u}$ occurs in this step. It involves choosing attributes either from the mutated vector $\boldsymbol{v}$ or the original vector $\boldsymbol{x}$, based on a crossover probability determined by the crossover rate CR. The process is encapsulated in the following equation:

$$u_{i,j}^{k+1} = \begin{cases} v_{i,j}^{k}, & \text{if } (rand_j \leq CR) \\ x_{i,j}^{k}, & \text{otherwise} \end{cases} \tag{3.4}$$

The trial vector $\boldsymbol{u}$ is constructed ensuring that it incorporates at least one parameter from the mutated vector $\boldsymbol{v}$.

Selection: In this final step, the algorithm selects the superior solution between the trial vector $\boldsymbol{u}$ and the original vector $\boldsymbol{x}$, based on their respective objective function values $f(\boldsymbol{x})$ and $f(\boldsymbol{u})$. The selected solution then proceeds to the next iteration.

Through these systematic steps, DE efficiently navigates the solution space, ensuring the continuous improvement of solutions towards optimal results.

The pseudo-code for the general DE is shown in Algorithm 3.

3.5 Self-organising Migrating Algorithm

The Self-Organizing Migrating Algorithm (SOMA), developed by Zelinka in 1999 [39, 40], represents a significant contribution to swarm-based algorithms, incorporating elements of self-organization and various innovative techniques. Influenced by other EAs, SOMA integrates techniques such as discrete perturbation, akin to the mutation process found in evolution strategies. This aspect of SOMA

Algorithm 3 DE

```
DE initialization
while Stopping criterion not met do
    for i = 1 to NP do
        mutation (3.3)
        crossover (3.4)
        selection
        evaluation
    end for
    record the best solution
end while
```

not only enhances its effectiveness but also facilitates scalability and adaptability across diverse search spaces.

Central to SOMA is the concept of cooperation among individuals, mirroring the collaborative dynamics observed in PSO. In this context, an individual $\boldsymbol{x}$ represents a potential solution to the optimization problem at hand. The essence of this cooperation, as defined by the algorithm's creator, is the migration process of an individual towards another member within the population, expressed mathematically as:

$$x_{i,j}^{k+1} = x_{i,j}^{k} + (x_{L,j}^{k} - x_{i,j}^{k}) \cdot t \cdot PRTVector_j \tag{3.5}$$

In this equation, $x_{i,j}^{k+1}$ denotes the new position of the i-th individual in the j-th dimension for the next iteration step $k+1$, with $x_{i,j}^{k}$ being its current position. The term $x_{L,j}^{k}$ refers to the position of a 'leader,' selected based on a specific SOMA strategy, and t parameterizes individual discrete steps towards this leader, influenced by user-defined parameters $PathLength$ and $Step$.

The mutation process in SOMA is emulated through the $PRTVector_j$, which is generated for each step t and dictates the dimensions in which an individual will migrate towards the leader. The probability of mutation in each dimension is controlled by a user-defined parameter prt, influencing the mutation strength during migration, as detailed in the following equation:

$$PRTVector_j = \begin{cases} 1, & \text{if } (rand_j < prt) \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

SOMA is characterized by its various strategies for leader selection, with the two primary ones being the 'All-To-One' and 'All-To-All' strategies. The 'All-To-One' strategy (the pseudo-code is shown in Algorithm 4) involves selecting one leader per migration cycle based on their objective function value, with other individuals migrating towards this leader. The 'All-To-All' strategy, conversely, sees each individual migrate towards all others in succession, returning to its original position after each migration. The effectiveness of SOMA is showcased in these diverse strategies, each contributing uniquely to the algorithm's ability to explore and exploit the solution space effectively.

Algorithm 4 SOMA - All-To-One

SOMA initialization
while Stopping criterion not met **do**
    select leader x_L from population
    for $i = 1$ **to** NP **do**
        for $t = Step$ **to** $pathLength$ **with** $t+ = Step$ **do**
            generate $\boldsymbol{PRTVector}$ (3.6)
            migrate x_i to x_L (3.5)
            evaluate the migrated x_i
        end for
        save best x_i to the new population
    end for
    record the best solution
end while

3.6 Current Trends in Evolutionary Algorithms

The widespread use of Evolutionary Algorithms, particularly swarm-based ones, presents a paradox: their popularity has been both beneficial and problematic, prompting criticisms for the swift proliferation of new algorithms [29]. Despite these criticisms, research indicates [41] that the development of new algorithms continues steadily, affirming that the trend of creating innovative algorithms is ongoing. Consequently, there is a growing focus among experts on exploring deeper aspects of metaheuristic algorithms, such as runtime analysis, benchmarking, communication dynamics, diversity management, population stagnation, and robustness and efficiency studies [42, 43, 44, 45, 46, 47, 48, 49, 50]. This shift is crucial because, despite their practical efficacy, metaheuristic algorithms require a stronger theoretical foundation to fully leverage their capabilities and facilitate the development of new technologies.

This dissertation is aligned with this emerging focus and aims to deliver a detailed examination of a frequently neglected aspect of metaheuristic algorithms—the boundary control methods (BCM). The reproducibility of research is fundamental; thus, omitting critical details renders such studies impractical. Given that BCM can significantly influence the performance and other behavioral aspects of algorithms, it is vital to address them with the same rigor as all other control variables within these algorithms.

4 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 [29], 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.

The literature presents various methods for addressing boundary constraint violations. Studies suggest that the choice of BCM might influence the performance and other characteristics of an optimization algorithm, as suggested by general studies mentioning potential impacts, often focused on single algorithms. Despite this, there appears to be a lack of comprehensive, generalized studies examining the impact of BCMs on the performance of metaheuristic optimizers in existing publications.

4.1 Literature Overview

The exploration of boundary control methods is addressed in various research publications, with a range of depth in problem coverage. The body of literature pertaining to EA, specifically within the realm of metaheuristic optimization, can be categorized as follows:

- papers primarily dedicated to the exploration of BCMs as a fundamental aspect of EA,
- articles that acknowledge the use of such methods (often within the framework of novel algorithm design) or that develop/implement these methods in a specific optimization algorithm,
- and studies that neglect to discuss BCMs, though inclusion might be beneficial.

It is important to note that searching for relevant literature on this topic is complicated by the similarity in terminology between parameter bounds and objective function constraints. Frequently, while objective function constraints are the focus, parameter bounds are also examined. Thus, thoroughly investigating all related research, which encompasses numerous detailed studies, is both time-intensive and complex.

The primary category of papers should be those entirely dedicated to boundary control handling as a general topic within EA, encompassing various families of metaheuristic algorithms. Regrettably, there are few, if any, comprehensive resources in the literature that address the full spectrum of metaheuristic algorithms.

Papers that discuss boundary control techniques exhibit a range of focuses. While some papers concentrate solely on comparing these methods within a single algorithm, accompanied by statistical analyses and recommendations on which methods generally enhance algorithm performance, the conclusions of these studies are often specific to the algorithm under consideration. The proposed BCMs typically exploit unique aspects of these algorithms, which may limit their applicability for general use.

For PSO-based algorithms, the experimental analysis by Helwig, Branke, and Mostaghim in 2012 [51] compares several BCMs within the PSO algorithm. They concluded that such methods significantly affect algorithm performance and can introduce a considerable search bias. This analysis also considered various aspects of PSO, such as the velocity of individual particles and variables for optimal positions. Oldewage, Engelbrecht, and Cleghorn conducted a similarly comprehensive study on BCMs for the PSO algorithm in 2018 [52], finding that the hyperbolic method performed best, though its application was limited exclusively to PSO. Clerc's slightly less detailed research in 2006 [53] categorized BCMs in PSO as "confinements," describing and testing several on a limited dataset. Zhang et al. conducted a parallel study in 2004, comparing a different set of methods [54]. Additionally, Michalewicz and Koziel published an extensive study on parameter bounds mixed with constrained numerical optimization for GAs [55, 56]. Detailed work on infeasible solutions and five BCM strategies for addressing them in terms of DE are discussed by Kononova et al. [57].

Another subset of articles that mention BCMs do so only as part of the design of new metaheuristic algorithms, with minimal to no analysis. For instance, in their seminal effort to define a standard for PSO, Bratton and Kennedy [58] in 2007 only briefly mention the boundary conditions, selecting one type without any detailed discussion or references. Similarly, the paper by Mostaghim et al. in 2006 [59] focuses on the multi-objective version of PSO. While primarily dedicated to objective function constraints, it also summarily addresses BCMs alongside previous work. However, extracting relevant results from this study is somewhat challenging due to its primary focus. In another context, four BCMs are discussed in Hansen's tutorial paper on the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) algorithm [60], yet the impact of these methods on CMA-ES performance was not explored in the tutorial.

Up to this point, the majority of referenced research articles have focused on the family of PSO algorithms, with one notable exception related to CMA-ES.

This category of papers also encompasses research in the field of Differential Evolution (DE) algorithms. Ronkkonen et al. [61] describe a method for the DE algorithm where a trial solution is reflected off the boundary by the amount of the violation. A similar technique is employed in the works of Brest et al. [62], Price and Storn [63], Guo et al. [64], and Zhang et al. [65]. Additionally, Caraffini et al. provide more detailed work on structural bias—primarily caused by boundary constraints—with extensive results and discussion in [66]. Further work by Vermetten et al. led to the creation of a toolbox called BIAS, designed specifically for benchmarking structural bias, as detailed in [67].

The results and implications derived from the collected works suggest that BCMs can directly influence the overall performance of a metaheuristic algorithm. Yet, many new algorithms, tutorials, and overview articles have overlooked or neglected this aspect. For instance, this oversight occurs in a tutorial for PSO [68], a general paper on EAs [69], articles on the Firefly Algorithm (FA) [70, 71], a publication on the Cuckoo Algorithm [72], and a paper introducing new mechanics for DE [73].

The following quotation from the article on experimental analysis of BCM (stated as bound handling techniques in the article) in particle swarm optimization by Helwig et al. [51] emphasizes this point:

"As was shown, the bound handling technique has a significant impact on the performance of PSO, particularly when the search space is of high dimensionality, as this greatly increases the likelihood of a particle exiting the feasible area."

This statement, along with other publications listed in subsection 4.1, strongly suggests that BCMs require careful consideration—whether in proposing a new metaheuristic algorithm or when modifying an existing one. The findings in these publications also indicate that different methods are preferred for different algorithms, underscoring the need for extensive research to examine the employed methods across a variety of optimizers. Such research should aim to deeply analyze their effects not only on performance but also on other characteristics of the algorithms.

4.2 Existing Boundary Control Methods

This subsection presents a consolidated overview of Boundary Control Methods (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.

In the ensuing subsections, mathematical formulations of various BCMs are detailed. To facilitate comprehension, a brief recapitulation of the variables employed in these formulations is provided here. An individual solution in the context of these methods is denoted as $\boldsymbol{x} = \{x_1, x_2, \ldots, x_D\}$, representing a vector of parameters where D indicates the dimension size. Each parameter x_j within this vector, corresponding to a specific dimension, is subject to defined bounds that demarcate the feasible solution space. These parameter bounds are expressed as $l_j \leq x_j \leq h_j$, where l_j and h_j represent the lower and upper bounds, respectively, for the j-th dimension.

4.2.1 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. This approach is succinctly captured by the following equation, ensuring that each solution stays within its maximum and minimum limits.

$$x_{i,j}^{k+1} = \begin{cases} h_j & \text{, if } \left(x_{i,j}^k > h_j\right) \\ l_j & \text{, if } \left(x_{i,j}^k < l_j\right) \\ x_{i,j}^k & \text{, otherwise} \end{cases} \tag{4.1}$$

The $x_{i,j}^{k+1}$ is the i-th individual in j-th dimension in $k+1$ iteration, and the pair h_j and l_j are the parameter bounds, maximum and minimum respectively.

4.2.2 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. This method is straightforward to implement, as reflected in its mathematical representation.

$$x_{i,j}^{k+1} = \begin{cases} U\left(l_j, h_j\right) & \text{, if } \left(x_{i,j}^k > h_j \text{ OR } x_{i,j}^k < l_j\right) \\ x_{i,j}^k & \text{, otherwise} \end{cases} \tag{4.2}$$

4.2.3 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.

$$x_{i,j}^{k+1} = \begin{cases} h_j - \left(x_{i,j}^k - h_j\right) & \text{, if } \left(x_{i,j}^k > h_j\right) \\ l_j + \left(l_j - x_{i,j}^k\right) & \text{, if } \left(x_{i,j}^k < l_j\right) \\ x_{i,j}^k & \text{, otherwise} \end{cases} \tag{4.3}$$

4.2.4 Periodic

The Periodic Method approaches boundary violations by considering an infinite solution space, effectively creating infinite copies of the optimized hyper-space. 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.

$$x_{i,j}^{k+1} = l_j + \left((x_{i,j}^k - h_j) \text{ MOD } (h_j - l_j)\right) \tag{4.4}$$

4.2.5 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.

$$x_{i,j}^{k+1} = \begin{cases} x_{i,j}^{k-1} + \left(h_j - x_{i,j}^{k-1}\right)/2 & \text{, if } \left(x_{i,j}^k > h_j\right) \\ l_j + \left(x_{i,j}^{k-1} - l_j\right)/2 & \text{, if } \left(x_{i,j}^k < l_j\right) \\ x_{i,j}^k & \text{, otherwise} \end{cases} \tag{4.5}$$

4.2.6 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.

5 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) [74]. These studies aimed to explore the effect of the Boundary Constraint Methods (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" [75]. 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. The motivation for this research was based on the findings described in previous initial studies [76, 77, 78], where the influence of BCMs on the performance of selected basic versions of metaheuristic algorithms was examined. Thus, a research question arose as to whether the choice of a BCM could significantly influence other competitive algorithms, especially the CEC competition winners. Furthermore, it was motivating to find out whether just changing the BCM can help achieve even better results for the top three performing algorithms from a given year of the competition, possibly changing their final order. The results presented in the article showed that the impact on the empirical performance of the top 3 algorithms from CEC17 is affected by the choice of BCMs.

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.

A timeline of the contributions mentioned, detailing the progression and key milestones of the research, is illustrated in Figure 5.1.

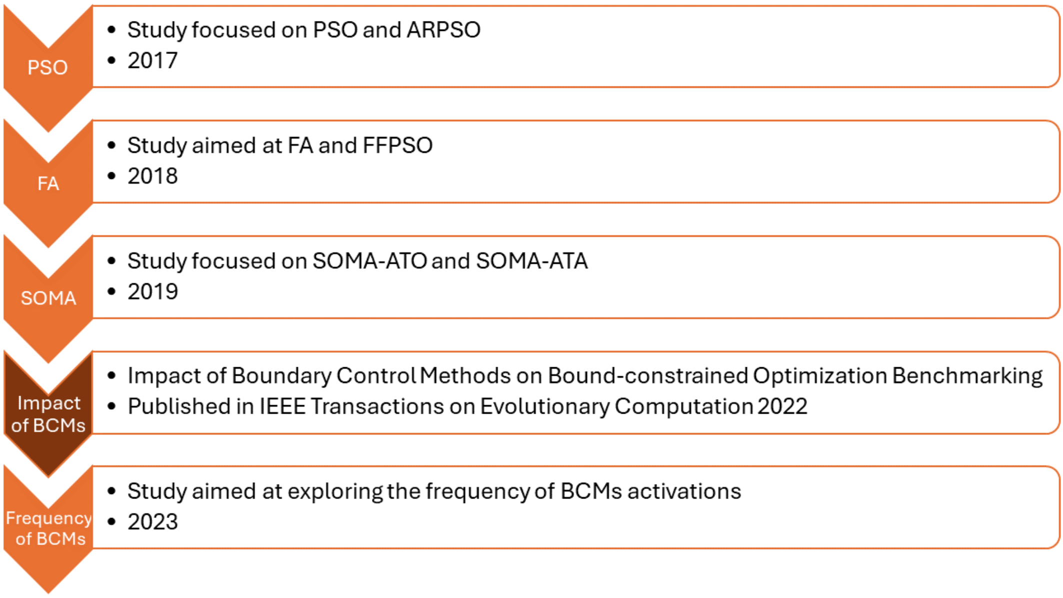


Fig. 5.1 Research timeline progress.

5.1 PSO

The study's results published in 2017 [76] 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) [35]. 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.

Operating on principles akin to those outlined in Section 3.3 for PSO, ARPSO involves computing the population's diversity in each iteration. Should this divergence fall below a predefined threshold, a repulsive phase is initiated to disperse the particles, thereby enhancing exploration. Conversely, if the divergence exceeds a certain upper threshold, an attractive phase is triggered, encouraging particles to converge, which aids in exploitation. This dynamic adjustment between attraction and repulsion phases helps in maintaining a balance between exploration and exploitation, which is crucial for avoiding premature convergence.

5.1.1 Experimental setup

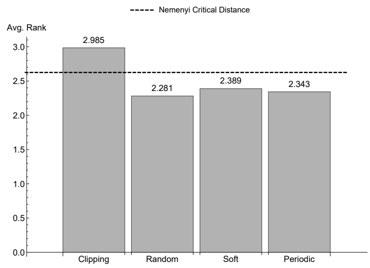
The experiments were conducted on the CEC 2015 benchmark set [79] for dimension sizes 10, 30, and 50. The benchmark encompasses 15 test functions, and every test function was repeated for 51 independent runs. The results were tested for statistical significance using the Friedman Rank test [80] with the significance level $\alpha = 0.05$, accompanied by Nemenyi critical distance (CD) [81].

5.1.2 Key findings

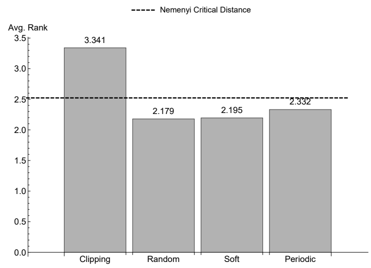
The Friedman rank tests (Fig. 5.2, Fig. 5.3) clearly show that the *Clipping* method negatively affected the overall performance of both PSO and ARPSO algorithms on all dimension sizes. The CD shows which BCMs do not alter the performance of the particular algorithm with statistical significance in comparison 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.

5.2 FA

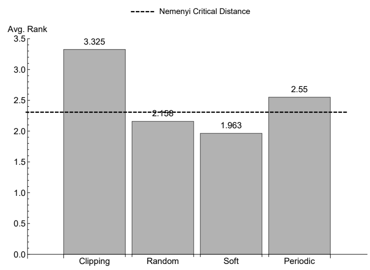
The subsequent study from 2018 [77] compared *Clipping*, *Random*, *Reflection*, and *Periodic* BCMs on FireFly Algorithm (FA) and on a hybrid of FA and PSO, called Firefly Particle Swarm Optimization (FFPSO) [82]. This hybrid algorithm was introduced in late 2016 by Padmavathi Kora and K. Sri Rama Krishna. The basic idea behind such an approach, according to the authors, is



(a) $D = 10$



(b) $D = 30$



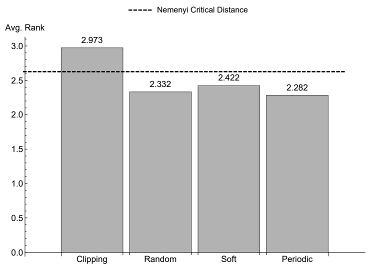
(c) $D = 50$

Fig. 5.2 Friedman rank comparison of the *Clipping*, *Random*, *Periodic*, and *Soft* methods on PSO on benchmark set CEC 2015. [76]

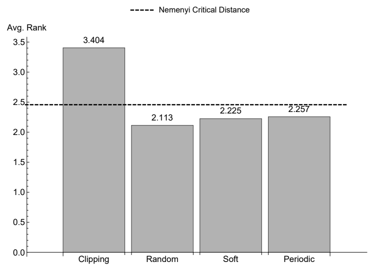
that the new hybrid strategy can share advantages from both algorithms and hopefully eliminate their disadvantages. The main principle remains the same as in the standard FA, but the equation for firefly motion is slightly changed according to PSO movement.

5.2.1 Experimental setup

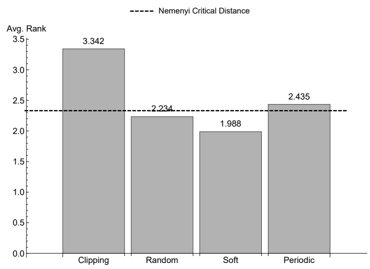
The experiments were performed on the CEC 2017 benchmark set [83], encompassing 30 test functions, and every test function was repeated for 51 independent runs. The tested dimension sizes were 10 and 30. The results were tested for statistical significance using the Friedman Rank test [80] with the significance level $\alpha = 0.05$, accompanied by Nemenyi critical distance (CD) [81].



(a) $D = 10$



(b) $D = 30$



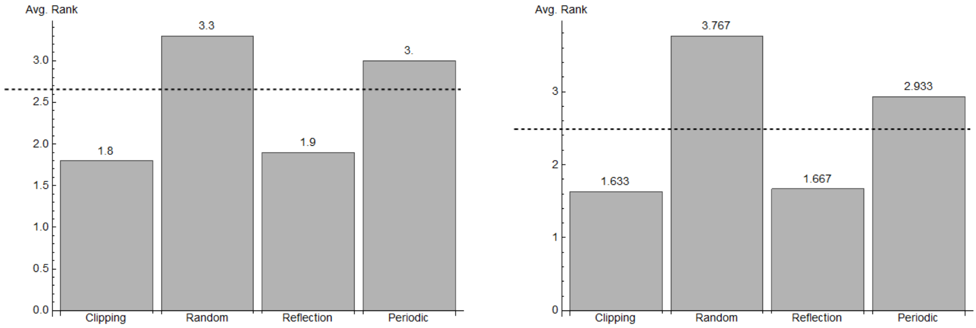
(c) $D = 50$

Fig. 5.3 Friedman rank comparison of the *Clipping*, *Random*, *Periodic*, and *Soft* methods on ARPSO on benchmark set CEC 2015. [76]

5.2.2 Key findings

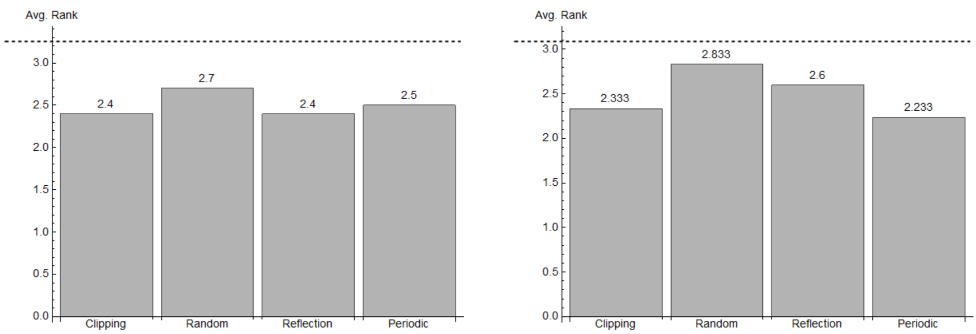
The figures Fig. 5.4 and Fig. 5.5 show the aforementioned ranks for FA and FFPSO algorithms, respectively. The lower the rank, the better the performance of the used BCM. Furthermore, the presented Friedman ranks are again accompanied by Nemenyi critical distance (CD), represented as the dashed line, measuring the critical distance from the best BCM (the lowest mean rank).

According to the ranking shown in both Figures (Fig. 5.4 and Fig. 5.5), the significant impact of the selected BCM is mostly observed on canonical FA. For the hybrid method FFPSO, the results indicate a minimal or negligible impact of the used method. For dimension sizes 10 and 30, the most favorable BCMs appear to be *clipping* and *reflection* methods.



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

Fig. 5.4 Friedman rank comparison of the *Clipping*, *Random*, *Reflection*, and *Periodic* methods on FA on benchmark set CEC 2017. [77]



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

Fig. 5.5 Friedman rank comparison of the *Clipping*, *Random*, *Reflection*, and *Periodic* methods on FFPSO on benchmark set CEC 2017. [77]

5.3 SOMA

The 2020 study [78] 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.

The study evaluated four distinct BCMs: *Clipping*, *Random*, *Reflection*, and *Periodic*. This comparative analysis sheds light on how different BCMs can influence the performance and robustness of SOMA variants, contributing valuable insights to the sparse body of knowledge on BCMs' role in enhancing SOMA's optimization capabilities.

5.3.1 Experimental setup

For the experiment, the CEC 2017 benchmark set [83] 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.

Regarding the SOMA parameters, they were configured as follows: the population size NP was 100 individuals; the probability of perturbation prt was set to 0.3; the $Step$ parameter was 0.11, and the $Path$ length was fixed at 3. These settings were in line with the recommendations of the original authors [39, 40, 84].

The analysis utilized the Friedman Rank test [80] to assess statistical significance, rejecting the null hypothesis of equal means at a 5% significance level.

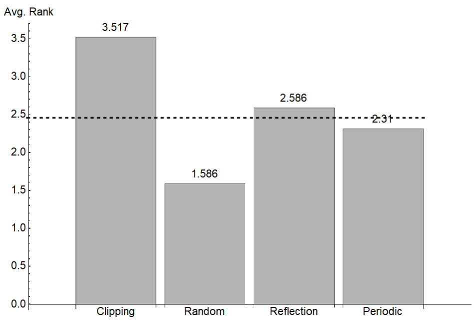
5.3.2 Key findings

Figure 5.6 displays the Friedman ranking outcomes for both the All-To-One and All-To-All SOMA strategies across the dimension sizes of 10 and 30, where lower ranks signify superior performance. To compare multiple BCMs, the Nemenyi Critical Distance post-hoc test was employed, determining a critical distance (CD). The critical distance is visually represented by a dashed line extending from the highest-ranked boundary method in the figure.

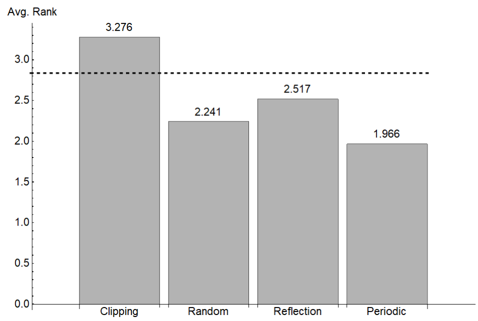
For the All-To-One strategy, the *Random* and *Periodic* methods emerged as notably effective, with the All-To-All strategy showing similar results and the *Reflection* method also presenting promise. Conversely, the *Clipping* method was identified as the least effective across strategies. The analysis distinguished two method groups: one (*Clipping*, *Reflection*, and *Periodic*) maintaining consistent individual migration across the search space without losing previous position information, and the other (*Random* method) facilitating a more stochastic search, potentially aiding SOMA in avoiding local minima traps.

The findings indicate the *Random* and *Periodic* methods as particularly promising, with the *Random* method introducing beneficial stochastic elements to SOMA. In contrast, the *Periodic* method preserves individuals' movement directions and patterns, offering a more predictable and natural behavior that aligns with original population dynamics.

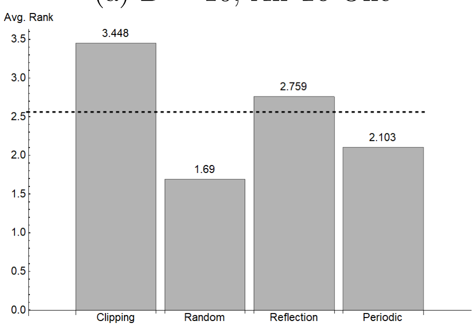
The least effective method for both SOMA strategies was found to be the *Clipping* method, potentially due to its restriction of individual movements to the borders of the feasible space, impacting the algorithm's performance negatively in both tested dimension sizes.



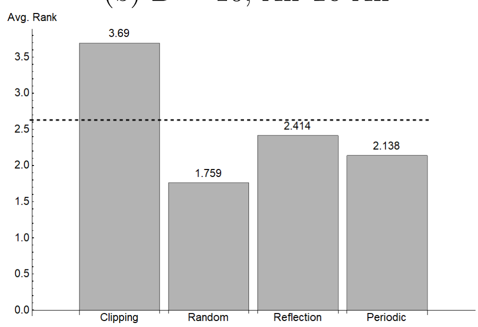
(a) $D = 10$; All-To-One



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



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



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

Fig. 5.6 Friedman rank comparison of the *Clipping*, *Random*, *Reflection*, and *Periodic* methods on SOMA on benchmark set CEC 2017. [78]

5.4 Analyzing the Impact of Boundary Control Methods on Algorithmic Performance

The impulse for the performance analysis study comes from earlier investigations [76, 77, 78] 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.

5.4.1 Experimental setup

The study was focused on the three top-ranking participants of two benchmark competitions: CEC17 [83], and CEC20 [85]. The goals for both testbeds were to:

- determine which BCM was used by the three winning algorithms,
- examine if there is a better choice of a BCM for a particular algorithm,
- if the algorithms used a different BCM, could it have changed the final order?

Both benchmarks encompass four groups of test functions: unimodal, multimodal, hybrid, and composition functions. An advantageous feature of the CEC benchmark is that all incorporated test functions are defined with equal and static (same values across all dimensions) search ranges for all parameters. The original implementation also supports a shift of the global optimum and rotation of each function.

The description of each benchmark is summarized in the following subsections alongside the descriptions of the three top-ranking algorithms of each benchmark.

Tab. 5.1 CEC17 – Algorithm overview

Algorithm	BCM	BCM listed
EBOwithCMAR [87]	Halving, Clipping	No
LSHADE-cnEpSin [88]	Halving	No
jSO [89]	Halving	Partially
DES [90]	Penalization	Yes
DYYPO [91]	–	No
IDEbestNsize [92]	Reflection	Partially
LSHADE_SPACMA [93]	–	No
MM_OED [94]	–	No
MOS-CEC2013 [95]	–	No
MOS-SOCO2011 [95]	–	No
PPSO [96]	Random bounce	Yes
RB-IPOP-CMA-ES [97]	–	No
TLBO-FL [98]	Clipping	Yes

5.4.2 Top 3 best-performing algorithms for CEC17

The testbed CEC17 published in 2016 [83] 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 [86]. Table 5.1 contains a list of all participants, including the used BCM, and if the BCM was mentioned in the accompanying paper.

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 [87] 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 [89] represents an improved variant of the iL-SHADE algorithm [99] and ranked in second place. The improvement lies predominantly in the new version

Tab. 5.2 CEC20 – Algorithm overview

Algorithm	BCM	BCM listed
IMODE [102]	Random, Halving	No
AGSK [103]	Halving	No
j2020 [104]	Periodic	Yes
Cssin [105]	–	No
MP-EEH [106]	Clamping	Yes
RASP-SHADE [107]	Halving	Yes
DISH-XX [108]	Halving	No
jDE100e [109]	–	No
OLSHADE [110]	Clipping, Halving	Yes
mpmL-SHADE [111]	Halving	Yes
SOMA-CL [112]	Random	No

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 [100], 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 [88] ranked third in the CEC17 competition and uses *Halving* BCM, which is unfortunately not mentioned in the paper by the authors.

5.4.3 Top 3 best-performing algorithms for CEC20

The CEC20 benchmark [85] 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 [101], and Table 5.2 contains a list of all participants, what boundary control method they used, and if the BCM was mentioned in the related paper.

IMODE is a Differential Evolution (DE) based algorithm ranked as the winner in the CEC20 competition. IMODE [102] 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 [103] is the enhanced version of the Gaining Sharing Knowledge-based algorithm (GSK) [113], which uses adaptive settings to its control parameters. The algorithm utilizes *Halving* BCM; however, the authors did not specify this.

j2020 [104] ranked third place in the competition, and it is based on the two self-adaptive DE algorithms: jDE [114] and jDE100 [115]. The used BCM is *Periodic*, which is described in the paper.

5.4.4 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 [80], and the results are presented in Table 5.3 for CEC17 and Table 5.4 for CEC20. The values in cells are the rankings for each algorithm for a particular dimension size. The columns indicate the tested BCM. If the algorithm used a different BCM than the selected (*Clipping*, *Random*, *Periodic*, *Reflection*, *Halving*), the column Default was used. Otherwise, this column states the name of the used BCM of the algorithm. The last column contains p-values of the Friedman rank test. The tested significance levels are 0.1, 0.05, 0.01, and 0.001. Each level corresponds to a certain symbol: *, †, **, and *** respectively. Therefore, a symbol represents the significance level of the result. The last row of each algorithm also contains the mean rank (given in bold) across the dimension sizes for a particular BCM. The last column, CD, stands for Nemenyi Critical Difference – if the difference between a pair of BCMs' ranks is higher than the CD value, they are significantly different. For example, in Table 5.3, the *Periodic* BCM in LSHADE-cnEpSin in 10D is significantly different (better according to rank) from *Clipping* BCM. But the same *Periodic* BCM is not significantly different from *Random* BCM.

Table 5.3 contains more results with statistically significant differences than Table 5.4. 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 [51] – 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.

5.4.5 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 [83, 85].

The final score value sorts the algorithms, and the higher score value, the better is the performance of the algorithm. This Score is a sum of two partial scores named Score 1 and Score 2. Score 2 is based on the weighted rank values, and Score 1 is computed from the normalized error values for CEC20 or mean (not normalized) error values for CEC17. Score 1 is computed using equations (5.1) and (5.2).

$$\begin{aligned} SE = 0.1 \cdot \sum_{i=1}^{29} ef_{10D} + 0.2 \cdot \sum_{i=1}^{29} ef_{30D} \\ +0.3 \cdot \sum_{i=1}^{29} ef_{50D} + 0.4 \cdot \sum_{i=1}^{29} ef_{100D} \end{aligned} \tag{5.1}$$

$$Score1 = \left(1 - \frac{SE - SE_{min}}{SE}\right) \cdot 50 \tag{5.2}$$

Where ef is the mean error value for a certain dimension size and SE_{min} is the minimal sum of errors among all algorithms. The Score 2 is then computed based on equations equations (5.3) and (5.4).

$$\begin{aligned} SR = 0.1 \cdot \sum_{i=1}^{29} rank_{10D} + 0.2 \cdot \sum_{i=1}^{29} rank_{30D} \\ +0.3 \cdot \sum_{i=1}^{29} rank_{50D} + 0.4 \cdot \sum_{i=1}^{29} rank_{100D} \end{aligned} \tag{5.3}$$

$$Score2 = \left(1 - \frac{SR - SR_{min}}{SR}\right) \cdot 50 \tag{5.4}$$

The final score is then defined as (5.5).

$$Score = Score1 + Score2 \tag{5.5}$$

The equations (5.1) – (5.5) describes the computation of the score for CEC17. The same equations are used to compute the CEC20 score but with different dimension sizes.

5.4.6 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.

The Tables 5.5 a) – f) contain the Score and rank of BCMs used for a particular algorithm. The default BCM is in bold. From the given results, not once did the default BCM ranked as the best performing variant; therefore, potentially better results for the algorithm may be achieved using the BCM with the highest Score. The parentheses under the score display percentual contribution of each dimensional setting to the score. The significant disproportion in values of Score 1 for dimension size 50 for CEC17 was caused by the last test function f_{30}, Since this disproportion is observed across all tested algorithms and their BCMs, the obtained results are still comparable, however further investigation is needed to find the cause of such behavior.

Friedman ranks suggest that higher dimension size has a more significant impact on the final score, as can be seen in Table 5.5. However, the higher impact of higher dimension sizes is also implicit due to the weighting of dimension parts of the score computation in (5.1) and (5.3).

Tab. 5.3 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. 5.4 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
	5D	3.25E+00	4.05E+00	3.45E+00	3.75E+00	3.40E+00	3.10E+00	4.60E–01	
	10D	3.40E+00	3.20E+00	3.60E+00	3.70E+00	3.10E+00	4.00E+00	8.63E–01	
IMODE	15D	3.95E+00	3.65E+00	2.85E+00	3.35E+00	3.55E+00	3.65E+00	7.44E–01	2.38E+00
	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**		
	5D		3.60E+00	3.10E+00	3.00E+00	3.15E+00	2.15E+00	9.27E–02*	
	10D		3.90E+00	2.30E+00	2.80E+00	2.70E+00	3.30E+00	1.45E–01	
AGSK	15D	Halving	3.45E+00	2.55E+00	3.25E+00	2.90E+00	2.85E+00	4.84E–01	1.93E+00
	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**		
	5D		3.70E+00	2.70E+00	2.90E+00	3.00E+00	2.70E+00	5.11E–01	
	10D		3.70E+00	1.70E+00	3.30E+00	2.90E+00	3.40E+00	1.98E–02†	
j2020	15D	Periodic	3.80E+00	2.20E+00	3.45E+00	2.65E+00	2.90E+00	9.56E–02*	1.93E+00
	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. 5.5 Score and rank of BCMs used for a particular algorithm. The rank is based on the final Score, which is a sum of partial scores 1 and 2. The default BCM of the algorithm is in bold. The parentheses under the score display percentual contribution of each dimensional setting to the score (CEC17 – {10D, 30D, 50D, 100D}, CEC20 – {5D, 10D, 15D, 20D}).

a) CEC17 – EBOwithCMAR

Rank	BCM	Score 1	Score 2	Score
1	Reflection	4.81E+01	5.00E+01	9.81E+01
		(0.4, 1.1, 92.6, 5.9)	(10.8, 23.4, 33.7, 32.)	
2	Random	4.98E+01	4.52E+01	9.49E+01
		(0.4, 1.1, 92.1, 6.3)	(9.7, 20.1, 31., 39.2)	
3	Halving	4.71E+01	4.75E+01	9.46E+01
		(0.4, 1.1, 92.7, 5.9)	(11.6, 19.5, 31., 37.9)	
4	Periodic	5.00E+01	4.30E+01	9.30E+01
		(0.4, 1.2, 92., 6.4)	(8.5, 18.8, 27.5, 45.2)	
5	**Default**	**4.78E+01**	**4.49E+01**	**9.27E+01**
		(0.4, 1.1, 92.4, 6.)	**(9.9, 19.5, 29.4, 41.3)**	
6	Clipping	4.62E+01	4.17E+01	8.79E+01
		(4.8, 1., 88.4, 5.8)	(9.7, 19.2, 28.1, 43.1)	

b) CEC17 – jSO

Rank	BCM	Score 1	Score 2	Score
1	Reflection	5.00E+01	5.00E+01	10.0E+01
		(0.4, 1.1, 90.5, 7.9)	(10.1, 22.8, 31.7, 35.4)	
2	Random	5.00E+01	4.69E+01	9.69E+01
		(0.4, 1.2, 89.9, 8.5)	(10.1, 20.5, 28.7, 40.7)	
3	**Halving**	**4.87E+01**	**4.38E+01**	**9.25E+01**
		(0.4, 1.1, 90.5, 8.)	**(10.2, 18.9, 29.2, 41.7)**	
4	Clipping	4.80E+01	4.04E+01	8.83E+01
		(2.7, 1.1, 88.7, 7.5)	(11.3, 20., 30.8, 38.)	
5	Periodic	4.98E+01	3.82E+01	8.80E+01
		(0.4, 1.2, 89.8, 8.6)	(8.4, 18.5, 29.7, 43.3)	

c) CEC17 – LSHADE-cnEpSin

Rank	BCM	Score 1	Score 2	Score
1	Random	5.00E+01	5.00E+01	1.00E+02
		(0.5, 1.2, 91.8, 6.5)	(9.2, 20., 30.4, 40.5)	
2	Periodic	4.92E+01	4.78E+01	9.71E+01
		(0.5, 1.2, 91.5, 6.8)	(8., 19.9, 28.3, 43.8)	
3	Reflection	4.60E+01	4.20E+01	8.79E+01
		(0.5, 1.1, 92.5, 5.9)	(10.6, 20.8, 30.2, 38.4)	
4	**Halving**	**4.44E+01**	**4.21E+01**	**8.65E+01**
		(3.7, 1.1, 89.7, 5.5)	**(10.9, 19.7, 29.9, 39.5)**	
5	Clipping	4.43E+01	3.66E+01	8.08E+01
		(0.4, 1.1, 92.7, 5.8)	(10.8, 19.7, 31., 38.6)	

d) CEC20 – IMODE

Rank	BCM	Score 1	Score 2	Score
1	Reflection	5.00E+01	5.00E+01	1.00E+02
		(1.4, 10.9, 34.5, 53.2)	(10.8, 19.7, 33.9, 35.6)	
2	Halving	4.95E+01	4.41E+01	9.36E+01
		(2.4, 11.1, 32.4, 54.1)	(8.7, 22.4, 30.7, 38.1)	
3	Random	4.47E+01	4.79E+01	9.27E+01
		(4.4, 10.3, 32.7, 52.6)	(10.5, 22., 26.1, 41.5)	
4	Periodic	4.35E+01	4.27E+01	8.63E+01
		(5.5, 7.9, 39.5, 47.)	(10.2, 20.1, 27.3, 42.4)	
5	**Default**	**4.37E+01**	**4.19E+01**	**8.57E+01**
		(4.3, 15.9, 29.9, 49.8)	**(8.7, 18.1, 31.6, 41.6)**	
6	Clipping	4.14E+01	4.39E+01	8.53E+01
		(4.1, 8.9, 27.4, 59.6)	(11.3, 17.9, 30.6, 40.2)	

e) CEC20 – AGSK

Rank	BCM	Score 1	Score 2	Score
1	Reflection	4.90E+01	4.58E+01	9.48E+01
		(0.4, 5., 29.2, 65.4)	(11.5, 19.7, 31.7, 37.2)	
2	**Halving**	**5.00E+01**	**4.38E+01**	**9.38E+01**
		(0.1, 5., 29.6, 65.3)	**(7.5, 23., 29.8, 39.7)**	
3	Random	4.36E+01	5.00E+01	9.36E+01
		(0.4, 4.7, 38., 56.9)	(12.3, 18.3, 30.4, 39.)	
4	Periodic	4.60E+01	3.96E+01	8.56E+01
		(0., 4.2, 31.9, 63.9)	(9.4, 17.6, 30.7, 42.2)	
5	Clipping	4.67E+01	3.40E+01	8.07E+01
		(0., 4.7, 30.1, 65.2)	(9.7, 21.1, 28., 41.1)	

Tab. 5.6 CEC17 – Score – Default BCM

Rank	Algorithm	Score 1	Score 2	Score
1	EBOwithCMAR	5.00E+01	5.00E+01	1.00E+02
2	jSO	4.97E+01	4.33E+01	9.30E+01
3	LSHADE-cnEpSin	4.68E+01	4.47E+01	9.15E+01
4	LSHADE_SPACMA	4.64E+01	4.47E+01	9.11E+01
5	DES	4.59E+01	4.12E+01	8.71E+01
6	MM_OED	4.60E+01	3.62E+01	8.22E+01
7	IDEbestNsize	2.98E+01	2.63E+01	5.61E+01
8	MOS-CEC2013	1.89E+01	1.74E+01	3.63E+01
9	RB-IPOP-CMA-ES	3.79E+00	3.21E+01	3.59E+01
10	MOS-SOCO2011	1.11E+01	1.92E+01	3.03E+01
11	PPSO	3.93E+00	1.73E+01	2.12E+01
12	DYYPO	5.93E–01	1.71E+01	1.77E+01
13	TLBO-FL	2.87E–02	1.64E+01	1.64E+01

f) CEC20 – j2020

Rank	BCM	Score 1	Score 2	Score
1	Halving	5.00E+01	4.27E+01	9.27E+01
		(0.5, 0.5, 42.1, 56.8)	(9.6, 24.3, 31.1, 35.)	
2	Random	4.19E+01	5.00E+01	9.19E+01
		(0., 5., 35.2, 59.7)	(11.3, 14.2, 27.6, 46.9)	
3	Reflection	4.35E+01	4.34E+01	8.69E+01
		(0., 5.6, 43.1, 51.4)	(10.9, 21.1, 28.9, 39.2)	
4	**Periodic**	**3.62E+01**	**3.59E+01**	**7.22E+01**
		(0., 4.3, 33.1, 62.6)	**(8.7, 19.8, 31.1, 40.3)**	
5	Clipping	3.68E+01	3.20E+01	6.88E+01
		(0., 6., 14., 80.)	(9.9, 19.8, 30.6, 39.7)	

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. Table 5.6 and Table 5.10 contain the Score and rank if the algorithms used their default BCMs. Unfortunately, the complete results of all competitors are available for the CEC17 benchmark only; therefore, Table 5.10 encompasses only the three top-ranking algorithms of the CEC20 competition. The best BCM for each algorithm was selected according to the ranks in Tables 5.5 - 5.5.

Tab. 5.7 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. 5.8 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. 5.9 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. 5.10 CEC20 – Score – Default BCM

Rank	Algorithm	Score 1	Score 2	Score
1	j2020	5.00E+01	4.56E+01	9.56E+01
2	IMODE	2.14E+01	5.00E+01	7.14E+01
3	AGSK	2.31E+01	4.56E+01	6.88E+01

Tab. 5.11 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. 5.12 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. 5.13 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. 5.14 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. 5.15 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

For the CEC17, tables 5.7, 5.8, and 5.9 represent the situations when only one algorithm selects its best variant of the BCM. If the rank is changed against Table 5.6, the original rank is shown in parentheses. The most noticeable difference is in Table 5.9, where the LSHADE-cnEpSin obtained the first rank. Table 5.14 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 5.10 - 5.15 and no change in the algorithms order was observed.

5.4.7 Key findings

The motivation behind this 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 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. Further, the attention was to the possibility of performance improvement by the use of alternate boundary control methods, as presented in the results of CEC17 benchmark participants, where:

- The LSHADE-cnEpSin algorithm would have won the CEC17 competition, if it did employ the *random* boundary control method.
- According to the scores as defined by the CEC17 benchmark, none of the three tested algorithms achieved the best results with the original BCM.

In other words, it was possible to improve the results of any of these algorithms by using a different BCM.

- Only five of the 12 participants reported on the employed BCM in the papers (albeit in two cases only partially, see Table 5.1). Moreover, the results of the two best-performing algorithms would be irreproducible without a detailed study of the source code.

The results of the second experiment, using the CEC20 benchmark seem to be less conclusive, as only three scenarios show statistically significant differences in performance (see Table 5.4). However, the scoring (Tables 5.5 d), 5.5 e) and 5.5 f)) seems to indicate not only a difference in performance, but a possible benefit of using other than default BCM.

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.

5.5 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.

5.5.1 Experimental setup

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. 5.7 and Fig. 5.8 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.

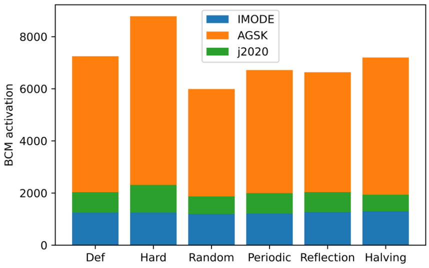
Figures Fig. 5.9 and Fig. 5.10 show different stacked graphs; these depict a particular BCM activation on a selected test function, dimension size, and algorithm. The x-axis represents the function evaluations (FEs) of the algorithm, and the y-axis shows the average number of activations (over 30 runs) for each problem dimension.

From the analysis of the first group of graphs (Fig. 5.7 and Fig. 5.8), it is concluded that the AGSK algorithm contributes to BCM activation the most across all test functions and both dimension sizes. 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.

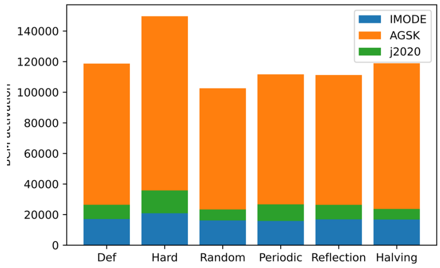
From Figures Fig. 5.9 and Fig. 5.10, it is evident that the number of BCM activations is dependent on the problem - specifically the test function. The number of BCM activations is almost equally distributed over the dimensions for f_2 (Fig. 5.9 (b)), whereas for test function f_1 (Fig. 5.9 (c)), the distribution is uneven, with the number of BCM activations for dimension two prevailing over the averaged algorithm run.

All figures are included in lower quality at the end of this thesis as supplementary data. For a more detailed view, high-resolution versions of the same figures are accessible from a designated webpage [1)] together with codes of the examined algorithms and their results.

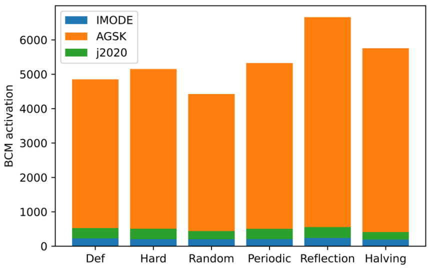
[1)] `https://go.fai.utb.cz/2023workshop`



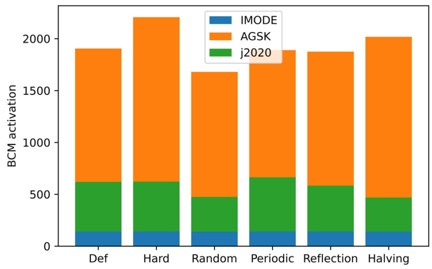
(a) f_1



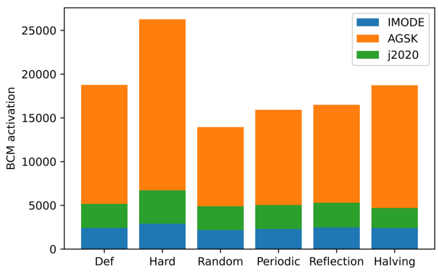
(b) f_2



(c) f_3

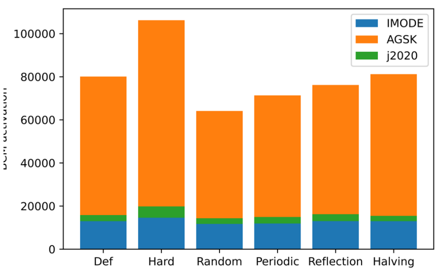


(d) f_4



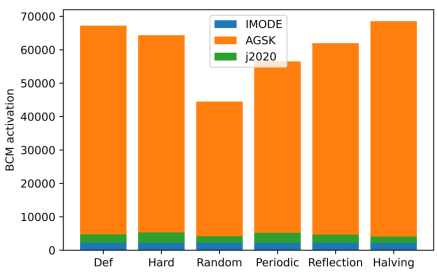
(e) f_5

Fig. 5.7 Total BCM activation for $dim = 5$. [116]



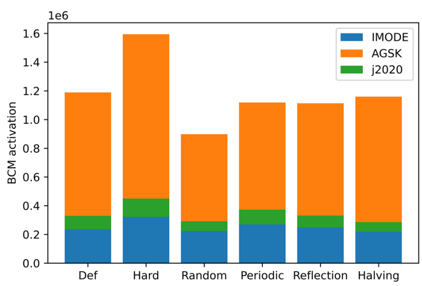
(a) f_1

(b) f_2



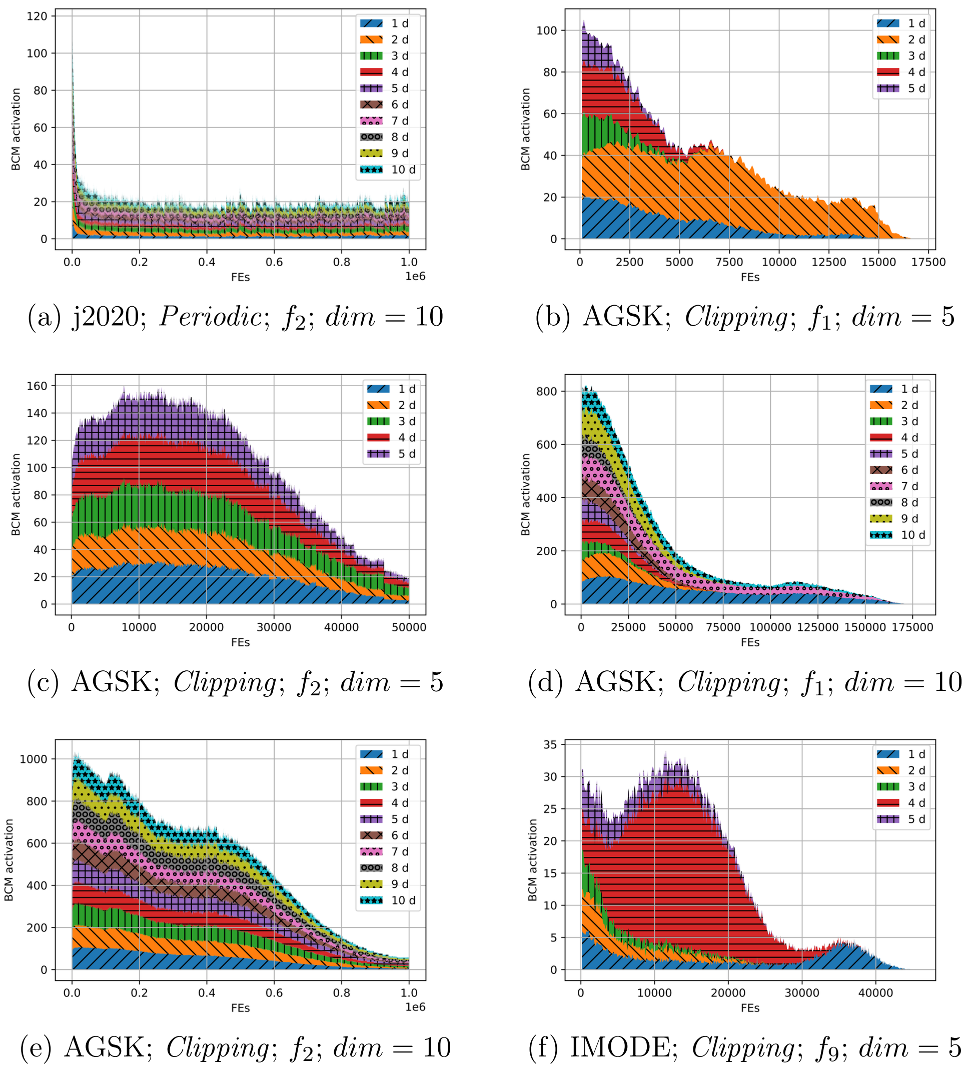
(c) f_3

(d) f_4



(e) f_5

Fig. 5.8 Total BCM activation for $dim = 10$. [116]



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

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

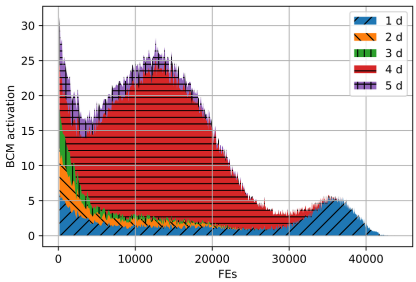
(c) AGSK; *Clipping*; f_2; $dim = 5$

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

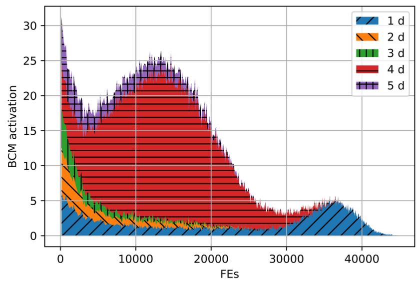
(e) AGSK; *Clipping*; f_2; $dim = 10$

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

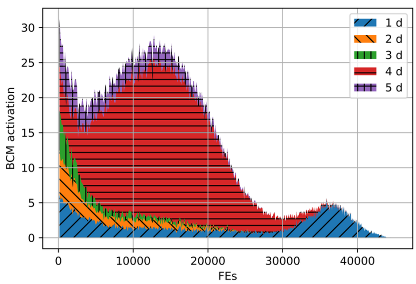
Fig. 5.9 BCM activation over the run of the algorithm. [116]



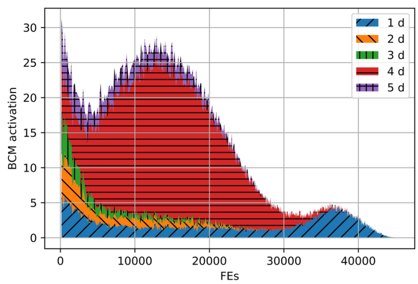
(a) IMODE; *Random*; f_9; $dim = 5$



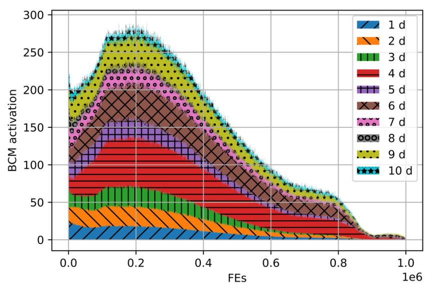
(b) IMODE; *Periodic*; f_9; $dim = 5$



(c) IMODE; *Reflection*; f_9; $dim = 5$



(d) IMODE; *Halving*; f_9; $dim = 5$



(e) IMODE; *Halving*; f_9; $dim = 10$

Fig. 5.10 BCM activation over the run of the algorithm. [116]

5.5.2 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 variations 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.

6 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.

7 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 in Section 4.1, where 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, detailed in Sections 5.1, 5.2 and 5.3, 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 5.4 and Section 5.5, 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 5.4, 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.

8 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 as outlined in subsection 3.6.

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.

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] Bäck, T. and Schwefel, H.-P. An overview of evolutionary algorithms for parameter optimization. *Evolutionary computation*. 1993, vol. 1, no. 1, pp. 1–23.

[6] Dasgupta, D. and Michalewicz, Z. *Evolutionary algorithms in engineering applications*. Springer Science & Business Media, 2013.

[7] Gendreau, M., Potvin, J.-Y. and others. *Handbook of metaheuristics*. vol. 2. Springer, 2010.

[8] Yang, X.-S. *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.

[9] 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.

[10] Raidl, G. R. A unified view on hybrid metaheuristics. In *International workshop on hybrid metaheuristics*, pp. 1–12. Springer, 2006.

[11] Kazikova, A., Pluhacek, M. and Senkerik, R. Why Tuning the Control Parameters of Metaheuristic Algorithms Is So Important for Fair Comparison? *MENDEL*. Dec. 2020, vol. 26,

no. 2, pp. 9–16. doi: 10.13164/mendel.2020.2.009. Available at: <http://ib-b2b.test.infv.eu/index.php/mendel/article/view/120>.

[12] López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Stützle, T. and Birattari, M. The irace package: Iterated Racing for Automatic Algorithm Configuration. *Operations Research Perspectives.* 2016, vol. 3, pp. 43–58. doi: 10.1016/j.orp.2016.09.002.

[13] Stein, N. and Bäck, T. *LLaMEA: A Large Language Model Evolutionary Algorithm for Automatically Generating Metaheuristics* [online]. 2024. Available at https://arxiv.org/abs/2405.20132.

[14] Stein, N., Vermetten, D., Kononova, A. V. and Bäck, T. *Explainable Benchmarking for Iterative Optimization Heuristics* [online]. 2024. Available at https://arxiv.org/abs/2401.17842.

[15] Kazikova, A., Pluhacek, M. and Senkerik, R. How Does the Number of Objective Function Evaluations Impact Our Understanding of Metaheuristics Behavior? *IEEE Access.* 2021, vol. 9, pp. 44032–44048. doi: 10.1109/ACCESS.2021.3066135.

[16] Wagdy, A., Hadi, A. A., Mohamed, A. K., Agrawal, P., Kumar, A. and Suganthan, P. N. Problem Definitions and Evaluation Criteria for the CEC 2021 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization. Technical report, Nanyang Technological University, Singapore, 2020.

[17] Hansen, N., Auger, A., Ros, R., Mersmann, O., Tušar, T. and Brockhoff, D. COCO: A platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software.* 2021, vol. 36, no. 1, pp. 114–144.

[18] Bartz-Beielstein, T. et al. *Benchmarking in Optimization: Best Practice and Open Issues* [online]. 2020. Available at https://arxiv.org/abs/2007.03488.

[19] LaTorre, A., Molina, D., Osaba, E., Poyatos, J., Del Ser, J. and Herrera, F. A prescription of methodological guidelines for comparing bio-inspired optimization algorithms. *Swarm and Evolutionary Computation*. 2021, vol. 67, pp. 100973. ISSN 2210-6502. doi: https://doi.org/10.1016/j.swevo.2021.100973.

[20] Fister Jr, I., Yang, X.-S., Fister, I., Brest, J. and Fister, D. *A brief review of nature-inspired algorithms for optimization* [online]. 2013. Available at `https://arxiv.org/abs/1307.4186`.

[21] Zang, H., Zhang, S. and Hapeshi, K. A review of nature-inspired algorithms. *Journal of Bionic Engineering*. 2010, vol. 7, no. 4, pp. S232–S237.

[22] Yang, X.-S. *Nature-inspired optimization algorithms*. Elsevier, 2014.

[23] Goss, S., Aron, S., Deneubourg, J.-L. and Pasteels, J. M. Self-organized shortcuts in the Argentine ant. *Naturwissenschaften*. 1989, vol. 76, no. 12, pp. 579–581.

[24] Greggers, U. and Menzel, R. Memory dynamics and foraging strategies of honeybees. *Behavioral Ecology and Sociobiology*. 1993, vol. 32, pp. 17–29.

[25] Rehmeyer, J. J. Mathematical lives of plants: Why plants grow in geometrically curious patterns. *Science News*. 2007, vol. 172, no. 3, pp. 42–45.

[26] Darwin, C. *On the Origin of Species by Means of Natural Selection*. Murray, 1859. or the Preservation of Favored Races in the Struggle for Life.

[27] Mendel, G. *Experiments in plant hybridization (1865)* [online]. 1996. Verhandlungen des naturforschenden Vereins Brün. Available at: `<http://old.esp.org/foundations/genetics/classical/gm-65-a.pdf>`.

[28] Canizo, J., Carrillo, J. and Rosado, J. Collective behavior of animals: Swarming and complex patterns. *Arbor*. 2010, vol. 186, no. 1035-1049, pp. 1.

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

[30] Campelo, F. and Aranha, C. EC Bestiary: A bestiary of evolutionary, swarm and other metaphor-based algorithms, June 2018. Available at: `<https://doi.org/10.5281/zenodo.1293352>`.

[31] Horst, R., Pardalos, P. M. and Van Thoai, N. *Introduction to global optimization*. Springer Science & Business Media, 2000.

[32] Črepinšek, M., Liu, S.-H. and Mernik, M. Exploration and exploitation in evolutionary algorithms: A survey. *ACM Comput. Surv.* jul 2013, vol. 45, no. 3. ISSN 0360-0300. doi: 10.1145/2480741.2480752. Available at: `<https://doi.org/10.1145/2480741.2480752>`.

[33] Holland, J. H. Genetic algorithms. *Scientific american*. 1992, vol. 267, no. 1, pp. 66–73.

[34] Kennedy, J. and Eberhart, R. Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4, pp. 1942–1948. IEEE, 1995.

[35] 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.

[36] Zhan, Z.-H., Zhang, J., Li, Y. and Shi, Y.-H. Orthogonal learning particle swarm optimization. *IEEE transactions on evolutionary computation*. 2010, vol. 15, no. 6, pp. 832–847.

[37] Storn, R. and Price, K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*. 1997, vol. 11, no. 4, pp. 341–359.

[38] Price, K. and Storn, R. Differential Evolution-a simple and efficient adaptive scheme for global optimization over continuous space. Technical report, Technical Report, International Computer Science Institute, 1995.

[39] Zelinka, I. SOMA—self-organizing migrating algorithm. In *New optimization techniques in engineering*, pp. 167–217. Springer, 2004.

[40] Zelinka, I. SOMA—self-organizing migrating algorithm. In *Self-Organizing Migrating Algorithm*, pp. 3–49. Springer, 2016.

[41] Rajwar, K., Deep, K. and Das, S. An exhaustive review of the metaheuristic algorithms for search and optimization: Taxonomy, applications, and open challenges. *Artificial Intelligence Review*. 2023, vol. 56, no. 11, pp. 13187–13257.

[42] Oliveto, P. S. and Witt, C. Improved runtime analysis of the simple genetic algorithm. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pp. 1621–1628, 2013.

[43] Oliveto, P. S. and Witt, C. On the runtime analysis of the simple genetic algorithm. *Theoretical Computer Science*. 2014, vol. 545, pp. 2–19.

[44] Oliveto, P. S. and Witt, C. Improved time complexity analysis of the simple genetic algorithm. *Theoretical Computer Science*. 2015, vol. 605, pp. 21–41.

[45] Doerr, C., Wang, H., Ye, F., Rijn, S. and Bäck, T. *IOHprofiler: A benchmarking and profiling tool for iterative optimization heuristics* [online]. 2018. Available at `https://arxiv.org/abs/1810.05281`.

[46] Zelinka, I., Davendra, D., Roman, S. and Roman, J. Do evolutionary algorithms dynamics create complex network structures? *Complex Systems*. 2011, vol. 20, no. 2, pp. 127.

[47] Zelinka, I., Skanderová, L., Davendra, D. D., Šenkeřík, R. and Oplatková, Z. Evolutionary dynamics and complex networks. In *MENDEL 2012*. Brno University of Technology, 2012.

[48] Lehre, P. K. and Oliveto, P. S. *Theoretical analysis of stochastic search algorithms* [online]. 2017. Available at `https://arxiv.org/abs/1709.00890`.

[49] Lehrexwid, P. K. and Oliveto, P. S. Runtime analysis of population-based evolutionary algorithms: introductory tutorial at GECCO 2020. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, pp. 458–494, 2020.

[50] Yang, X.-S. Metaheuristic optimization: algorithm analysis and open problems. In *International Symposium on Experimental Algorithms*, pp. 21–32. Springer, 2011.

[51] 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.

[52] Oldewage, E. T., Engelbrecht, A. P. and Cleghorn, C. W. Boundary constraint handling techniques for particle swarm optimization in high dimensional problem spaces. In *International Conference on Swarm Intelligence*, pp. 333–341. Springer, 2018.

[53] Clerc, M. Confinements and biases in particle swarm optimisation, 2006.

[54] Zhang, W.-J., Xie, X.-F. and Bi, D.-C. Handling boundary constraints for numerical optimization by particle swarm flying in periodic search space. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753)*, vol. 2, pp. 2307–2311. IEEE, 2004.

[55] Michalewicz, Z. and Schoenauer, M. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary computation*. 1996, vol. 4, no. 1, pp. 1–32.

[56] Koziel, S. and Michalewicz, Z. Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. *Evolutionary computation*. 1999, vol. 7, no. 1, pp. 19–44.

[57] Kononova, A. V., Caraffini, F. and Bäck, T. Differential evolution outside the box. *Information Sciences*. 2021, vol. 581, pp. 587–604. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2021.09.058. Available at: <https://www.sciencedirect.com/science/article/pii/S0020025521009956>.

[58] Bratton, D. and Kennedy, J. Defining a standard for particle swarm optimization. In *2007 IEEE swarm intelligence symposium*, pp. 120–127. IEEE, 2007.

[59] Sanaz, M., Werner, H. and Anja, W. Linear multi-objective particle swarm optimization. In *Stigmergic Optimization*, pp. 209–238. Springer, 2006.

[60] Hansen, N. *The CMA evolution strategy: A tutorial* [online]. 2016. Available at `https://arxiv.org/abs/1604.00772`.

[61] Ronkkonen, J., Kukkonen, S. and Price, K. V. Real-parameter optimization with differential evolution. In *2005 IEEE congress on evolutionary computation*, vol. 1, pp. 506–513. IEEE, 2005.

[62] Brest, J., Zumer, V. and Maucec, M. S. Self-adaptive differential evolution algorithm in constrained real-parameter optimization. In *2006 IEEE international conference on evolutionary computation*, pp. 215–222. IEEE, 2006.

[63] Price, K., Storn, R. M. and Lampinen, J. A. *Differential evolution: a practical approach to global optimization.* Springer Science & Business Media, 2006.

[64] Guo, S.-M., Tsai, J. S.-H., Yang, C.-C. and Hsu, P.-H. A self-optimization approach for L-SHADE incorporated with eigenvector-based crossover and successful-parent-selecting framework on CEC 2015 benchmark set. In *2015 IEEE congress on evolutionary computation (CEC)*, pp. 1003–1010. IEEE, 2015.

[65] Zhang, J. and Sanderson, A. C. JADE: adaptive differential evolution with optional external archive. *IEEE Transactions on evolutionary computation.* 2009, vol. 13, no. 5, pp. 945–958.

[66] Caraffini, F., Kononova, A. V. and Corne, D. Infeasibility and structural bias in differential evolution. *Information Sciences.* 2019, vol. 496, pp. 161–179.

[67] Vermetten, D., Stein, B., Caraffini, F., Minku, L. L. and Kononova, A. V. BIAS: A Toolbox for Benchmarking Structural Bias in the Continuous Domain. *IEEE Transactions on Evolutionary Computation*. 2022, vol. 26, no. 6, pp. 1380–1393. doi: 10.1109/TEVC.2022.3189848.

[68] Marini, F. and Walczak, B. Particle swarm optimization (PSO). A tutorial. *Chemometrics and Intelligent Laboratory Systems*. 2015, vol. 149, pp. 153–165.

[69] Eiben, A. E., Smith, J. E. and others. *Introduction to evolutionary computing*. Springer, 2003.

[70] Yang, X.-S. Firefly algorithm, stochastic test functions and design optimisation. *International journal of bio-inspired computation*. 2010, vol. 2, no. 2, pp. 78–84.

[71] Yang, X.-S. Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms*, pp. 169–178. Springer, 2009.

[72] Yang, X.-S. and Deb, S. Engineering optimisation by cuckoo search. *International Journal of Mathematical Modelling and Numerical Optimisation*. 2010, vol. 1, no. 4, pp. 330–343.

[73] Mallipeddi, R., Suganthan, P. N., Pan, Q.-K. and Tasgetiren, M. F. Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied soft computing*. 2011, vol. 11, no. 2, pp. 1679–1696.

[74] Yang, X.-S. and others. Firefly algorithm. *Nature-inspired metaheuristic algorithms*. 2008, vol. 20, pp. 79–90.

[75] 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.

[76] Kadavy, T., Pluhacek, M., Viktorin, A. and Senkerik, R. Comparing strategies for search space boundaries violation in PSO. In *In-*

ternational Conference on Artificial Intelligence and Soft Computing, pp. 655–664. Springer, 2017.

[77] 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.

[78] 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.

[79] 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.

[80] 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.

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

[82] 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.

[83] 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.

[84] Čičková, Z. and Lukáčik, M. Setting of Control Parameters of SOMA on the Base of Statistics. In *Self-Organizing Migrating Algorithm: Methodology and Implementation*, pp. 255–275. Springer, 2016.

[85] 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.

[86] 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.

[87] 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.

[88] 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.

[89] 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.

[90] Jagodziński, D. and Arabas, J. A differential evolution strategy. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1872–1876. IEEE, 2017.

[91] Maharana, D., Kommadath, R. and Kotecha, P. Dynamic Yin-Yang Pair Optimization and its performance on single objective real parameter problems of CEC 2017. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2390–2396. IEEE, 2017.

[92] Bujok, P. and Tvrdík, J. Enhanced individual-dependent differential evolution with population size adaptation. In *2017 IEEE congress on evolutionary computation (CEC)*, pp. 1358–1365. IEEE, 2017.

[93] Mohamed, A. W., Hadi, A. A., Fattouh, A. M. and Jambi, K. M. LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems. In *2017 IEEE Congress on evolutionary computation (CEC)*, pp. 145–152. IEEE, 2017.

[94] Sallam, K. M., Elsayed, S. M., Sarker, R. A. and Essam, D. L. Multi-method based orthogonal experimental design algorithm for solving CEC2017 competition problems. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1350–1357. IEEE, 2017.

[95] LaTorre, A. and Peña, J.-M. A comparison of three large-scale global optimizers on the CEC 2017 single objective real parameter numerical optimization benchmark. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1063–1070. IEEE, 2017.

[96] Tangherloni, A., Rundo, L. and Nobile, M. S. Proactive particles in swarm optimization: A settings-free algorithm for real-parameter single objective optimization problems. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1940–1947. IEEE, 2017.

[97] Biedrzycki, R. A version of IPOP-CMA-ES algorithm with midpoint for CEC 2017 single objective bound constrained problems. In *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1489–1494. IEEE, 2017.

[98] Kommadath, R. and Kotecha, P. Teaching learning based optimization with focused learning and its performance on CEC2017 functions. In *2017 IEEE congress on evolutionary computation (CEC)*, pp. 2397–2403. IEEE, 2017.

[99] 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.

[100] 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.

[101] 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.

[102] 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.

[103] 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.

[104] 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.

[105] Salgotra, R., Singh, U., Saha, S. and Gandomi, A. H. Improving cuckoo search: Incorporating changes for CEC 2017 and CEC 2020 benchmark problems. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–7. IEEE, 2020.

[106] Bolufé-Röhler, A. and Chen, S. A multi-population exploration-only exploitation-only hybrid on CEC-2020 single objective bound constrained problems. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8. IEEE, 2020.

[107] Stanovov, V., Akhmedova, S. and Semenkin, E. Ranked Archive Differential Evolution with Selective Pressure for CEC 2020 Numerical Opti-

mization. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–7. IEEE, 2020.

[108] Viktorin, A., Senkerik, R., Pluhacek, M., Kadavy, T. and Zamuda, A. DISH-XX Solving CEC2020 Single Objective Bound Constrained Numerical optimization Benchmark. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8. IEEE, 2020.

[109] Bujok, P., Kolenovsky, P. and Janisch, V. Eigenvector Crossover in jDE100 Algorithm. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–6. IEEE, 2020.

[110] Biswas, P. P. and Suganthan, P. N. Large initial population and neighborhood search incorporated in LSHADE to solve CEC2020 benchmark problems. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–7. IEEE, 2020.

[111] Jou, Y.-C., Wang, S.-Y., Yeh, J.-F. and Chiang, T.-C. Multi-population Modified L-SHADE for Single Objective Bound Constrained optimization. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8. IEEE, 2020.

[112] Kadavy, T., Pluhacek, M., Viktorin, A. and Senkerik, R. Self-Organizing Migrating Algorithm with Clustering-Aided Migration. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, GECCO '20, pp. 1441–1447, New York, NY, USA, 2020. Association for Computing Machinery. doi: 10.1145/3377929.3398129. Available at: `<https://doi.org/10.1145/3377929.3398129>`. ISBN 9781450371278.

[113] 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.

[114] 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.

[115] 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.

[116] 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

[P.1] 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.

[P.2] 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.

[P.3] 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.

[P.4] 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.

[P.5] 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.

[P.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.

[P.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

[P.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.

[P.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.

[P.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.

[P.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.

[P.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.

[P.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.

Conference papers

[P.14] PLUHACEK, Michal, KADAVY, Tomas, SENKERIK, Roman, VIKTORIN, Adam, ZELINKA, Ivan. Comparing Selected PSO Modifications on CEC 15 Benchmark Set. 2016 IEEE Symposium Series on Computational Intelligence (SSCI) Proceedings. Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2016. ISBN 978-1-5090-4240-1.

[P.15] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas, ZAMUDA, Ales. Distance Based Parameter Adaptation for Differential Evolution. 2017 IEEE Symposium Series on Computational Intelligence (SSCI) Proceedings. New Jersey, Piscataway : IEEE, 2017, pp. 2612-2618. ISBN 978-1-5386-2725-9.

[P.16] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas. Towards better population sizing for differential evolution through active population analysis with complex network. Advances in Intelligent Systems and Computing. Berlin : Springer Verlag, 2017, pp. 225-235. ISSN 2194-5357. ISBN 978-331961565-3.

[P.17] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas. The Influence of Archive Size to SHADE. Artificial Intelligence Trends in Intelligent Systems, CSOC2017, VOL 1 Book Series: Advances in Intelligent Systems and Computing. Cham : Springer International Publishing AG, 2017, pp. 517-527. ISSN 2194-5357. ISBN 978-3-319-57260-4.

[P.18] VIKTORIN, Adam, PLUHACEK, Michal, SENKERIK, Roman, KADAVY, Tomas. Detecting potential design weaknesses in shade through network feature analysis. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in

Bioinformatics). Heidelberg : Springer-Verlag Berlin, 2017, pp. 662-673. ISSN 0302-9743. ISBN 978-3-319-59649-5.

[P.19] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas. SHADE Algorithm Dynamic Analyzed Through Complex Network. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Berlin : Springer Verlag, 2017, pp. 666-677. ISSN 0302-9743. ISBN 978-331962388-7.

[P.20] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas. L-SHADE Algorithm with Distance Based Parameter Adaptation. Lecture Notes in Electrical Engineering. Berlin : Springer Verlag, 2017, pp. 69-80. ISSN 1876-1100. ISBN 978-331969813-7.

[P.21] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas. Shade mutation strategy analysis via dynamic simulation in complex network. Proceedings - 31st European Conference on Modelling and Simulation, ECMS 2017. Madrid : European Council for Modelling and Simulation, 2017, pp. 299-305. ISBN 978-099324404-9.

[P.22] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas. Archive analysis in SHADE. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Heidelberg : Springer-Verlag Berlin, 2017, pp. 688-699. ISSN 0302-9743. ISBN 978-3-319-59059-2.

[P.23] SENKERIK, Roman, VIKTORIN, Adam, PLUHACEK, Michal, KADAVY, Tomas, KOMINKOVA OPLATKOVA, Zuzana. Performance Comparison of Differential Evolution Driving Analytic Programming for Regression. 2017 IEEE Symposium Series on Computational Intelligence (SSCI) Proceedings. New Jersey, Piscataway : IEEE, 2017, pp. 2358-2365. ISBN 978-1-5386-2725-9.

[P.24] SENKERIK, Roman, VIKTORIN, Adam, PLUHACEK, Michal, KADAVY, Tomas, ZELINKA, Ivan. Hybridization of analytic programming and differential evolution for time series prediction. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Heidelberg : Springer-Verlag Berlin, 2017, pp. 686-698. ISSN 0302-9743. ISBN 978-3-319-59649-5.

[P.25] SENKERIK, Roman, PLUHACEK, Michal, VIKTORIN, Adam, KOMINKOVA OPLATKOVA, Zuzana, KADAVY, Tomas. Simulation of chaotic dynamics for chaos based optimization-an extended study. Proceedings - 31st European Conference on Modelling and Simulation, ECMS 2017. Madrid : European Council for Modelling and Simulation, 2017, pp. 319-325. ISBN 978-099324404-9.

[P.26] SENKERIK, Roman, PLUHACEK, Michal, VIKTORIN, Adam, KADAVY, Tomas. On the Randomization of Indices Selection for Differential Evolution. Artificial Intelligence Trends in Intelligent Systems, CSOC2017, VOL 1 Book Series: Advances in Intelligent Systems and Computing. Cham : Springer International Publishing AG, 2017, pp. 537-547. ISSN 2194-5357. ISBN 978-3-319-57260-4.

[P.27] SENKERIK, Roman, VIKTORIN, Adam, PLUHACEK, Michal, KADAVY, Tomas, ZELINKA, Ivan. Differential evolution driven analytic programming for prediction. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Heidelberg : Springer-Verlag Berlin, 2017, pp. 676-687. ISSN 0302-9743. ISBN 978-3-319-59059-2.

[P.28] PLUHACEK, Michal, SENKERIK, Roman, VIKTORIN, Adam, KADAVY, Tomas, ZELINKA, Ivan. PSO with Partial Population Restart Based on Complex Network Analysis. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Heidelberg : Springer-Verlag Berlin, 2017, pp. 183-192. ISSN 0302-9743. ISBN 978-3-319-59649-5.

[P.29] PLUHACEK, Michal, SENKERIK, Roman, VIKTORIN, Adam, KADAVY, Tomas. Exploring the Shortest Path in PSO Communication Network. 2017 IEEE Symposium Series on Computational Intelligence

(SSCI) Proceedings. New Jersey, Piscataway : IEEE, 2017, pp. 1494-1499. ISBN 978-1-5386-2725-9.

[P.30] PLUHACEK, Michal, SENKERIK, Roman, VIKTORIN, Adam, KADAVY, Tomas. How Chaotic Sequences and Generator Sequencing Affect the Particle Trajectory in PSO. 2017 IEEE Symposium Series on Computational Intelligence (SSCI) Proceedings. New Jersey, Piscataway : IEEE, 2017, pp. 2570-2577. ISBN 978-1-5386-2725-9.

[P.31] PLUHACEK, Michal, KADAVY, Tomas, SENKERIK, Roman. ARPSO and fk-PSO on CEC 15 Benchmark – Comparative Study. AIP Conference Proceedings. Maryland : American Institute of Physics Inc., 2017. ISSN 0094-243X. ISBN 978-073541538-6.

[P.32] PLUHACEK, Michal, SENKERIK, Roman, VIKTORIN, Adam, KADAVY, Tomas. PSO with Attractive Search Space Border Points. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Heidelberg : Springer-Verlag Berlin, 2017, pp. 665-675. ISSN 0302-9743. ISBN 978-3-319-59059-2.

[P.33] PLUHACEK, Michal, SENKERIK, Roman, VIKTORIN, Adam, KADAVY, Tomas. UNCOVERING COMMUNICATION DENSITY IN PSO USING COMPLEX NETWORK. Proceedings - 31st European Conference on Modelling and Simulation, ECMS 2017. Madrid : European Council for Modelling and Simulation, 2017, pp. 306-312. ISBN 978-099324404-9.

[P.34] PLUHACEK, Michal, SENKERIK, Roman, VIKTORIN, Adam, KADAVY, Tomas, ZELINKA, Ivan. USING COMPLEX NETWORK VISUALIZATION AND ANALYSIS FOR UNCOVERING THE INNER DYNAMICS OF PSO ALGORITHM. Mendel. Brno : VUT Brno, 2017, pp. 87-94. ISSN 1803-3814.

[P.35] KADAVY, Tomas, PLUHACEK, Michal, VIKTORIN, Adam, SENKERIK, Roman. Comparing strategies for search space boundaries violation in PSO. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Heidelberg : Springer-Verlag Berlin, 2017, pp. 655-664. ISSN 0302-9743. ISBN 978-3-319-59059-2.

[P.36] KADAVY, Tomas, PLUHACEK, Michal, VIKTORIN, Adam, SENKERIK, Roman. Partial Population Restart of Firefly Algorithm Using Complex Network Analysis. 2017 IEEE Symposium Series on Computational Intelligence (SSCI) Proceedings. New Jersey, Piscataway : IEEE, 2017, pp. 2505-2511. ISBN 978-1-5386-2725-9.

[P.37] KADAVY, Tomas, PLUHACEK, Michal, VIKTORIN, Adam, SENKERIK, Roman. Hypersphere universe boundary method comparison on hclpso and PSO. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Heidelberg : Springer-Verlag Berlin, 2017, pp. 173-182. ISSN 0302-9743. ISBN 978-3-319-59649-5.

[P.38] KADAVY, Tomas, PLUHACEK, Michal, VIKTORIN, Adam, SENKERIK, Roman. Firework algorithm dynamics simulated and analyzed with the aid of complex network. Proceedings - 31st European Conference on Modelling and Simulation, ECMS 2017. Madrid : European Council for Modelling and Simulation, 2017, pp. 313-318. ISBN 978-099324404-9.

[P.39] KADAVY, Tomas, PLUHACEK, Michal, VIKTORIN, Adam, SENKERIK, Roman. Dynamic of firework algorithm analyzed with complex network. Mendel. Brno : VUT Brno, 2017, pp. 79-86. ISSN 1803-3814.

[P.40] KADAVY, Tomas, PLUHACEK, Michal, VIKTORIN, Adam, SENKERIK, Roman. Comparing border strategies for roaming particles on single and Multi-Swarm PSO. Artificial Intelligence Trends in Intelligent Systems, CSOC2017, VOL 1 Book Series: Advances in Intelligent Systems and Computing. Cham : Springer International Publishing AG, 2017, pp. 528-536. ISSN 2194-5357. ISBN 978-3-319-57260-4.

[P.41] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas. COMPARATIVE STUDY OF THE DISTANCE/IMPROVEMENT BASED SHADE. Proceedings - 32nd European Conference on Modelling and Simulation, ECMS 2018. Madrid : European Council for Modelling and Simulation, 2018, pp. 163-169. ISSN 2522-2414. ISBN 978-0-9932440-6-3.

[P.42] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas. Addressing Premature Convergence with Distance based Parameter Adaptation in SHADE. International Conference on Systems, Signals, and Image Processing. Los Alamitors : IEEE Computer Society, 2018. ISSN 21578672. ISBN 978-153866979-2.

[P.43] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas. On the prolonged exploration of distance based parameter adaptation in SHADE. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) Volume 10841. Berlin : Springer Verlag, 2018, pp. 561-571. ISSN 03029743. ISBN 978-331991252-3.

[P.44] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas. Cluster Occurrence in the DbL-SHADE Population. 2018 IEEE Congress on Evolutionary Computation, CEC 2018 - Proceedings. Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2018, pp. 1-8. ISBN 978-150906017-7.

[P.45] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas. How distance based parameter adaptation affects population diversity. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) Volume 10835. Berlin : Springer Verlag, 2018, pp. 307-319. ISSN 03029743. ISBN 978-331991640-8.

[P.46] SENKERIK, Roman, PLUHACEK, Michal, VIKTORIN, Adam, KADAVY, Tomas, JANOSTIK, Jakub, KOMINKOVA OPLATKOVA, Zuzana. A Review On The Simulation of Social Networks Inside Heuristic Algorithms. Proceedings - 32nd European Conference on Modelling and Simulation, ECMS 2018. Madrid : European Council for Modelling and Simulation, 2018, pp. 176-182. ISSN 2522-2414. ISBN 978-0-9932440-6-3.

[P.47] SENKERIK, Roman, VIKTORIN, Adam, PLUHACEK, Michal, KADAVY, Tomas, ZELINKA, Ivan. How unconventional chaotic pseudo-random generators influence population diversity in differential evolution. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) Volume 10841. Berlin : Springer Verlag, 2018, pp. 524-535. ISSN 03029743. ISBN 978-331991252-3.

[P.48] SENKERIK, Roman, VIKTORIN, Adam, PLUHACEK, Michal, KADAVY, Tomas, ZELINKA, Ivan. Differential Evolution for Constrained Industrial Optimization. Lecture Notes in Electrical Engineering. Springer Nature, 2018, pp. 123-132. ISSN 18761100. ISBN 978-331969813-7.

[P.49] SENKERIK, Roman, VIKTORIN, Adam, PLUHACEK, Michal, KADAVY, Tomas. Population diversity analysis for the chaotic based selection of individuals in differential evolution. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) Volume 10835. Berlin : Springer Verlag, 2018, pp. 283-294. ISSN 03029743. ISBN 978-331991640-8.

[P.50] SENKERIK, Roman, VIKTORIN, Adam, PLUHACEK, Michal, KADAVY, Tomas. On the Population Diversity for the Chaotic Differential Evolution. 2018 IEEE Congress on Evolutionary Computation, CEC 2018 - Proceedings. Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2018. ISBN 978-150906017-7.

[P.51] SENKERIK, Roman, VIKTORIN, Adam, PLUHACEK, Michal, KADAVY, Tomas, KOMINKOVA OPLATKOVA, Zuzana. Differential

Evolution and Chaotic Series. International Conference on Systems, Signals, and Image Processing. Los Alamitors : IEEE Computer Society, 2018. ISSN 21578672. ISBN 978-153866979-2.

[P.52] SPACEK, Lubos, VOJTESEK, Jiri, GAZDOS, Frantisek, KADAVY, Tomas. Ball & Plate Model for Robotic System. Proceedings - 32nd European Conference on Modelling and Simulation, ECMS 2018. Madrid : European Council for Modelling and Simulation, 2018, pp. 226-231. ISSN 2522-2414. ISBN 978-0-9932440-6-3.

[P.53] PLUHACEK, Michal, SENKERIK, Roman, VIKTORIN, Adam, KADAVY, Tomas. STUDY ON VELOCITY CL AMPING IN PSO USING CEC'13 BENCHMARK. Proceedings - 32nd European Conference on Modelling and Simulation, ECMS 2018. Madrid : European Council for Modelling and Simulation, 2018, pp. 150-155. ISSN 2522-2414. ISBN 978-0-9932440-6-3.

[P.54] PLUHACEK, Michal, SENKERIK, Roman, VIKTORIN, Adam, KADAVY, Tomas, ZELINKA, Ivan. Chaos Driven PSO with Attractive Search Space Border Points. 2018 IEEE Congress on Evolutionary Computation, CEC 2018 - Proceedings. Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2018, pp. 1147-1152. ISBN 978-150906017-7.

[P.55] PLUHACEK, Michal, SENKERIK, Roman, VIKTORIN, Adam, KADAVY, Tomas. Particle Swarm Optimization with Single Particle Repulsivity for Multi-modal Optimization. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) Volume 10841. Berlin : Springer Verlag, 2018, pp. 486-494. ISSN 03029743. ISBN 978-331991252-3.

[P.56] PLUHACEK, Michal, SENKERIK, Roman, VIKTORIN, Adam, KADAVY, Tomas, ZELINKA, Ivan. A Review of Real-world Applications of Particle Swarm Optimization Algorithm. Lecture Notes in Electrical Engineering. Springer Nature, 2018, pp. 115-122. ISSN 1876-1100. ISBN 978-331969813-7.

[P.57] KOVAR, Stanislav, VALOUCH, Jan, KADAVY, Tomas, POSPISILIK, Martin, ADAMEK, Milan. Enclosure shielding effectiveness calculation using SHADE algorithm. 2018 19th International Scientific Conference on Electric Power Engineering, EPE 2018 - Proceedings. Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2018. ISBN 978-153864611-3.

[P.58] KADAVY, Tomas, PLUHACEK, Michal, VIKTORIN, Adam, SENKERIK, Roman. Comparing Boundary Control Methods for Firefly Algorithm. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) Volume 10835. Berlin : Springer Verlag, 2018, pp. 163-173. ISSN 03029743. ISBN 978-331991640-8.

[P.59] KADAVY, Tomas, PLUHACEK, Michal, VIKTORIN, Adam, SENKERIK, Roman. Orthogonal Learning Firefly Algorithm. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Berlin : Springer Verlag, 2018, pp. 315-326. ISSN 03029743. ISBN 978-331992638-4.

[P.60] KADAVY, Tomas, PLUHACEK, Michal, VIKTORIN, Adam, SENKERIK, Roman. Boundary Strategies For Firefly Algorithm Analysed Using CEC'17 Benchmark. Proceedings - 32nd European Conference on Modelling and Simulation, ECMS 2018. Madrid : European Council for Modelling and Simulation, 2018, pp. 170-175. ISSN 2522-2414. ISBN 978-0-9932440-6-3.

[P.61] KADAVY, Tomas, PLUHACEK, Michal, VIKTORIN, Adam, SENKERIK, Roman. Multi-swarm Optimization Algorithm Based on Firefly and Particle Swarm Optimization Techniques. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) Volume 10841. Berlin : Springer Verlag, 2018, pp. 405-416. ISSN 03029743. ISBN 978-331991252-3.

[P.62] KADAVY, Tomas, PLUHACEK, Michal, VIKTORIN, Adam, SENKERIK, Roman. Firefly Algorithm: Enhanced Version with Partial Population Restart Using Complex Network Analysis. Lecture Notes in Electrical Engineering. Springer Nature, 2018, pp. 59-68. ISSN 18761100. ISBN 978-331969813-7.

[P.63] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas. Enhanced archive for SHADE. Advances in Intelligent Systems and Computing, Volume 837. Berlin : Springer Verlag, 2019, pp. 40-55. ISSN 21945357. ISBN 978-331997887-1.

[P.64] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas. Analyzing Control Parameters in DISH. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Berlin : Springer Verlag, 2019, pp. 519-529. ISSN 03029743. ISBN 978-3-030-20911-7.

[P.65] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas. Distance vs. Improvement based parameter adaptation in SHADE. Advances in Intelligent Systems and Computing, Volume 764. Berlin : Springer Verlag, 2019, pp. 455-464. ISSN 2194-5357. ISBN 978-331991188-5.

[P.66] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas, ZAMUDA, Ales. DISH Algorithm Solving the CEC 2019 100-Digit Challenge. 2019 IEEE Congress on Evolutionary Computation (CEC). Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2019, pp. 1-6. ISBN 978-1-72812-153-6.

[P.67] SENKERIK, Roman, VIKTORIN, Adam, KADAVY, Tomas, PLUHACEK, Michal, KAZIKOVA, Anezka, DIEP, Quoc Bao, ZELINKA, Ivan. Population Diversity Analysis in Adaptive Differential Evolution Variants with Unconventional Randomization Schemes. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Berlin : Springer Verlag, 2019, pp. 506-518. ISSN 03029743. ISBN 978-3-030-20911-7.

[P.68] SENKERIK, Roman, VIKTORIN, Adam, PLUHACEK, Michal, KADAVY, Tomas, KOMINKOVA OPLATKOVA, Zuzana. On the applicability of random and the best solution driven metaheuristics for analytic programming and time series regression. Advances in Intelligent Systems and Computing, Volume 764. Berlin : Springer Verlag, 2019, pp. 489-498. ISSN 2194-5357. ISBN 978-331991188-5.

[P.69] SENKERIK, Roman, PLUHACEK, Michal, VIKTORIN, Adam, KADAVY, Tomas, KOMINKOVA OPLATKOVA, Zuzana. Randomization of individuals selection in differential evolution. Advances in Intelligent Systems and Computing, Volume 837. Berlin : Springer Verlag, 2019, pp. 180-191. ISSN 21945357. ISBN 978-331997887-1.

[P.70] SENKERIK, Roman, KADAVY, Tomas, VIKTORIN, Adam, PLUHACEK, Michal. Ensemble of Strategies and Perturbation Parameter Based SOMA for Constrained Technological Design Optimization Problem. 2019 IEEE Congress on Evolutionary Computation, CEC 2019 - Proceedings. Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2019, pp. 2872-2877. ISBN 978-1-72812-153-6.

[P.71] PLUHACEK, Michal, SENKERIK, Roman, VIKTORIN, Adam, KADAVY, Tomas. Why Simple Population Restart Does Not Work in PSO. Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018. Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2019, pp. 770-776. ISBN 978-1-5386-9276-9.

[P.72] PLUHACEK, Michal, HRDY, Michal, VIKTORIN, Adam, KADAVY, Tomas, SENKERIK, Roman. Spiral extrusion die design using modified differential evolution algorithm. Mendel. Brno : Brno University of Technology, 2019, pp. 121-130. ISSN 1803-3814.

[P.73] KADAVY, Tomas, PLUHACEK, Michal, SENKERIK, Roman, VIKTORIN, Adam. On the Performance Significance of Boundary Strategies for Firefly Algorithm. Proceedings of the 2018 IEEE Symposium

Series on Computational Intelligence, SSCI 2018. Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2019, pp. 765-769. ISBN 978-1-5386-9276-9.

[P.74] KADAVY, Tomas, PLUHACEK, Michal, VIKTORIN, Adam, SENKERIK, Roman. Firefly Algorithm Enhanced by Orthogonal Learning. Advances in Intelligent Systems and Computing, Volume 764. Berlin : Springer Verlag, 2019, pp. 477-488. ISSN 2194-5357. ISBN 978-331991188-5.

[P.75] KADAVY, Tomas, PLUHACEK, Michal, SENKERIK, Roman, VIKTORIN, Adam. Introducing Self-Adaptive Parameters to Self-organizing Migrating Algorithm. 2019 IEEE Congress on Evolutionary Computation (CEC). Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2019, pp. 2908-2914. ISBN 978-1-72812-153-6.

[P.76] KADAVY, Tomas, KOVAR, Stanislav, PLUHACEK, Michal, VIKTORIN, Adam, SENKERIK, Roman. Evolutionary Algorithms Applied to a Shielding Enclosure Design. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Berlin : Springer Verlag, 2019, pp. 445-455. ISSN 03029743. ISBN 978-3-030-20911-7.

[P.77] KADAVY, Tomas, PLUHACEK, Michal, SENKERIK, Roman, VIKTORIN, Adam. The Ensemble of Strategies and Perturbation Parameter in Self-organizing Migrating Algorithm Solving CEC 2019 100-Digit Challenge. 2019 IEEE Congress on Evolutionary Computation (CEC). Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2019, pp. 372-375. ISBN 978-1-72812-153-6.

[P.78] KADAVY, Tomas, KOVAR, Stanislav, SENKERIK, Roman, PLUHACEK, Michal. On the design of a front-face grid for shielding enclosure using evolutionary computations. 2019 Photonics and Electromagnetics Research Symposium - Fall, PIERS - Fall 2019 -

Proceedings. Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2019, pp. 443-450. ISBN 978-1-72815-304-9.

[P.79] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas, JASEK, Roman. A lightweight SHADE-based algorithm for global optimization - liteSHADE. Lecture Notes in Electrical Engineering. Berlin : Springer Verlag, 2020, pp. 197-206. ISSN 1876-1100. ISBN 978-3-030-14906-2.

[P.80] VIKTORIN, Adam, SENKERIK, Roman, PLUHACEK, Michal, KADAVY, Tomas, ZAMUDA, Ales. DISH-XX solving CEC2020 single objective bound constrained numerical optimization benchmark. 2020 IEEE Congress on Evolutionary Computation, CEC 2020 - Conference Proceedings. Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2020. ISBN 978-172816929-3.

[P.81] SENKERIK, Roman, KADAVY, Tomas, VIKTORIN, Adam, PLUHACEK, Michal. Ensemble of strategies and perturbation parameter based SOMA for optimal stabilization of chaotic oscillations. GECCO 2020 Companion - Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion. New York : Association for Computing Machinery, 2020, pp. 1468-1475. ISBN 978-145037127-8.

[P.82] SENKERIK, Roman, VIKTORIN, Adam, KADAVY, Tomas, PLUHACEK, Michal, ZELINKA, Ivan. Insight into adaptive differential evolution variants with unconventional randomization schemes. Communications in Computer and Information Science. Londyn : Springer Nature, 2020, pp. 177-188. ISSN 1865-0929. ISBN 978-3-030-37837-0.

[P.83] SENKERIK, Roman, VIKTORIN, Adam, KADAVY, Tomas, PLUHACEK, Michal, ZELINKA, Ivan. Is Chaotic Randomization Advantageous for Higher Dimensional Optimization Problems?. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Berlin

: Springer Science and Business Media Deutschland GmbH, 2020, pp. 423-434. ISSN 03029743. ISBN 978-303061533-8.

[P.84] KAZIKOVA, Anezka, PLUHACEK, Michal, KADAVY, Tomas, SENKERIK, Roman. Introducing the run support strategy for the bison algorithm. Lecture Notes in Electrical Engineering. Berlin : Springer Verlag, 2020, pp. 272-282. ISSN 1876-1100. ISBN 978-3-030-14906-2.

[P.85] PLUHACEK, Michal, SENKERIK, Roman, VIKTORIN, Adam, KADAVY, Tomas. Self-Organizing Migrating Algorithm with non-binary perturbation. Communications in Computer and Information Science. Londyn : Springer Nature, 2020, pp. 43-57. ISSN 1865-0929. ISBN 978-3-030-37837-0.

[P.86] KADAVY, Tomas, PLUHACEK, Michal, VIKTORIN, Adam, SENKERIK, Roman. Self-organizing migrating algorithm with clustering-aided migration. GECCO 2020 Companion - Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion. New York : Association for Computing Machinery, 2020, pp. 1441-1447. ISBN 978-145037127-8.

[P.87] KADAVY, Tomas, PLUHACEK, Michal, SENKERIK, Roman, VIKTORIN, Adam. Boundary Strategies for Self-organizing Migrating Algorithm Analyzed Using CEC'17 Benchmark. Communications in Computer and Information Science. Londyn : Springer Nature, 2020, pp. 58-69. ISSN 1865-0929. ISBN 978-3-030-37837-0.

[P.88] KADAVY, Tomas, PLUHACEK, Michal, VIKTORIN, Adam, SENKERIK, Roman. SOMA-CL for competition on single objective bound constrained numerical optimization benchmark: A competition entry on single objective bound constrained numerical optimization at the genetic and evolutionary computation conference (GECCO) 2020. GECCO 2020 Companion - Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion. New York : Association for Computing Machinery, 2020, pp. 9-10. ISBN 978-145037127-8.

[P.89] ULRICH, Adam, VIKTORIN, Adam, PLUHACEK, Michal, KADAVY, Tomas, KRNAVEK, Jan. Novelty Search in Particle Swarm Optimization. 2021 IEEE Symposium Series on Computational Intelligence, SSCI 2021 - Proceedings. Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2021. ISBN 978-172819048-8.

[P.90] PLUHACEK, Michal, KAZIKOVA, Anezka, KADAVY, Tomas, VIKTORIN, Adam, SENKERIK, Roman. Explaining SOMA: The relation of stochastic perturbation to population diversity and parameter space coverage. GECCO 2021 Companion - Proceedings of the 2021 Genetic and Evolutionary Computation Conference Companion. New York : Association for Computing Machinery, 2021, pp. 1944-1952. ISBN 978-145038351-6.

[P.91] PLUHACEK, Michal, VIKTORIN, Adam, KADAVY, Tomas, KAZIKOVA, Anezka. On the common population diversity measures in metaheuristics and their limitations. 2021 IEEE Symposium Series on Computational Intelligence, SSCI 2021 - Proceedings. Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2021, pp. 1. ISBN 978-172819048-8.

[P.92] KADAVY, Tomas, PLUHACEK, Michal, VIKTORIN, Adam, SENKERIK, Roman. SOMA-CLP for competition on bound constrained single objective numerical optimization benchmark: a competition entry on bound constrained single objective numerical optimization at the genetic and evolutionary computation conference (GECCO) 2021. GECCO 2021 Companion - Proceedings of the 2021 Genetic and Evolutionary Computation Conference Companion. New York : Association for Computing Machinery, 2021, pp. 11-12. ISBN 978-145038351-6.

[P.93] KADAVY, Tomas, PLUHACEK, Michal, VIKTORIN, Adam, SENKERIK, Roman. Self-organizing migrating algorithm with clustering-aided migration and adaptive perturbation vector control. GECCO 2021 Companion - Proceedings of the 2021 Genetic and Evo-

lutionary Computation Conference Companion. New York : Association for Computing Machinery, 2021, pp. 1916-1922. ISBN 978-145038351-6.

[P.94] KADAVY, Tomas, VIKTORIN, Adam, PLUHACEK, Michal, KOVAR, Stanislav. On Modifications Towards Improvement of the Exploitation Phase for SOMA Algorithm with Clustering-aided Migration and Adaptive Perturbation Vector Control. 2021 IEEE Symposium Series on Computational Intelligence, SSCI 2021 - Proceedings. Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2021, pp. 1-8. ISBN 978-172819048-8.

[P.95] PLUHACEK, Michal, KADAVY, Tomas, KAZIKOVA, Anezka, VIKTORIN, Adam, SENKERIK, Roman. Inner dynamics of particle swarm optimization interpreted by complex network analysis. 2022 IEEE Workshop on Complexity in Engineering, COMPENG 2022. Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2022. ISBN 978-1-72817-124-1.

[P.96] KOVAR, Stanislav, KAVANKOVA, Iva, RENZLER, Michael, VALOUCH, Jan, KADAVY, Tomas, MAIR, Dominik. Evolutionary Algorithms as a Tool for Shielding Design. 2022 IEEE International Symposium on Electromagnetic Compatibility and Signal/Power Integrity, EMCSI 2022. Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2022, pp. 575-579. ISBN 978-1-66540-929-2.

[P.97] KADAVY, Tomas, PLUHACEK, Michal, VIKTORIN, Adam, KAZIKOVA, Anezka, SENKERIK, Roman. Exploring clustering in SOMA. 2022 IEEE Workshop on Complexity in Engineering, COMPENG 2022. Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2022, pp. 1-5. ISBN 978-1-72817-124-1.

[P.98] SENKERIK, Roman, KADAVY, Tomas, JANKU, Peter, PLUHACEK, Michal, GUZOWSKI, Hubert, PEKAR, Libor, MATUSU, Radek, VIKTORIN, Adam, SMOŁKA, Maciej, BYRSKI, Aleksander, KOMINKOVA OPLATKOVA, Zuzana. Maximizing efficiency: A comparative study of SOMA variants and constraint handling methods for time delay system optimization. GECCO 2023 Companion - Proceedings of the 2023 Genetic and Evolutionary Computation Conference Companion. New York : Association for Computing Machinery, Inc, 2023, pp. 1821-1829. ISBN 979-840070120-7.

[P.99] SENKERIK, Roman, BYRSKI, Alexander, GUZOWSKI, Hubert, JANKU, Peter, KADAVY, Tomas, KOMINKOVA OPLATKOVA, Zuzana, MATUSU, Radek, PLUHACEK, Michal, PEKAR, Libor, SMOLKA, Maciej, VIKTORIN, Adam. Efficient Time-Delay System Optimization with Auto-Configured Metaheuristics. 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC). New Jersey, Piscataway : IEEE, 2023, pp. 1084-1089. ISSN 1062-922X. ISBN 979-8-3503-3703-7.

[P.100] PLUHACEK, Michal, KAZIKOVA, Anezka, KADAVY, Tomas, VIKTORIN, Adam, SENKERIK, Roman. Leveraging Large Language Models for the generation of novel metaheuristic optimization algorithms. GECCO 2023 Companion - Proceedings of the 2023 Genetic and Evolutionary Computation Conference Companion. New York : Association for Computing Machinery, Inc, 2023, pp. 1812-1820. ISBN 979-840070120-7.

[P.101] PLUHACEK, Michal, KAZIKOVA, Anezka, VIKTORIN, Adam, KADAVY, Tomas, SENKERIK, Roman. Investigating the Potential of AI-Driven Innovations for Enhancing Differential Evolution in Optimization Tasks. 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC). New Jersey, Piscataway : IEEE, 2023, pp. 1070-1075. ISSN 1062-922X. ISBN 979-8-3503-3703-7.

[P.102] MATUSIKOVA, Marcela, PLUHACEK, Michal, KADAVY, Tomas, VIKTORIN, Adam, SENKERIK, Roman. Exploring Adaptive Components of SOMA. GECCO 2023 Companion - Proceedings of the 2023 Genetic and Evolutionary Computation Conference Companion. New York : Association for Computing Machinery, Inc, 2023, pp. 1803 - 1811. ISBN 979-840070120-7.

[P.103] KOVAR, Stanislav, KADAVY, Tomas, KAVANKOVA, Iva, VALOUCH, Jan, NEMEC, Jan. Metal-Coated Building Blocks as a Tool for Creating Electromagnetic Shielding. 2023 IEEE Symposium on Electromagnetic Compatibility and Signal/Power Integrity, EMC+SIPI 2023. Piscataway, New Jersey : Institute of Electrical and Electronics Engineers Inc., 2023, pp. 47-52. ISBN 979-8-3503-0976-8.

[P.104] KADAVY, Tomas, PLUHACEK, Michal, VIKTORIN, Adam, SENKERIK, Roman. Exploring the Frequency of Boundary Control Methods Activation in Metaheuristic Algorithms. GECCO 2023 Companion - Proceedings of the 2023 Genetic and Evolutionary Computation Conference Companion. New York : Association for Computing Machinery, Inc, 2023, pp. 2330-2336. ISBN 979-840070120-7.

[P.105] KADAVY, Tomas, VIKTORIN, Adam, KAZIKOVA, Anezka, PLUHACEK, Michal, SENKERIK, Roman. Impact of Boundary Control Methods on Bound-constrained Optimization Benchmarking. GECCO 2023 Companion - Proceedings of the 2023 Genetic and Evolutionary Computation Conference Companion. New York : Association for Computing Machinery, Inc, 2023, pp. 25-26. ISBN 979-840070120-7.

Chapters in books

[P.106] PLUHACEK, Michal, SENKERIK, Roman, VIKTORIN, Adam, KADAVY, Tomas. Complex Networks in Particle Swarm. Evolutionary Algorithms, Swarm Dynamics and Complex Networks. Heidelberg : Springer-Verlag GmbH, 2018, pp. 145-159. ISBN 978-3-662-55661-0.

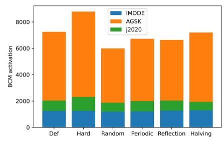
[P.107] SPOLAOR, Simone, GRIBAUDO, Marco, IACONO, Mauro, KADAVY, Tomas, KOMINKOVA OPLATKOVA, Zuzana, MAURI, Giancarlo, PILANA, Sabri, SENKERIK, Roman, STOJANOVIC, Natalija, TURUNEN, Esko, VIKTORIN, Adam, VITABILE, Salvatore, ZAMUDA, Ales, NOBILE, Marco Salvatore. Towards Human Cell Simulation. High-Performance Modelling and Simulation for Big Data Applications. Cham : Springer, 2019, pp. 221-249. ISBN 978-3-030-16271-9.

CURRICULUM VITAE

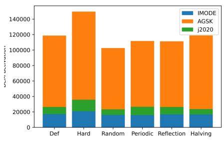
Personal Information:	
Name	Tomas Kadavy
Date of birth	08 September 1990
Nationality	Czech
Contact	phone: +420 731 861 391 email: kadavy@utb.cz
Education:	
2010 - 2014	Tomas Bata University in Zlín, Faculty of Applied Informatics, Information and Control Technologies (Bc.)
2014 - 2016	Tomas Bata University 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)

LIST OF APPENDICES

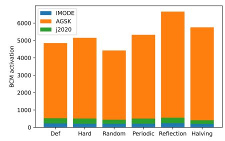
APPENDIX A: TOTAL BCM ACTIVATION FREQUENCY RESULTS



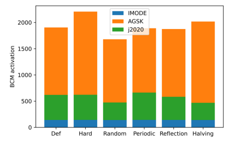
$dim = 5, f_1$



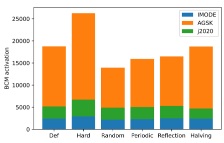
$dim = 5, f_2$



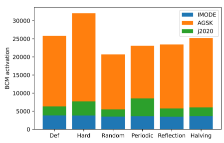
$dim = 5, f_3$



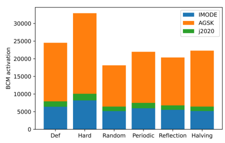
$dim = 5, f_4$



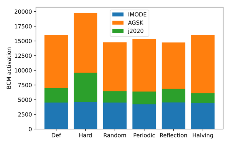
$dim = 5, f_5$



$dim = 5, f_6$

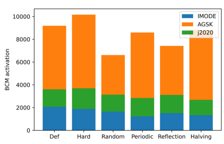


$dim = 5, f_7$

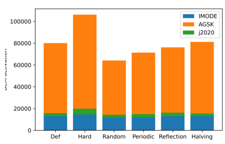


$dim = 5, f_8$

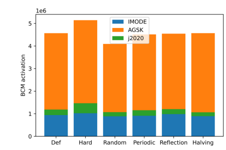
$dim = 5, f_9$



$dim = 5, f_{10}$



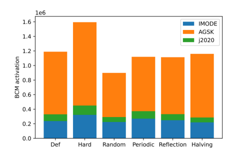
$dim = 10, f_1$



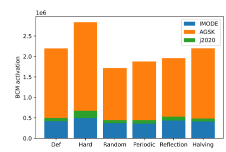
$dim = 10, f_2$

$dim = 10, f_3$

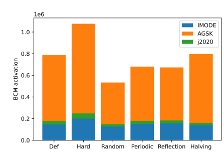
$dim = 10, f_4$



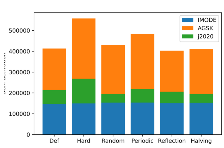
$dim = 10, f_5$



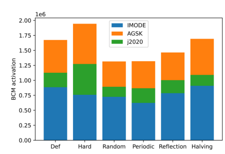
$dim = 10, f_6$



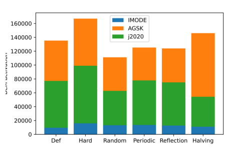
$dim = 10, f_7$



$dim = 10, f_8$



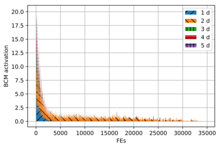
$dim = 10, f_9$



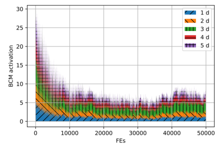
$dim = 10, f_{10}$

APPENDIX B: BCM ACTIVATION DYNAMICS

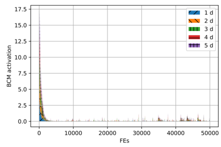
Algorithm j2020, BCM Default



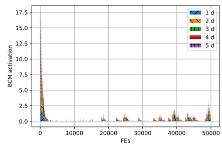
$dim = 5, f_1$



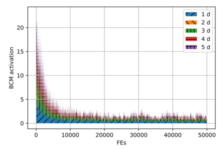
$dim = 5, f_2$



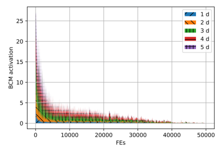
$dim = 5, f_3$



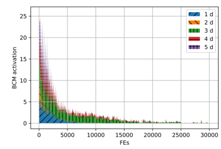
$dim = 5, f_4$



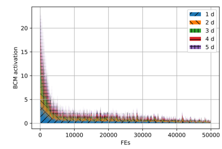
$dim = 5, f_5$



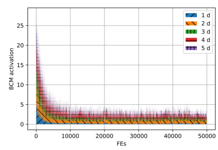
$dim = 5, f_6$



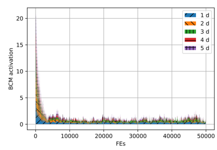
$dim = 5, f_7$



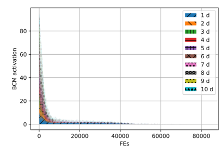
$dim = 5, f_8$



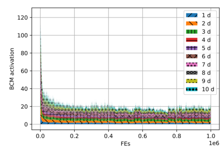
$dim = 5, f_9$



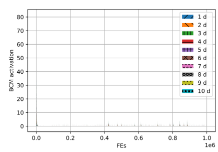
$dim = 5, f_{10}$



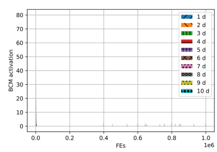
$dim = 10, f_1$



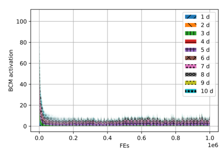
$dim = 10, f_2$



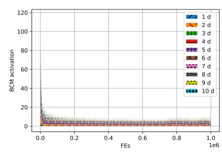
$dim = 10, f_3$



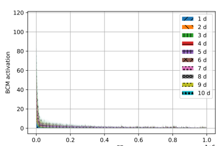
$dim = 10, f_4$



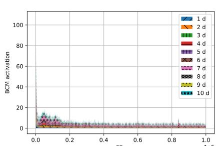
$dim = 10, f_5$



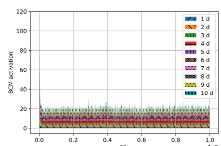
$dim = 10, f_6$



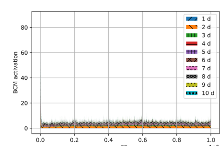
$dim = 10, f_7$



$dim = 10, f_8$

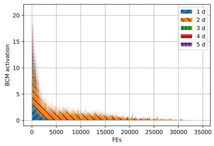


$dim = 10, f_9$

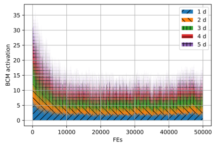


$dim = 10, f_{10}$

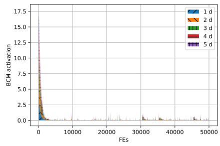
Algorithm j2020, BCM Clipping



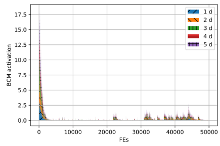
$dim = 5, f_1$



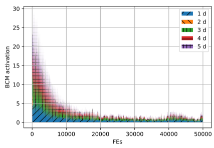
$dim = 5, f_2$



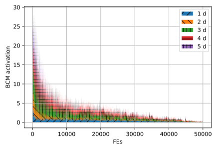
$dim = 5, f_3$



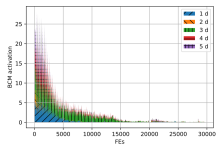
$dim = 5, f_4$



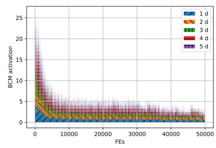
$dim = 5, f_5$



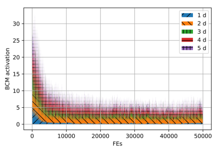
$dim = 5, f_6$



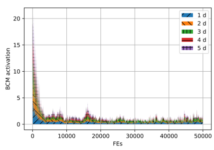
$dim = 5, f_7$



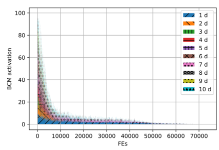
$dim = 5, f_8$



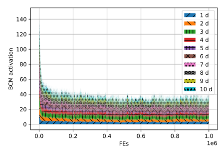
$dim = 5, f_9$



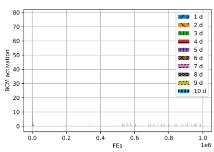
$dim = 5, f_{10}$



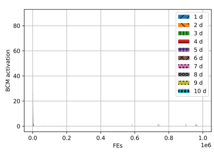
$dim = 10, f_1$



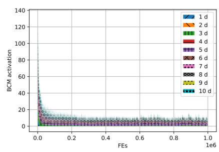
$dim = 10, f_2$



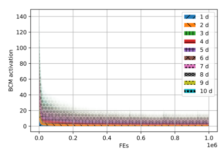
$dim = 10, f_3$



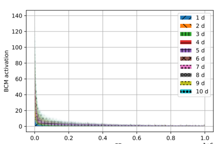
$dim = 10, f_4$



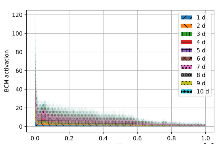
$dim = 10, f_5$



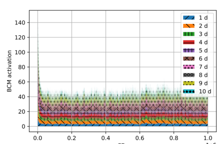
$dim = 10, f_6$



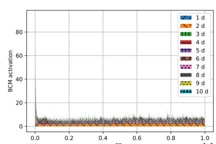
$dim = 10, f_7$



$dim = 10, f_8$

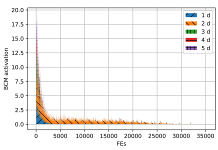


$dim = 10, f_9$

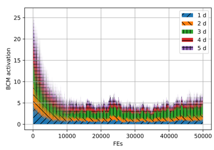


$dim = 10, f_{10}$

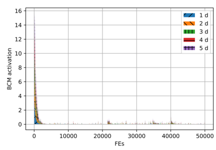
Algorithm j2020, BCM Random



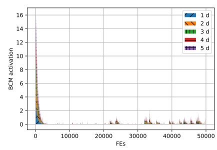
$dim = 5, f_1$



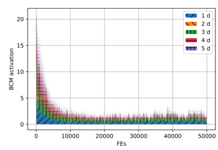
$dim = 5, f_2$



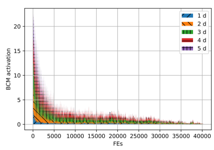
$dim = 5, f_3$



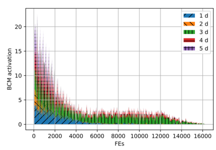
$dim = 5, f_4$



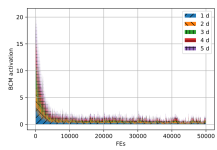
$dim = 5, f_5$



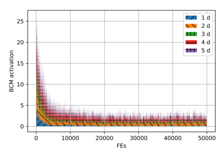
$dim = 5, f_6$



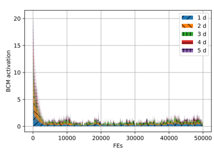
$dim = 5, f_7$



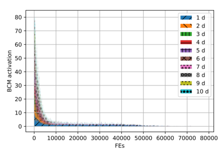
$dim = 5, f_8$



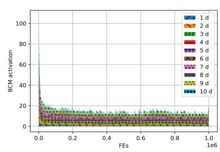
$dim = 5, f_9$



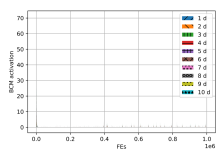
$dim = 5, f_{10}$



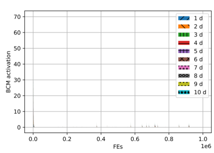
$dim = 10, f_1$



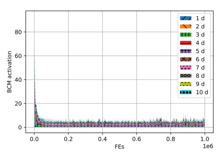
$dim = 10, f_2$



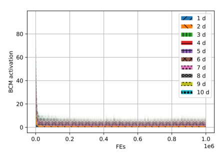
$dim = 10, f_3$



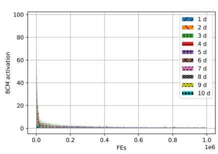
$dim = 10, f_4$



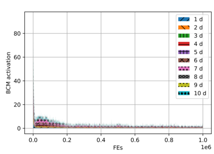
$dim = 10, f_5$



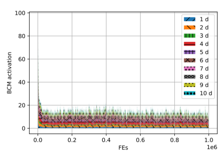
$dim = 10, f_6$



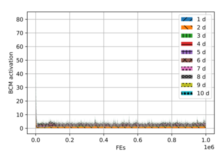
$dim = 10, f_7$



$dim = 10, f_8$

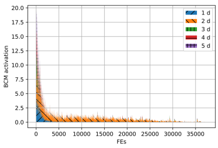


$dim = 10, f_9$

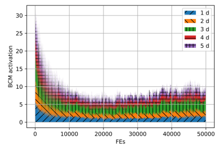


$dim = 10, f_{10}$

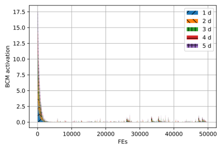
Algorithm j2020, BCM Periodic



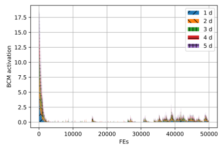
$dim = 5, f_1$



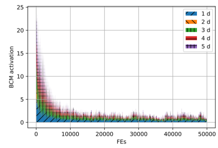
$dim = 5, f_2$



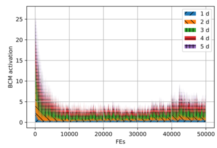
$dim = 5, f_3$



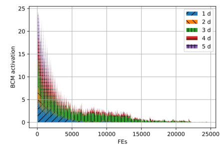
$dim = 5, f_4$



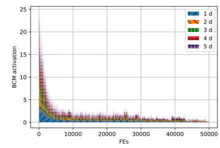
$dim = 5, f_5$



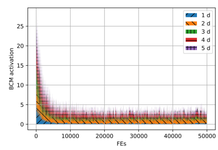
$dim = 5, f_6$



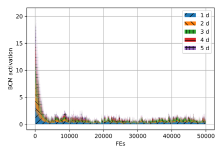
$dim = 5, f_7$



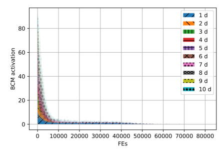
$dim = 5, f_8$



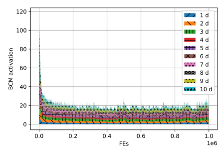
$dim = 5, f_9$



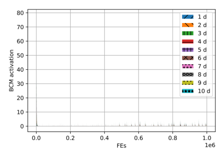
$dim = 5, f_{10}$



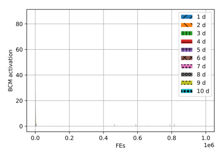
$dim = 10, f_1$



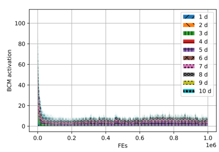
$dim = 10, f_2$



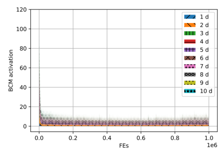
$dim = 10, f_3$



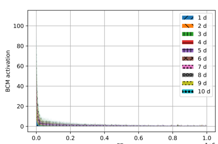
$dim = 10, f_4$



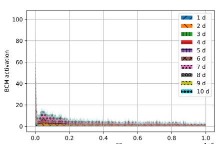
$dim = 10, f_5$



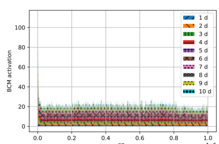
$dim = 10, f_6$



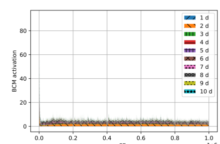
$dim = 10, f_7$



$dim = 10, f_8$

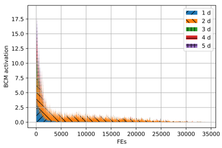


$dim = 10, f_9$

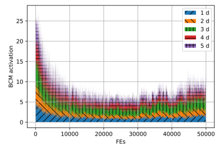


$dim = 10, f_{10}$

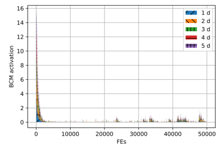
Algorithm j2020, BCM Reflection



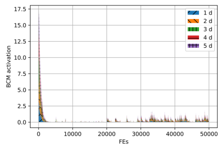
$dim = 5, f_1$



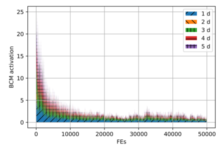
$dim = 5, f_2$



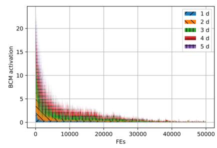
$dim = 5, f_3$



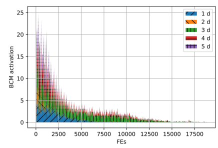
$dim = 5, f_4$



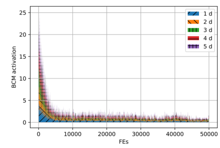
$dim = 5, f_5$



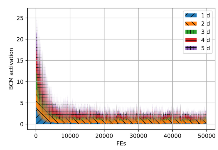
$dim = 5, f_6$



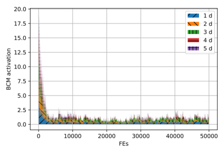
$dim = 5, f_7$



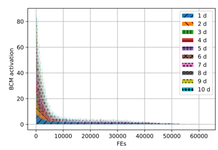
$dim = 5, f_8$



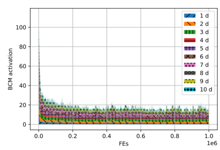
$dim = 5, f_9$



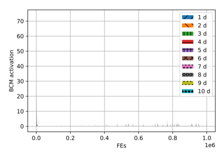
$dim = 5, f_{10}$



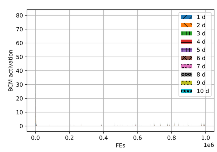
$dim = 10, f_1$



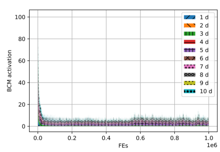
$dim = 10, f_2$



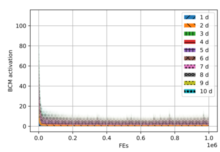
$dim = 10, f_3$



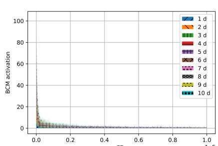
$dim = 10, f_4$



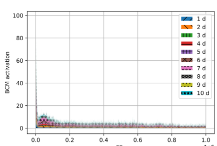
$dim = 10, f_5$



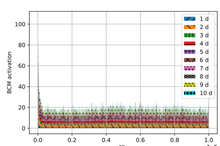
$dim = 10, f_6$



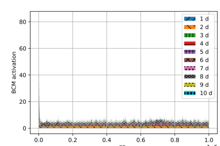
$dim = 10, f_7$



$dim = 10, f_8$

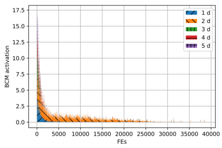


$dim = 10, f_9$

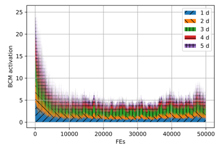


$dim = 10, f_{10}$

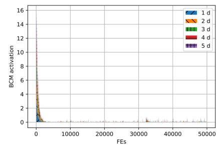
Algorithm j2020, BCM Halving



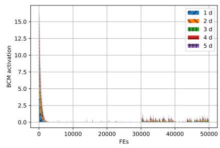
$dim = 5, f_1$



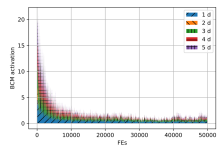
$dim = 5, f_2$



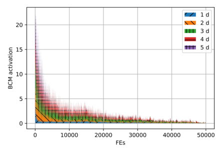
$dim = 5, f_3$



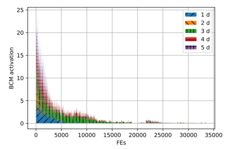
$dim = 5, f_4$



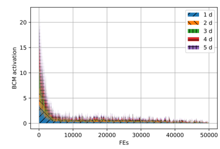
$dim = 5, f_5$



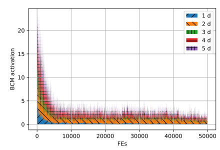
$dim = 5, f_6$



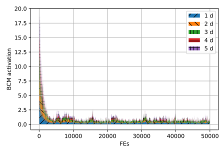
$dim = 5, f_7$



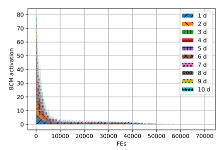
$dim = 5, f_8$



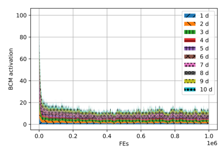
$dim = 5, f_9$



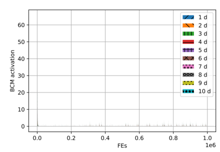
$dim = 5, f_{10}$



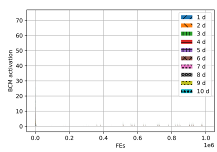
$dim = 10, f_1$



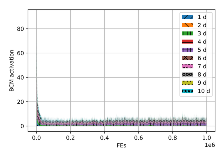
$dim = 10, f_2$



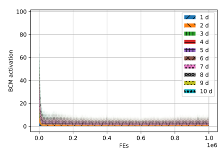
$dim = 10, f_3$



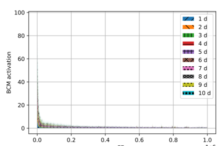
$dim = 10, f_4$



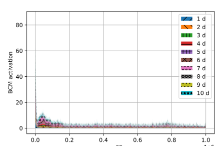
$dim = 10, f_5$



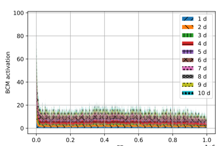
$dim = 10, f_6$



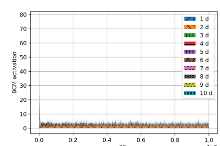
$dim = 10, f_7$



$dim = 10, f_8$

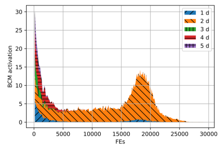


$dim = 10, f_9$

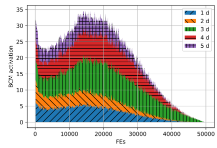


$dim = 10, f_{10}$

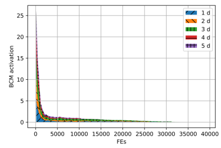
Algorithm IMODE, BCM Default



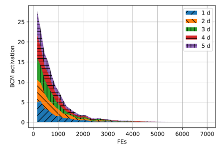
$dim = 5, f_1$



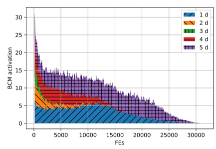
$dim = 5, f_2$



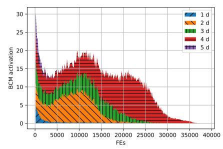
$dim = 5, f_3$



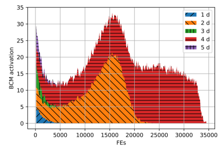
$dim = 5, f_4$



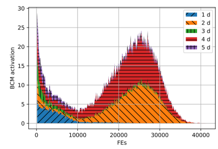
$dim = 5, f_5$



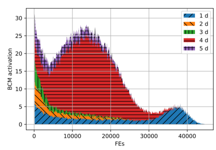
$dim = 5, f_6$



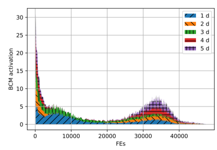
$dim = 5, f_7$



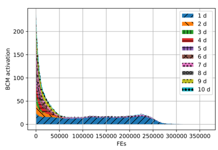
$dim = 5, f_8$



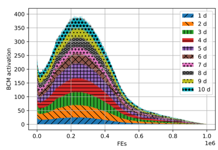
$dim = 5, f_9$



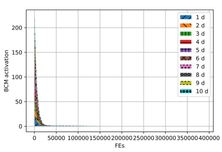
$dim = 5, f_{10}$



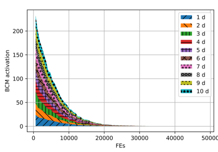
$dim = 10, f_1$



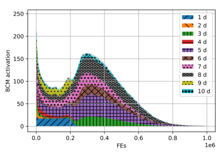
$dim = 10, f_2$



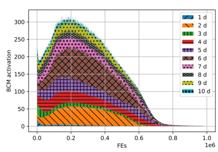
$dim = 10, f_3$



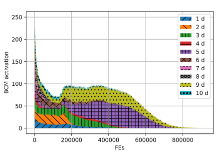
$dim = 10, f_4$



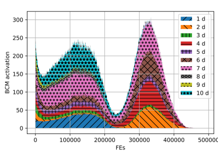
$dim = 10, f_5$



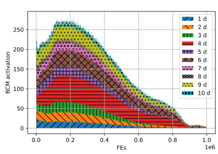
$dim = 10, f_6$



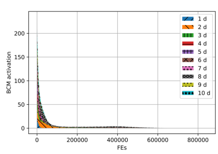
$dim = 10, f_7$



$dim = 10, f_8$

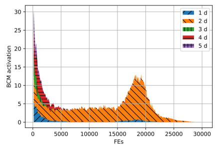


$dim = 10, f_9$

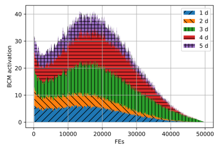


$dim = 10, f_{10}$

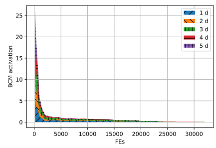
Algorithm IMODE, BCM Clipping



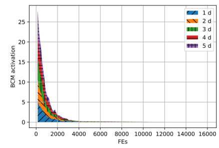
$dim = 5, f_1$



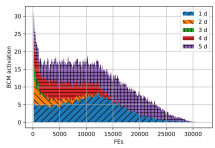
$dim = 5, f_2$



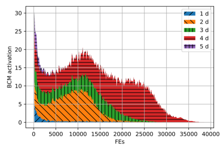
$dim = 5, f_3$



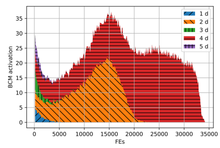
$dim = 5, f_4$



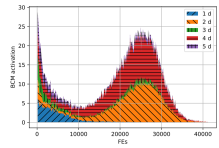
$dim = 5, f_5$



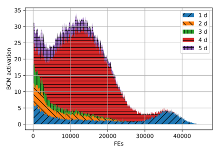
$dim = 5, f_6$



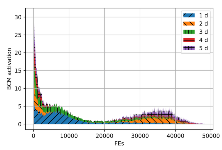
$dim = 5, f_7$



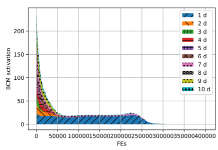
$dim = 5, f_8$



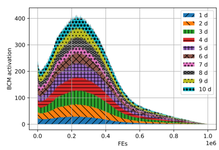
$dim = 5, f_9$



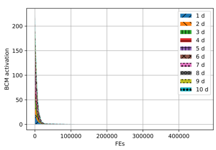
$dim = 5, f_{10}$



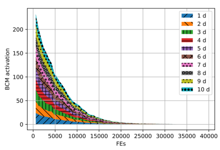
$dim = 10, f_1$



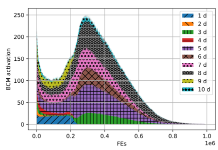
$dim = 10, f_2$



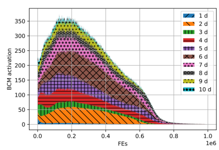
$dim = 10, f_3$



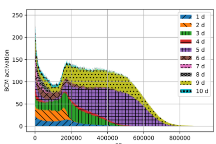
$dim = 10, f_4$



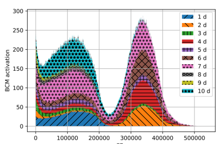
$dim = 10, f_5$



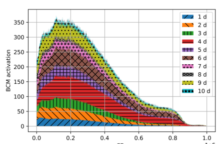
$dim = 10, f_6$



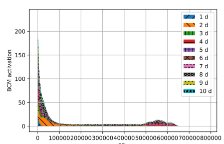
$dim = 10, f_7$



$dim = 10, f_8$

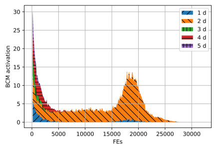


$dim = 10, f_9$

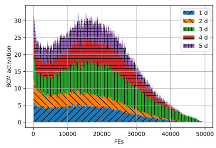


$dim = 10, f_{10}$

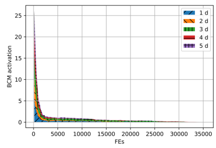
Algorithm IMODE, BCM Random



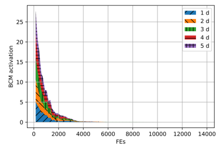
$dim = 5, f_1$



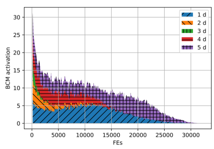
$dim = 5, f_2$



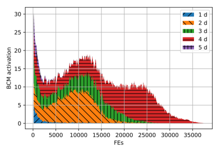
$dim = 5, f_3$



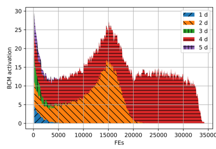
$dim = 5, f_4$



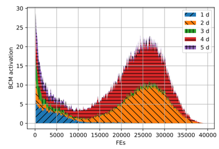
$dim = 5, f_5$



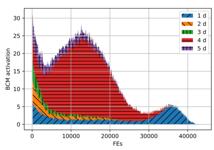
$dim = 5, f_6$



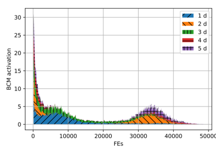
$dim = 5, f_7$



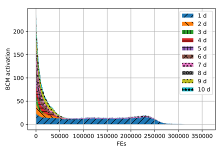
$dim = 5, f_8$



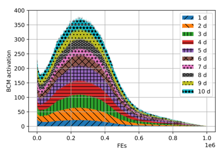
$dim = 5, f_9$



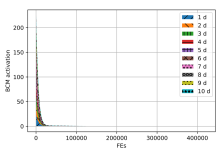
$dim = 5, f_{10}$



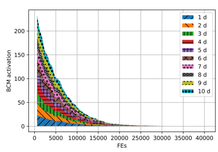
$dim = 10, f_1$



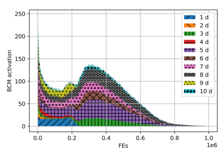
$dim = 10, f_2$



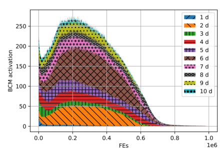
$dim = 10, f_3$



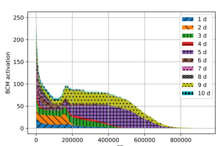
$dim = 10, f_4$



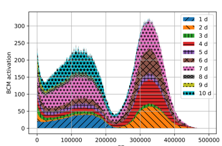
$dim = 10, f_5$



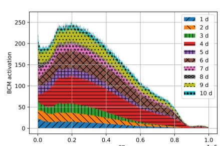
$dim = 10, f_6$



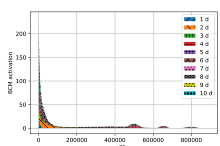
$dim = 10, f_7$



$dim = 10, f_8$

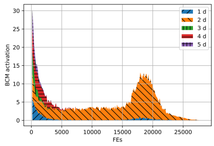


$dim = 10, f_9$

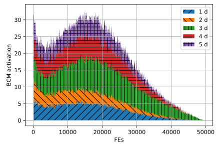


$dim = 10, f_{10}$

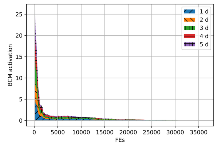
Algorithm IMODE, BCM Periodic



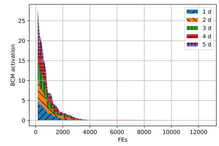
$dim = 5, f_1$



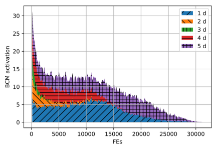
$dim = 5, f_2$



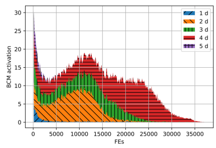
$dim = 5, f_3$



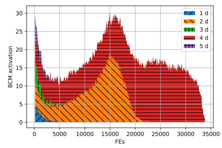
$dim = 5, f_4$



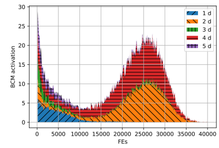
$dim = 5, f_5$



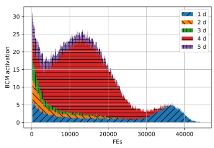
$dim = 5, f_6$



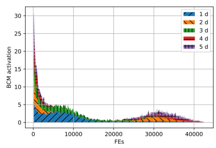
$dim = 5, f_7$



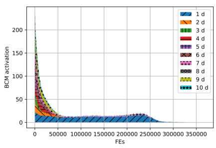
$dim = 5, f_8$



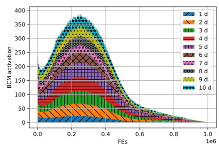
$dim = 5, f_9$



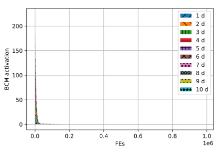
$dim = 5, f_{10}$



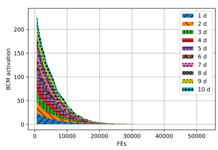
$dim = 10, f_1$



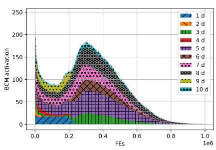
$dim = 10, f_2$



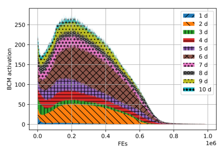
$dim = 10, f_3$



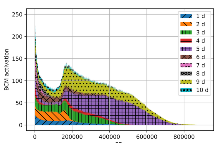
$dim = 10, f_4$



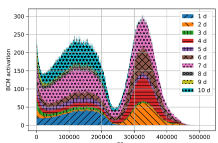
$dim = 10, f_5$



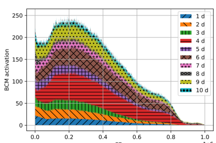
$dim = 10, f_6$



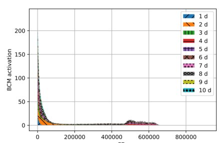
$dim = 10, f_7$



$dim = 10, f_8$

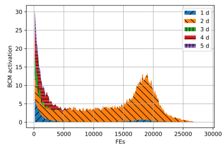


$dim = 10, f_9$

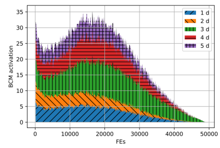


$dim = 10, f_{10}$

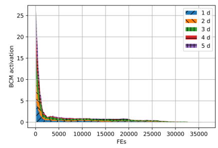
Algorithm IMODE, BCM Reflection



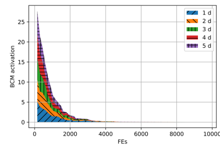
$dim = 5, f_1$



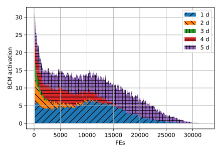
$dim = 5, f_2$



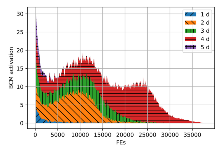
$dim = 5, f_3$



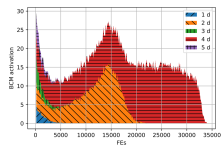
$dim = 5, f_4$



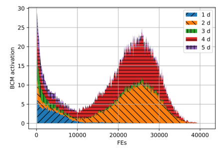
$dim = 5, f_5$



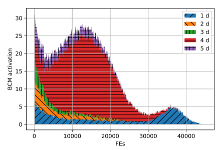
$dim = 5, f_6$



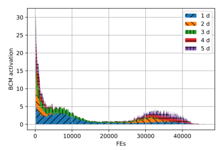
$dim = 5, f_7$



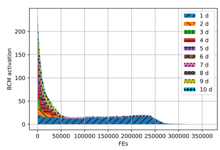
$dim = 5, f_8$



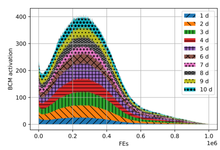
$dim = 5, f_9$



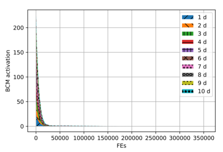
$dim = 5, f_{10}$



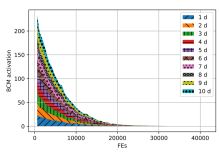
$dim = 10, f_1$



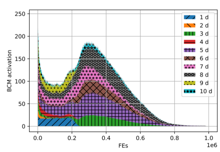
$dim = 10, f_2$



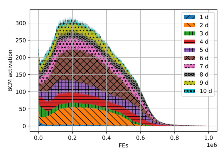
$dim = 10, f_3$



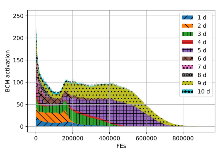
$dim = 10, f_4$



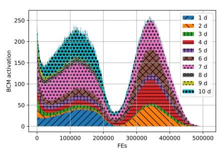
$dim = 10, f_5$



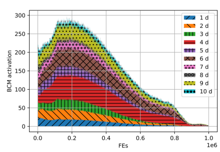
$dim = 10, f_6$



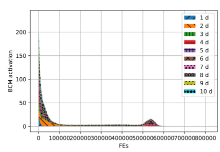
$dim = 10, f_7$



$dim = 10, f_8$

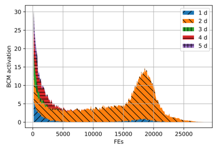


$dim = 10, f_9$

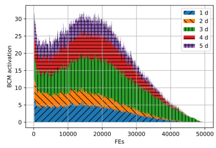


$dim = 10, f_{10}$

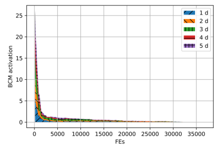
Algorithm IMODE, BCM Halving



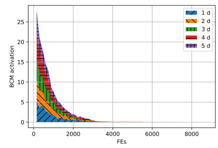
$dim = 5, f_1$



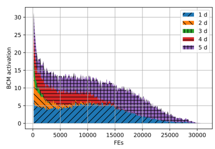
$dim = 5, f_2$



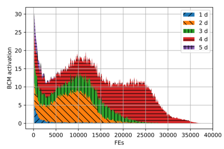
$dim = 5, f_3$



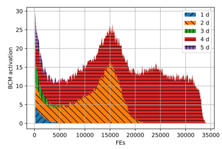
$dim = 5, f_4$



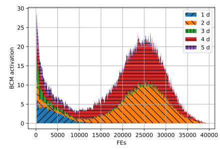
$dim = 5, f_5$



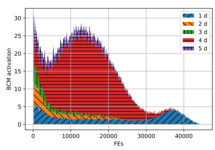
$dim = 5, f_6$



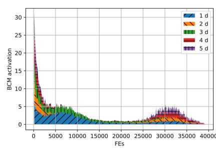
$dim = 5, f_7$



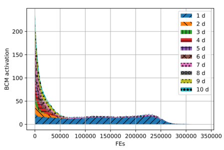
$dim = 5, f_8$



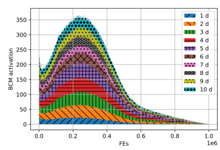
$dim = 5, f_9$



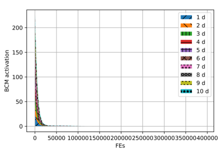
$dim = 5, f_{10}$



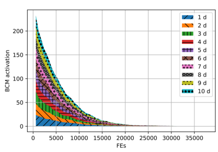
$dim = 10, f_1$



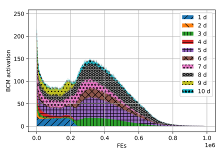
$dim = 10, f_2$



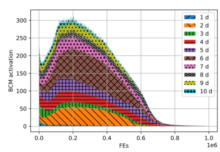
$dim = 10, f_3$



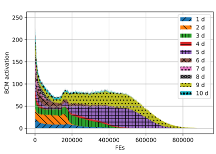
$dim = 10, f_4$



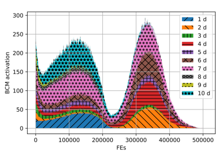
$dim = 10, f_5$



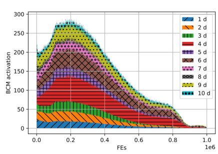
$dim = 10, f_6$



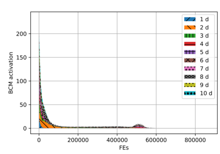
$dim = 10, f_7$



$dim = 10, f_8$

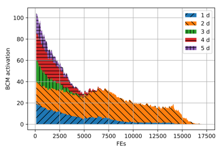


$dim = 10, f_9$

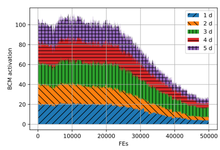


$dim = 10, f_{10}$

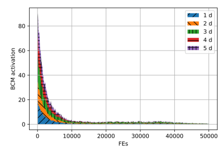
Algorithm AGSK, BCM Default



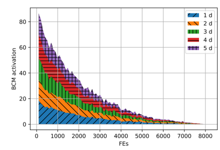
$dim = 5, f_1$



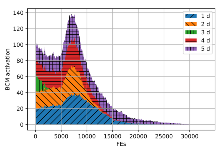
$dim = 5, f_2$



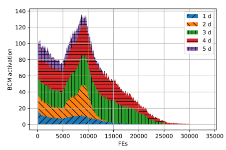
$dim = 5, f_3$



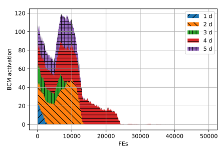
$dim = 5, f_4$



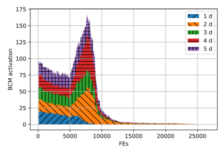
$dim = 5, f_5$



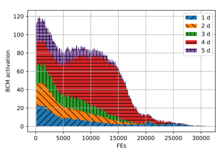
$dim = 5, f_6$



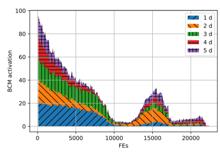
$dim = 5, f_7$



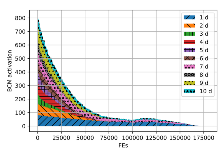
$dim = 5, f_8$



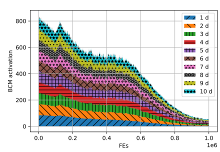
$dim = 5, f_9$



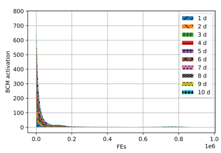
$dim = 5, f_{10}$



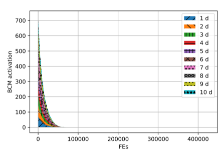
$dim = 10, f_1$



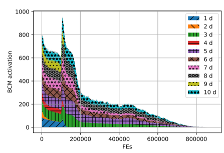
$dim = 10, f_2$



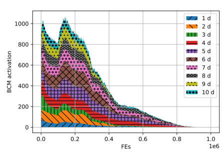
$dim = 10, f_3$



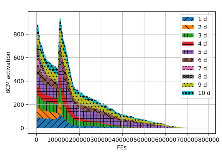
$dim = 10, f_4$



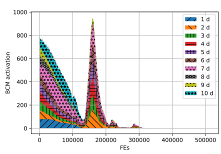
$dim = 10, f_5$



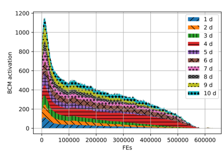
$dim = 10, f_6$



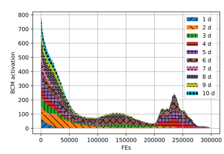
$dim = 10, f_7$



$dim = 10, f_8$

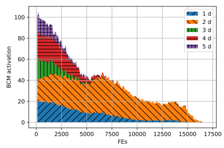


$dim = 10, f_9$

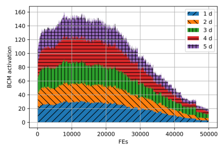


$dim = 10, f_{10}$

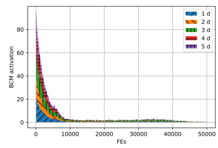
Algorithm AGSK, BCM Clipping



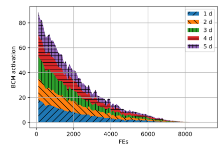
$dim = 5, f_1$



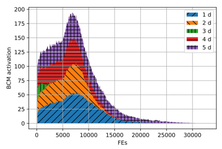
$dim = 5, f_2$



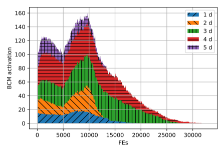
$dim = 5, f_3$



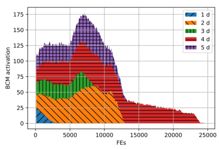
$dim = 5, f_4$



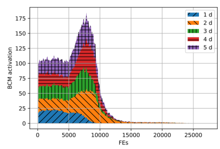
$dim = 5, f_5$



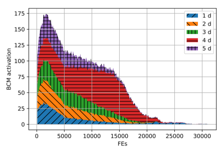
$dim = 5, f_6$



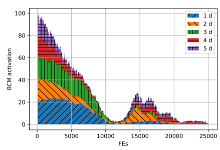
$dim = 5, f_7$



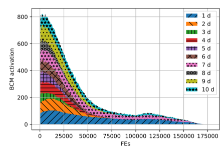
$dim = 5, f_8$



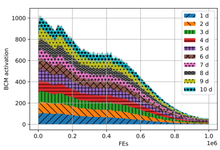
$dim = 5, f_9$



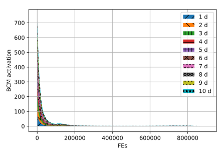
$dim = 5, f_{10}$



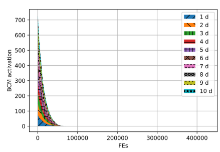
$dim = 10, f_1$



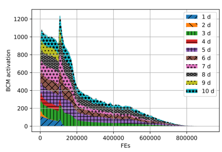
$dim = 10, f_2$



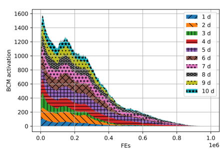
$dim = 10, f_3$



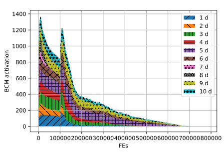
$dim = 10, f_4$



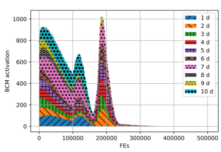
$dim = 10, f_5$



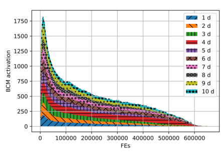
$dim = 10, f_6$



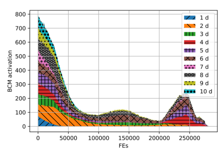
$dim = 10, f_7$



$dim = 10, f_8$

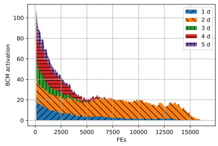


$dim = 10, f_9$

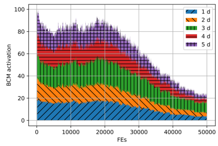


$dim = 10, f_{10}$

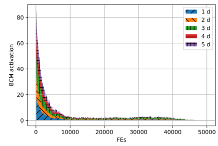
Algorithm AGSK, BCM Random



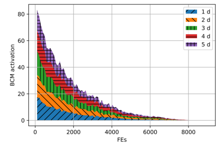
$dim = 5, f_1$



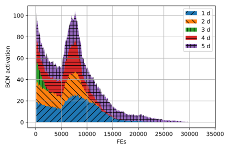
$dim = 5, f_2$



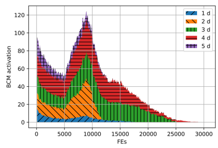
$dim = 5, f_3$



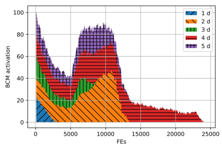
$dim = 5, f_4$



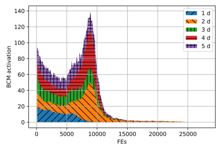
$dim = 5, f_5$



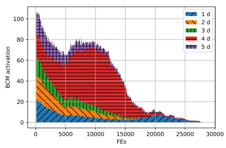
$dim = 5, f_6$



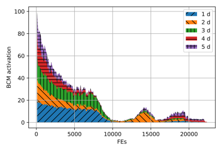
$dim = 5, f_7$



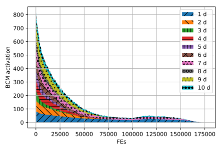
$dim = 5, f_8$



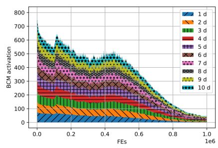
$dim = 5, f_9$



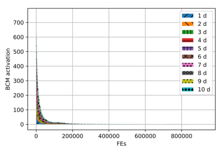
$dim = 5, f_{10}$



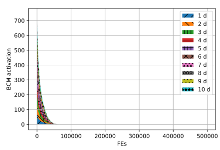
$dim = 10, f_1$



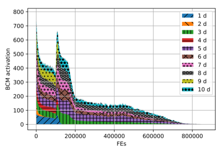
$dim = 10, f_2$



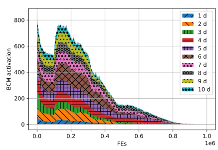
$dim = 10, f_3$



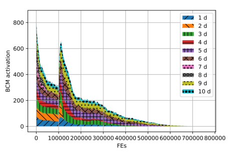
$dim = 10, f_4$



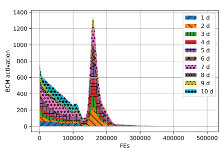
$dim = 10, f_5$



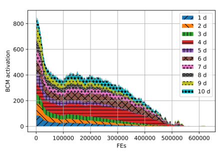
$dim = 10, f_6$



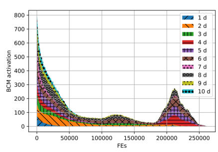
$dim = 10, f_7$



$dim = 10, f_8$

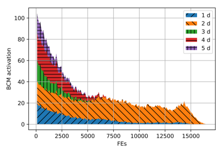


$dim = 10, f_9$

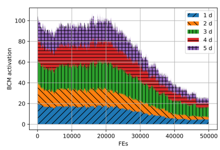


$dim = 10, f_{10}$

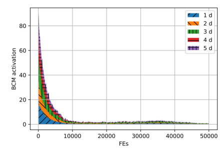
Algorithm AGSK, BCM Periodic



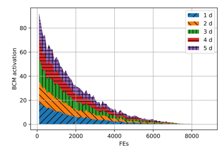
$dim = 5, f_1$



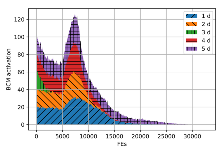
$dim = 5, f_2$



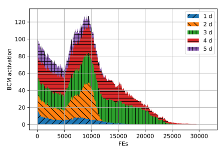
$dim = 5, f_3$



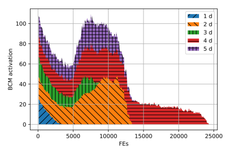
$dim = 5, f_4$



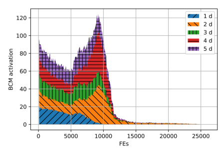
$dim = 5, f_5$



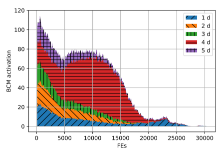
$dim = 5, f_6$



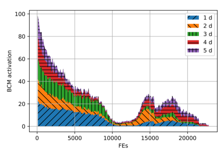
$dim = 5, f_7$



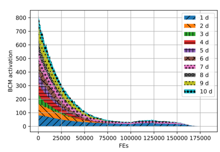
$dim = 5, f_8$



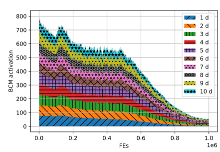
$dim = 5, f_9$



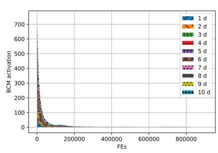
$dim = 5, f_{10}$



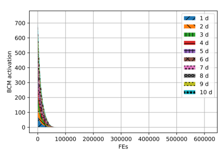
$dim = 10, f_1$



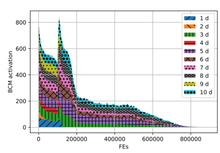
$dim = 10, f_2$



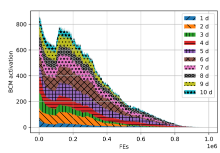
$dim = 10, f_3$



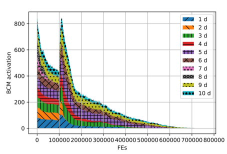
$dim = 10, f_4$



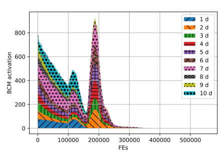
$dim = 10, f_5$



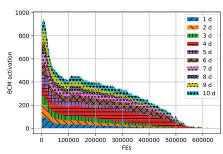
$dim = 10, f_6$



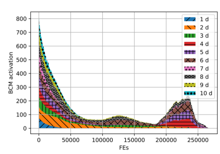
$dim = 10, f_7$



$dim = 10, f_8$

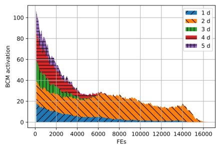


$dim = 10, f_9$

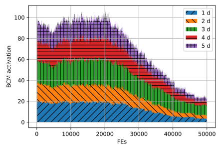


$dim = 10, f_{10}$

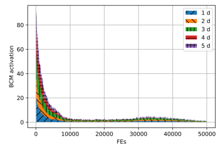
Algorithm AGSK, BCM Reflection



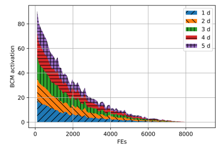
$dim = 5, f_1$



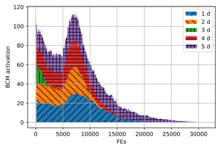
$dim = 5, f_2$



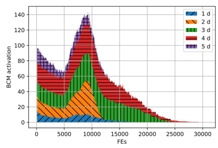
$dim = 5, f_3$



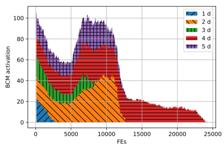
$dim = 5, f_4$



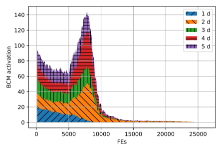
$dim = 5, f_5$



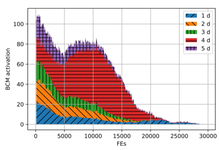
$dim = 5, f_6$



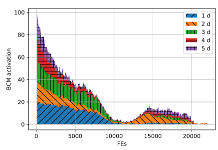
$dim = 5, f_7$



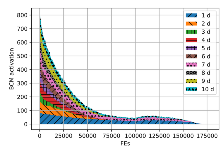
$dim = 5, f_8$



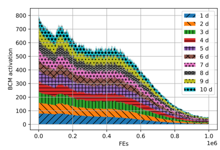
$dim = 5, f_9$



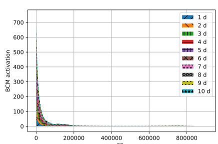
$dim = 5, f_{10}$



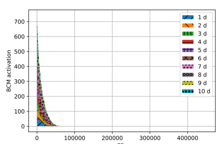
$dim = 10, f_1$



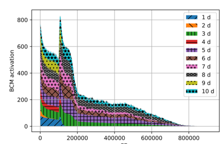
$dim = 10, f_2$



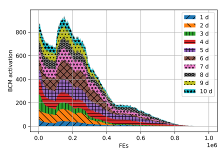
$dim = 10, f_3$



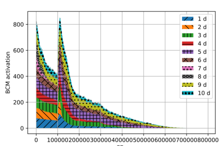
$dim = 10, f_4$



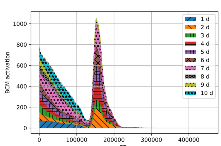
$dim = 10, f_5$



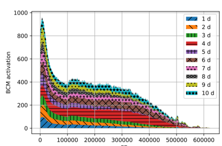
$dim = 10, f_6$



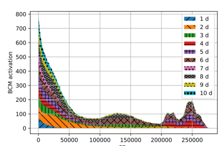
$dim = 10, f_7$



$dim = 10, f_8$

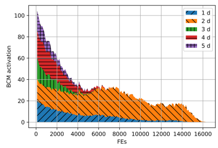


$dim = 10, f_9$

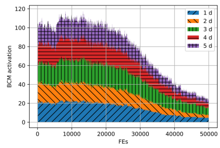


$dim = 10, f_{10}$

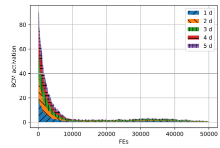
Algorithm AGSK, BCM Halving



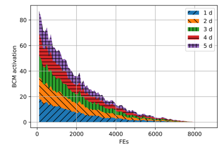
$dim = 5, f_1$



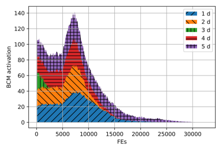
$dim = 5, f_2$



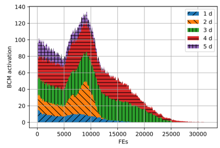
$dim = 5, f_3$



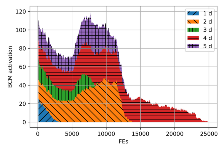
$dim = 5, f_4$



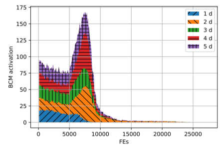
$dim = 5, f_5$



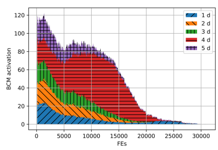
$dim = 5, f_6$



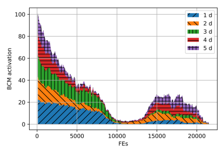
$dim = 5, f_7$



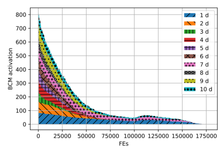
$dim = 5, f_8$



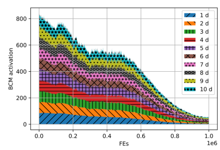
$dim = 5, f_9$



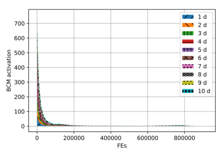
$dim = 5, f_{10}$



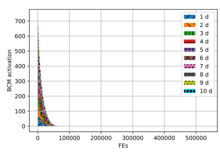
$dim = 10, f_1$



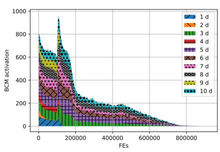
$dim = 10, f_2$



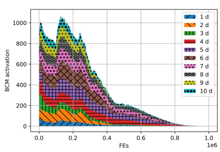
$dim = 10, f_3$



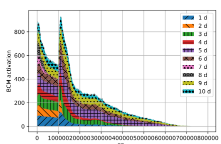
$dim = 10, f_4$



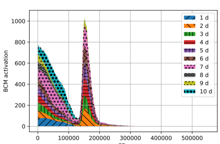
$dim = 10, f_5$



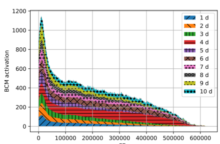
$dim = 10, f_6$



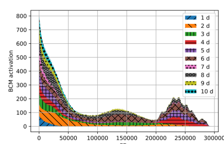
$dim = 10, f_7$



$dim = 10, f_8$



$dim = 10, f_9$



$dim = 10, f_{10}$