Development and Modification of Modern Bio-Inspired Swarm Algorithms

Ing. Anežka Kazíková, Ph.D.

Doctoral Thesis Summary

Tomas Bata University in Zlín

Tomas Bata University in Zlín
Faculty of Applied Informatics

Doctoral Thesis Summary

Vývoj a modifikace moderních bio-inspirovaných hejnových algoritmů

Development and Modification of Modern Bio-Inspired Swarm Algorithms

Author:	**Ing. Anežka Kazíková, Ph.D.**
Degree programme:	Engineering Informatics / P3902
Degree course:	Engineering Informatics / 3902V023
Supervisor:	Assoc. Prof. Ing. Roman Šenkeřík, Ph.D.
Advisor:	Ing. Michal Pluháček, Ph.D.
External examiners:	Prof. Dr. Peter Korošec
	Prof. Ing. Petr Dostál, CSc.
	Assoc. Prof. Ing. Pavel Krömer, Ph.D.

Zlín, October 2022

© Anežka Kazíková

Published by **Tomas Bata University in Zlín** in edition **Doctoral Thesis Summary.**
The publication was issued in the year 2022.

Klíčová slova: *Metaheuristická optimalizace, Hejnové algoritmy, Vývoj nových metaheuristik, Benchmarking, Bizoní algoritmus*

Key words: *Metaheuristic Optimization, Swarm Algorithms, Metaheuristic Development, Benchmarking, Bison Algorithm*

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

ISBN 978-80-7678-124-5

ABSTRAKT

Hejnové algoritmy se staly standardním nástrojem novodobé optimalizace. Příliv nových metaheuristik však přinesl kritiku vůči kvalitě, kvantitě a diskutabilní novosti těchto optimalizačních technik. Tato práce se zabývá momentálními trendy hejnových algoritmů v oblasti vývoje a modifikace, ale i nástrahami, které skýtají.

Už přes 30 let se metaheuristické algoritmy potýkají se stále stejnými problémy. Otázka stagnace, předčasné konvergence či nízké rozlišnosti řešení je výzvou, která je důležitá dnes stejně jako v počátcích oboru. To nemění ani vývoj nových algoritmů, protože ty mnohdy spíše odkrývají limity stávajících metodologických postupů v benchmarkingu, než aby přispívaly ke skutečnému posunu v optimalizaci. Nové metaheuristiky tak čelí předsudkům a všeobecné nedůvěře. Přestože otázka správných postupů je velmi aktuální, většina současných doporučení zůstává zpravidla v teoretické rovině bez praktické aplikace. To si tato práce klade za cíl začít měnit.

Autorka navrhuje sadu doporučení pro vývoj nových metaheuristik, které pak implementuje ve vlastním návrhu hejnového algoritmu s únikovým mechanismem z lokálního optima. Bizoní algoritmus představuje ukázku vývoje orientovaného na konkrétní optimalizační problém a zároveň funguje jako model vybraných aktuálních trendů a modifikací. Spojením teorie s praxí tato práce otevírá cestu k řešení nové generace výzev.

ABSTRACT

Swarm algorithms have become standard tools of modern optimization. However, the advent of new metaheuristics brought a wave of criticism against the quantity, quality, and novelty of these optimization techniques. This work describes the current trends in development and modification of swarm algorithms, as well as the challenges it includes.

For several decades metaheuristic algorithms have fought the very same optimization problems. The issues of stagnation, premature convergence, or low diversity of the solutions are dealt with today as well as in the beginning. The development of new algorithms does not state a change. Rather than genuinely advancing the field, new algorithms raise malpractice awareness in benchmarking. Due to the common low standard of their proposal studies, novel metaheuristics face a significant stigma of general distrust and disrespect. Although the good practice in benchmarking is a very recent topic, most current guidelines stay strictly in theory, i.e., are not applied. This work aims to start a change in this regard.

The Author proposes a set of recommendations for new metaheuristic development and implements them in a new swarm algorithm, which was developed with an escape mechanism out of the local optimum containment challenge. The Bison Algorithm showcases problem-oriented development and models current trends and modifications. The connection between theory and practice opens a way toward a new generation of challenges.

TABLE OF CONTENTS

1 INTRODUCTION 7

2 GOALS AND METHODS OF THE DISSERTATION 9

3 SWARM ALGORITHMS 10

3.1 Swarm Algorithms Challenges 11
3.2 Recommendations for Development of New Metaheuristics 14

4 BISON ALGORITHM 16

4.1 Motivation 16
4.2 Inspiration 16
4.3 Definition 17
4.4 Swarming Group 19
4.5 Running Group 20
4.6 Modifications of the Bison Algorithm 22

5 PERFORMANCE OF THE BISON ALGORITHM 23

5.1 Comparison with Other Metaheuristics 23
5.2 Local Optimum Containment Challenge 26

6 CONTRIBUTION TO SCIENCE AND PRACTICE 27

6.1 Does the World Need Yet Another Swarm Algorithm? 27
6.2 Applications of the Bison Algorithm 28
6.3 Dissertation Goal Fulfillment 30

7 CONCLUSION 30

8 CURRICULUM VITAE 32

REFERENCES 33

PUBLICATIONS OF THE AUTHOR 40

LIST OF FIGURES 42

LIST OF TABLES 43

LIST OF ABBREVIATIONS 43

1 INTRODUCTION

Artificial Intelligence has become an essential part of human lives today. Sometimes explicitly, sometimes undercover, AI guides us on the road, assists our phones, drives autonomous vehicles, controls our calendar, Google, Netflix, even Facebook have an entire department focused on advancement and more profound research on Artificial Intelligence. As a technology, it helps almost everywhere and it also proves to be a brilliant optimization tool.

The current trend in optimization is to find inspiration in nature. Simulations of various bio-inspired phenomena solve nontrivial optimization tasks. Complex optimization problems may find solutions by simulating diverse natural phenomena like foraging, hunting, courtship behavioral patterns, the Darwinian theory of evolution, and Mendelian genetic processes. The foundation lies in the multi-agent system. Each agent represents one particular solution to the solved problem, whose quality is determined by an objective function.

The bio-inspired optimizers are called metaheuristics. They include a wide range of algorithms like Differential Evolution [42], the Genetic Algorithm [14], Particle Swarm Optimization [27], the Cuckoo Search [46], the Grey Wolf Optimizer [33], the Self-Organizing Migrating Algorithm [48], the Passing Vehicle Search Algorithm [39], and many others.

Unlike classical mathematical optimization methods (e.g., linear, dynamic, or integer programming), metaheuristics cannot guarantee the actual discovery of optimal solutions. However, as compensation, they offer a viable solution in a reasonable time. Thus, the metaheuristic approach can be constructive, especially when exact mathematical methods struggle to solve a problem in a tolerable period of time.

But even metaheuristics sometimes sail against the wind. There are currently more than 300 algorithms [34], which often face criticism questioning their actual novelty, contribution, and their quality. Many troubles come from fallacious parameter settings, the inapplicability of the proposed algorithms, or embedding the same principles under a new alias [34, 41]. General disrespect for the population-based algorithms is not the only problem they face. Other struggles include standard optimization setbacks such as stagnation, premature convergence, low diversification of the population, or local optimum containment.

This work analyses the current trends in metaheuristic algorithms, including the optimization and development struggles. It investigates specific methods employable to tackle these problems and suggests recommendations for novel metaheuristic development. As proof of concept, adopting the suggested guidelines, the Author introduces a new metaheuristic swarm algorithm with a mechanism against local optimum containment.

The doctoral thesis summary is structured as follows:

- Section 2 designates the goals and methods of this dissertation.
- Section 3 introduces swarm algorithms, names a selection of the most popular ones, and analyses the numerous reservations against the development of novel swarm algorithms. As a reaction, it formulates a set of recommendations for new metaheuristic development.
- Section 4 proposes a new swarm algorithm based on bison herds' behavioral patterns while adopting previous section's principles and recommendations.
- Section 5 investigates the performance of the proposed algorithm.
- Section 6 evaluates the benefit of the work, considering both the applications and critical points of view.
- Finally, the conclusion contemplates the work and considers its meaning for science and practice.

2 GOALS AND METHODS OF THE DISSERTATION

1. **Map the current scene** of modern swarm algorithms, its trends, and challenges.
2. **Investigate** the methods addressing the weaknesses of swarm algorithms.
3. **Propose** a set of recommendations for new metaheuristics creation.
4. **Proof of concept testing**: Implement the proposed recommendations and methods in a new swarm algorithm.
5. **Evaluate the benefits** of the proposed algorithm for applied sciences.

Methods of fulfillment of goals of the dissertation include:

Critical analysis:
- Of novel metaheuristics creation process and its challenges.
- Of modifications, trends, and weaknesses of swarm algorithms.

Experiments:
- The proposed algorithm is compared to other state-of-the-art algorithms on various benchmarking testbeds IEEE CEC 2015, and 2017.
- The experiments focus on the investigation of the dynamics and inner processes of the proposed algorithm.

Evaluation:
- Evaluation of the algorithms comply with the evaluation criteria [1].
- The results are examined for statistical significance.
- Identification of the types of optimization problems suitable for the proposed techniques through an in-depth analysis of results.

Programming:
- Algorithms are coded in Python or MATLAB.
- Results are examined in Wolfram Mathematica.

3 SWARM ALGORITHMS

Metaheuristics are optimization algorithms that typically simulate bio-inspired phenomena. The inspiration comes from both natural (and supernatural) sources, including a wide range of scientific fields such as Physics [5], Chemistry [17], Biology [4], the Darwinian theory of evolution, and Mendelian principles of genetics [14, 42], or behavioral patterns of animal groups [6, 47].

Swarm algorithms are metaheuristics based on the collective intelligence phenomenon as a characteristic feature of animal swarms, flocks, or herds, as animal groups often make smart decisions only with local information and simple rules but no centralized control [47]. The fascination with the inspiration source and the success of some swarm optimizers ignited an unconventional interest. Hence, swarm algorithms nowadays represent approximately half of all the known classified metaheuristics with more than 150 specimens [34]. Table 3.1 represents the list of top 10 most popular swarm-based metaheuristics. The data were derived from the *Comprehensive Taxonomies of Nature- and Bio-Inspired Optimization* by Molina et al. [34]. The popularity was determined by number of references to the proposing publications on Scopus Database accessed 27/01/2021.

Tab. 3.1 Swarm-inspired metaheuristics sorted by the number of citations of the original proposal paper (accessed 27/01/2021).

	Swarm Algorithm	**Acronym**	**Year**	**Original Paper**	**Scopus Citations**
1	Particle Swarm Optimization	PSO	1995	[11]	11198
2	Ant Colony Optimization	ACO	1996	[10]	8199
3	Artificial Bee Colony	ABC	2005	[18]	4096
4	Cuckoo Search	CS	2009	[46]	3856
5	Grey Wolf Optimizer	GWO	2014	[33]	3842
6	Bat Inspired Algorithm	BAT	2010	[45]	2597
7	Whale Optimization Algorithm	WOA	2016	[32]	2222
8	Bacterial Foraging Optimization	BFOA	2002	[35]	2197
9	Firefly Algorithm	FA	2009	[44]	2143
10	Moth Flame Optimization Algorithm	MFO	2015	[31]	1105

3.1 Swarm Algorithms Challenges

Nowadays, swarm algorithms face two kinds of challenges: classical optimization issues and existential problems. The first one includes problems like stagnation, premature convergence, local optimum containment, or low population diversity. These struggles have been challenging metaheuristics since their beginning, and various methods try to tackle them, such as: multiple populations, population restart, randomization, self-adaptive parameters, and others.

The existential problems concern the criticism closely linked to novel metaheuristics development. The increasing emergence of swarm algorithms during the last few decades evoked a *"novel algorithms dilemma."* The introduction of metaphor-based development was followed by a massive wave of new swarm algorithms [9]. To clarify the trend, Figure 3.1 shows the proportion of swarm-based algorithms compared to the total number of new metaheuristics created in the years 1973–2018 based on data from [34].

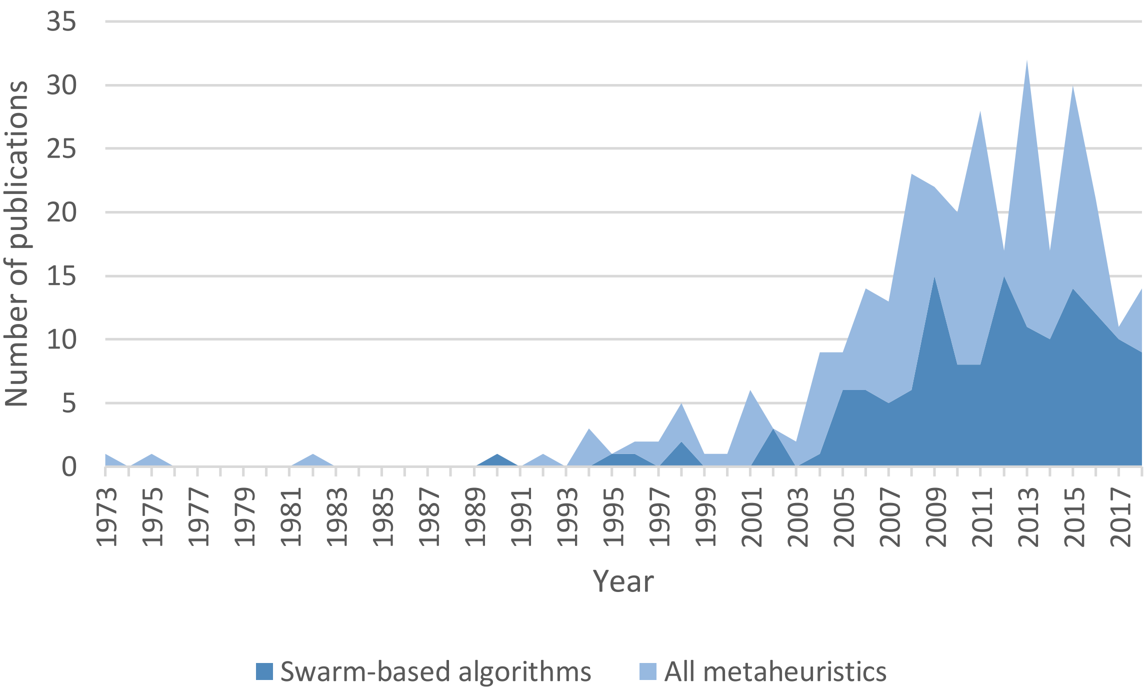


Fig. 3.1 The number of metaheuristic proposals in the years 1973-2018.

As a reaction to this "*metaheuristic avalanche*," an eye-catching project, the Evolutionary Computation Bestiary, started in 2018. The EC Bestiary[1] catalogs the metaphor-based metaheuristics. The primary purpose of this catalog is to highlight the number of metaheuristics and to point out (and make fun of[2]) the pitfalls of metaheuristic development. However, it also offers valuable sources that bring to light valid recurring mistakes linked to many metaheuristic proposals [38, 40, 41]. The authors point out these reservations:

- Bio-inspired lingo
- Duality of the algorithms
- Too simplified models of bio-inspiration
- Excessive focus on competition and novelty
- Experiments of poor quality
- Undefined relation between academic and real-world problems
- Lack of insight into the algorithms' functionality

There are three general responses to the aforementioned reservations: A) a complete ignorance of the critical points, B) rejection of novel metaheuristics altogether, or C) reflection of valid critical point in the future development.

Unfortunately, the first two approaches are adopted in most scenarios: the users of metaheuristics, completely ignoring the suggestions for good practice on one hand, versus the reviewers, who are getting fed up with metaheuristic malpractice and are inclined to reject the novel metaheuristic once they have read the title.

In 2005, Lee and Geem proposed the Harmony Search algorithm inspired by musicians' improvisation [29]. Five years later, the algorithm was proved to be a particular case of Evolutionary Strategies [43]. Its novelty and contribution were impeached. However, the effect was minimal. Figure 3.2 presents the number of publications citing the Harmony Search proposal versus the exposing publication. The ratio illustrates that the algorithm's popularity was not caused by, nor despite, the duality revelation. It was rather unnoticed.

[1] Available at `https://github.com/fcampelo/EC-Bestiary`, accessed 01/11/2021

[2] See, e.g., the Twitter account Daily Bio-heuristics of metaheuristic inspiration for every day (`https://twitter.com/BioHeuristics`) or the Ghost Detection Algorithm Parody (`http://oneweirdkerneltrick.com/spectral.pdf`), both accessed 01/11/2021

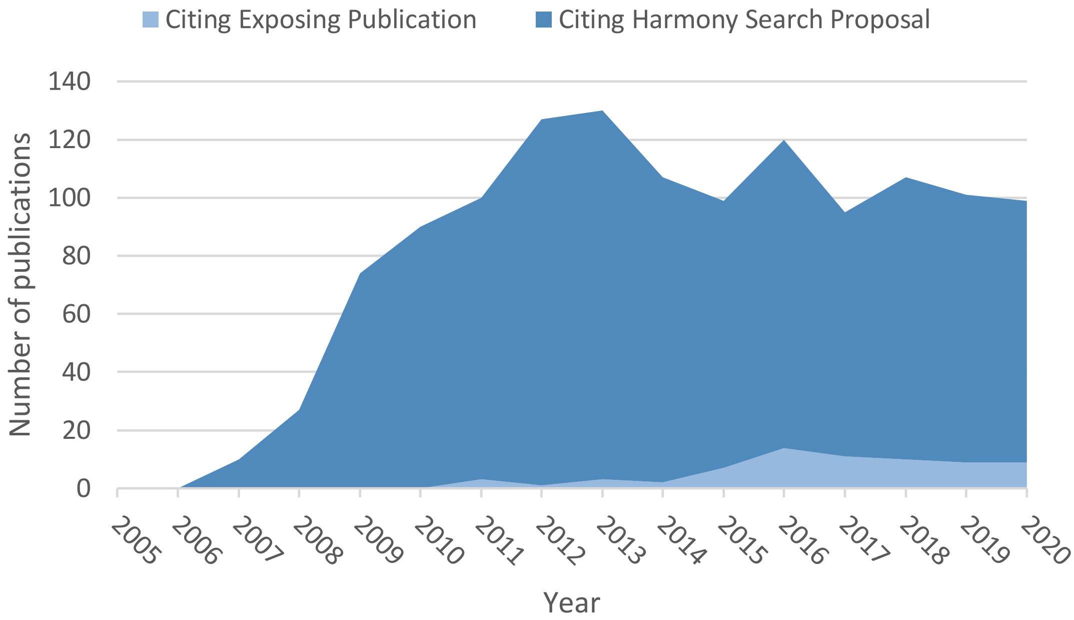


Fig. 3.2 Number of publications citing the Harmony Search Algorithm versus the exposure publication in 2005-2020.

Most recently, in 2020/2021, researchers called attention to current metaheuristic problems [13, 38, 41]. IEEE established a benchmarking taskforce[3] and networks[4]. Molina et al. (2020) fought the metaphor threat by proposing behavior-based classification of metaheuristics [34]. Guidelines for fair methodology in metaheuristic development and comparison [28] and for benchmarking [2] were proposed.

Still, when compared to the number of contrasting literature, the effort to improve the metaheuristic situation concerns just a drop in the ocean of academic publications. Critical publications mainly state the problem and mark the territory of metaheuristic badlands. The reservations point to the corrupt practice and wrongs in metaheuristic optimization but rarely offer solutions, and the applications are scarce.

[3] `https://cmte.ieee.org/cis-benchmarking/`, accessed 01/11/2021
[4] `https://sites.google.com/view/benchmarking-network/home`, accessed 01/11/2021

3.2 Recommendations for Development of New Metaheuristics

The criticism mentioned can be formulated on a positive note into guidelines for metaheuristic design. The Author presents a set of rules based on recommendations from the following sources: [2, 16, 28, 34, 38, 40].

Guidelines for Algorithm Design

- Name motivation (not metaphor-based)
- Use standard vocabulary
- Share the source code of novel algorithms
- Describe algorithms with flowcharts for a better understanding
- Analyze components of the proposed algorithm individually
- Keep it simple

Guidelines for Selection of Algorithms to be Compared and Benchmark

- Select algorithms to be compared with respect to the experiment's goal

For performance-oriented comparison, compare algorithms with:

- Original version of the algorithm (first proposal)
- Reference version of the algorithm (the one that is modified)
- Best algorithms on the benchmark being examined (competition winner)
- Other algorithms operating on a similar principle

Select benchmark problems:

- Of broad characteristics without bias
- Prefer standard benchmark test sets

Guidelines for Experimental Setup

- Prefer own implementation over literature-based results
- Provide the same conditions for all the experiments
- Share the source codes of all the algorithms

- Tune the parameters of all the algorithms for the problem at hand with statistical tests
- Combine multiple performance measures

When examining the CPU execution time, all algorithms should:

- Be coded by the same programmer
- Be coded in the same programming language
- Share most functions
- Be examined on the same computer

Guidelines for Results' Analysis

- Use statistical tests for significance
- Allow negative results
- Show results in context, provide interpretation
- Be cautious with generalization
- Depict the results in both graphs and tables
- Advocate assets and contribution of the algorithm (novelty/superior performance/methodology/challenge particular problem)

The need for justification of new algorithm development was expressed in multiple publications [8, 28, 38, 41]. According to LaTorre [28], the arguments in favor of the usefulness of novel methodology are: undeniable novelty, results surpassing state-of-the-art optimizers, and contribution to methodology.

In this regard, the Author would like to add another motivation for the justification of novel metaheuristic development: aiming the development of novel algorithms at the known problems of current metaheuristic practice. Tackling the fundamental puzzles of metaheuristic optimization such as stagnation or premature convergence may lead to the evolution of metaheuristics.

4 BISON ALGORITHM

4.1 Motivation

Manifold publications [2, 7, 15, 38, 40, 41] point to the common substandard practice of metaheuristic proposals and result in general conclusions. The actual applications of these studies are, however, scarce (see, e.g., [28]). The proposed algorithm was designed to follow the presented guidelines, demonstrating the significance of the recommendations from Section 3.2.

The second motivation was to advocate an additional justification argument for metaheuristic development. So far, three arguments have justified the usefulness of novel algorithm proposals [28], namely:

- Superior performance surpassing state-of-the-art optimizers
- Undeniable novelty
- Contribution to methodology, e.g., improving a common technique

In addition, the Author would like to add a fourth argument: **Aiming development at tackling the current optimization problem**. To support this argument, an algorithm aimed at fighting the local optimum containment problem was developed. It embeds a unique mechanism to avoid local optimum and ensures the same rate of exploration throughout the whole optimization process.

4.2 Inspiration

The Bison Algorithm simulates the typical behavior of bison herds: swarming and running. When predators attack bison, they form a circle with strong cattle at the outer edge of the circle. The weaker ones hide inside the circle in a safer position. A running bison can reach a velocity of 56 km per hour and keep it up for as much as thirty minutes [3, 37]. These behavior patterns serve as model exploitation and exploration techniques.

4.3 Definition

The main characteristic of the Bison Algorithm lies in the division of the population into two groups. The first group, called the swarming group, takes care of exploitation, approaching closer to the center of several fittest solutions. In contrast, the second group steadily examines the search space for new, promising solutions.

The algorithm is outlined in Algorithm 1 and a simplified flowchart in Figure 4.1. The UTB A.I. Lab Github repository[5] hosts the source code of the Bison Algorithm, which is free to use.

Algorithm 1 Pseudocode of the Bison Algorithm.

1. Initialization:
    Objective function: $f(\boldsymbol{x}),\ \boldsymbol{x} = (x_1,\ x_2,\ ...,\ x_D)$
    Generate swarming group ***SG*** randomly
    Generate running group ***RG*** around $\boldsymbol{x}_{best}$, $f(\boldsymbol{x}_{best}) \leq f(\boldsymbol{x}), \forall \boldsymbol{x} \in \boldsymbol{SG}$
    Select elite bison group ***EG*** based on obj. function value
    Sort the population and redefine ***SG*** based on obj. function value $\boldsymbol{SG}_j = (sort(\boldsymbol{SG} \cup \boldsymbol{EG}))_j, j = 1, ..., |\boldsymbol{SG}|$
    Generate the ***run direction*** vector (Eq. 4.4)
2. **For** every iteration i **do**
3.   Compute the ***center*** of ***EG*** (Eqs. 4.1, 4.2)
4.   **For** every bison ***x*** in ***SG*** **do**
5.   Compute a new candidate solution $\boldsymbol{x}_{new}$ (Eq. 4.3)
6.   **If** $f(\boldsymbol{x}_{new}) < f(\boldsymbol{x})$ **then** $\boldsymbol{x} = \boldsymbol{x}_{new}$
7.   **End for**
8.   Adjust ***run direction*** vector (Eq. 4.5)
9.   **For** every bison ***x*** in ***RG*** **do**
10. $\boldsymbol{x} = \boldsymbol{x} +$ ***run direction*** (Eq. 4.6)
11.   **End for**
12.   Redefine ***SG*** for the next iteration $i+1$:
    $\boldsymbol{SG}_{i+1,j} = (sort(\boldsymbol{SG}_i \cup \boldsymbol{RG}_i))_j, j = 1, ..., |\boldsymbol{SG}|$
13. **End for**

[5] `https://github.com/TBU-AILab/Bison-Algorithm-OOP`, accessed 02/06/2022

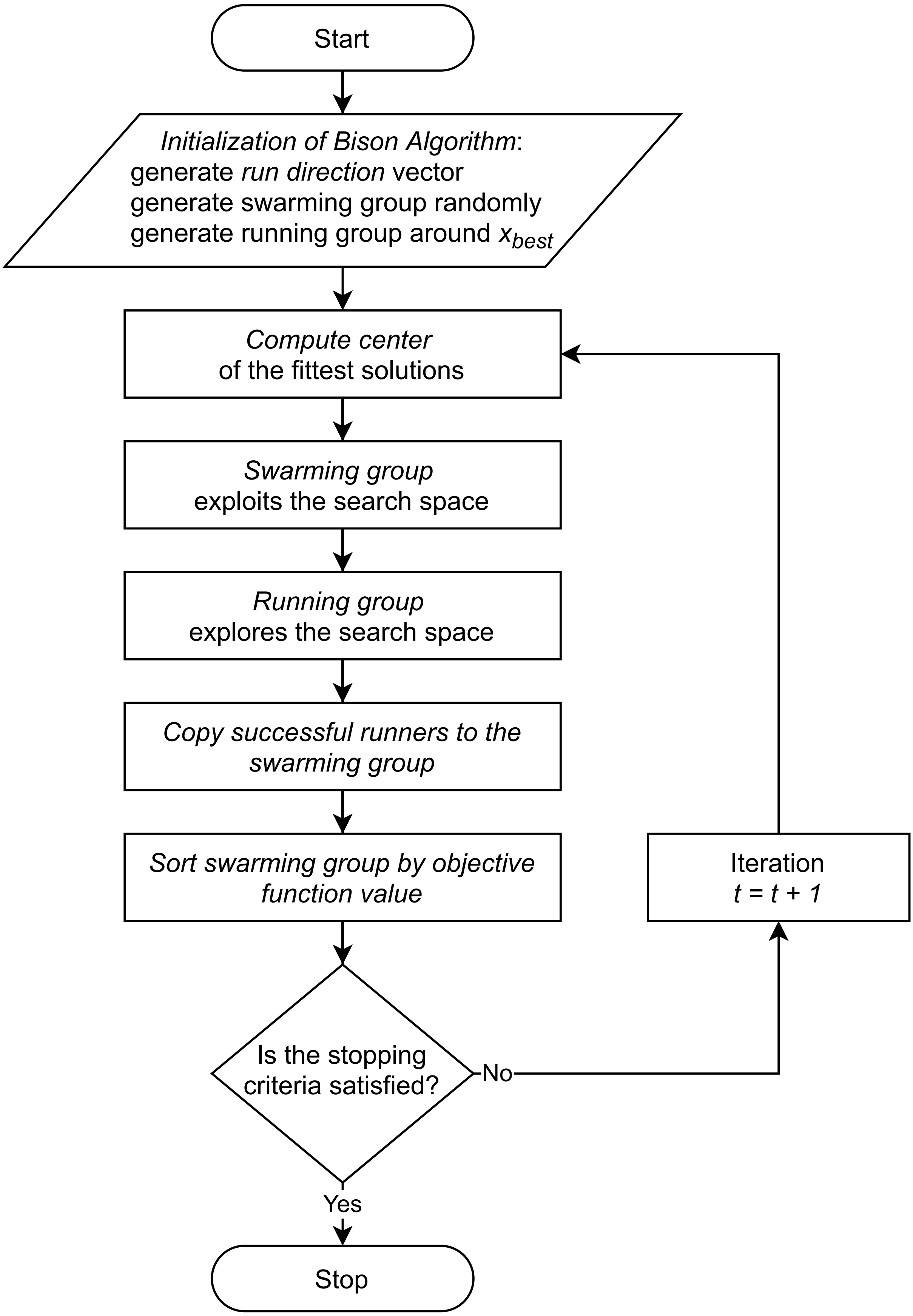


Fig. 4.1 Simplified flowchart of the Bison Algorithm.

The control parameters of the Bison Algorithm include:

- **Population** NP defines the number of individuals in the population,
- **Elite group size** EG determines the number of fittest solutions for center computation,
- **Swarm group size** SG represents the number of solutions in the swarming group,
- **Overstep parameter** defines the maximum length of the swarming movement to the center (0 - no movement, 1 - max. movement to the center)

The recommended configuration for IEEE CEC 2017 benchmark is: population of 50, elite group size of 20, swarm group size of 40 and overstep of 3.5 [19].

4.4 Swarming Group

The swarming group shifts its members in the direction of the center of several fittest solutions. Preliminary experiments promoted ranked center computation (Eqs. 4.1, 4.2), considering the order of the best solutions during the calculation. Every solution in the swarming group computes a new solution candidate that replaces the current swarmer if it improves its quality (Eq. 4.3).

The flowchart Figure 4.2 represents the principle of the swarming movement. Figure 4.3 shows the cumulative locations of the solutions in 50 iterations during the actual run on the 2-dimensional Rastrigin's Function.

$$\boldsymbol{weight} = (10, 20, 30, ..., 10 \cdot EG) \tag{4.1}$$

$$\boldsymbol{ranked\ center} = \sum_{i=1}^{EG} \frac{weight_i \cdot \boldsymbol{x}_i}{\sum_{j=1}^{EG} weight_j} \tag{4.2}$$

$$\boldsymbol{x}_{new} = \boldsymbol{x} + (\boldsymbol{ranked\ center} - \boldsymbol{x}) \cdot \boldsymbol{random}(0, overstep) \tag{4.3}$$

Where:

– EG is the elite group size parameter,
– i, j are indexes for center computation $i, j = 1, ..., |EG|$,
– $\boldsymbol{x}$ and $\boldsymbol{x}_{new}$ represent a swarming group and a new candidate solutions,

– ***random*** yields a random vector of within the arguments,
– and *overstep* defines the maximum length of the swarming movement.

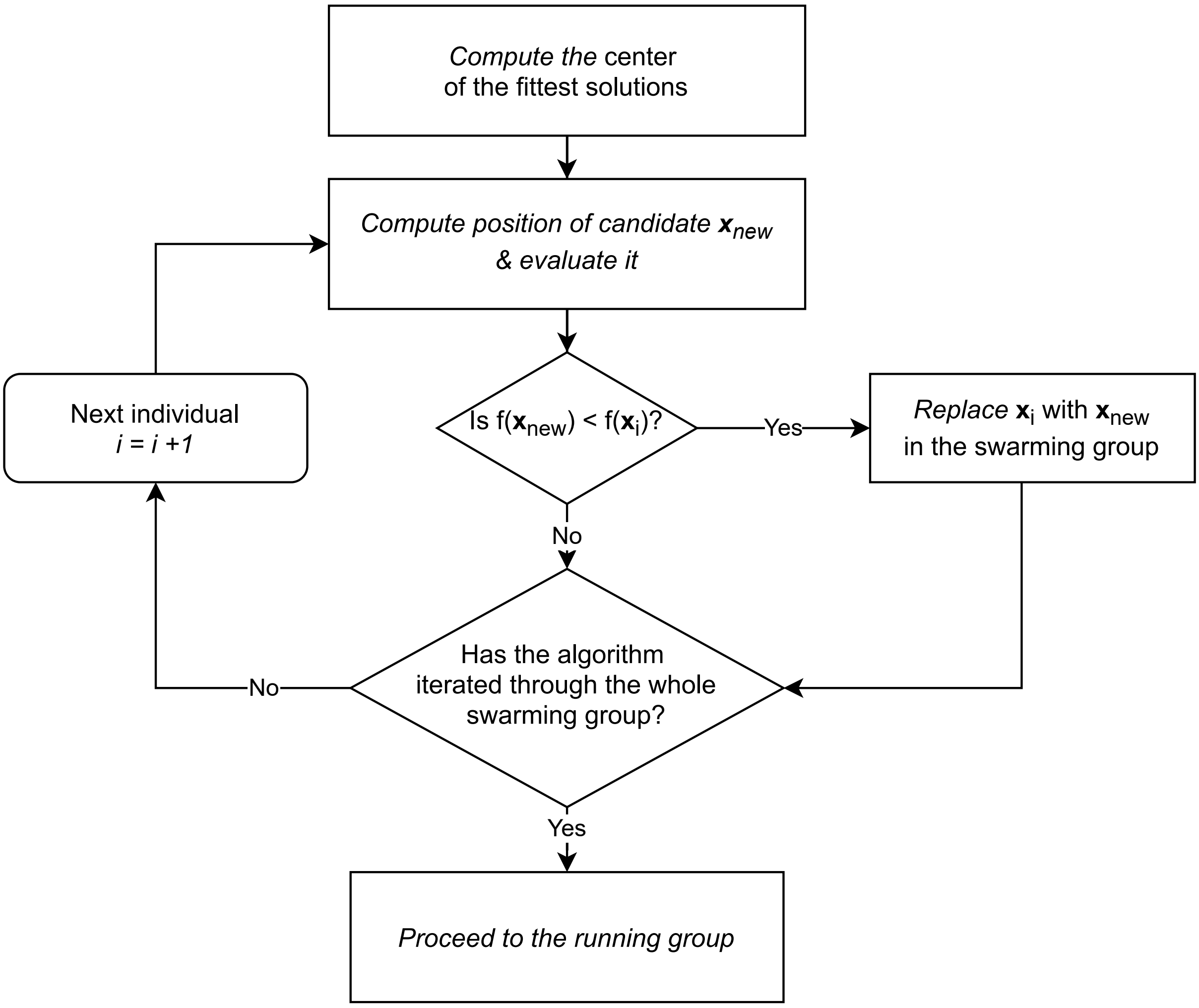


Fig. 4.2 Flowchart of swarming movement.

4.5 Running Group

The running group systematically shifts the solutions in the run direction vector. The algorithm generates the run direction vector randomly during the initialization (Eq. 4.4), and then it slightly changes in every iteration (Eq. 4.5). The running movement (Eq. 4.6) happens even if it lowers the quality of the solutions.

$$run\ direction_D = random(\frac{ub - lb}{45}, \frac{ub - lb}{15}) \tag{4.4}$$

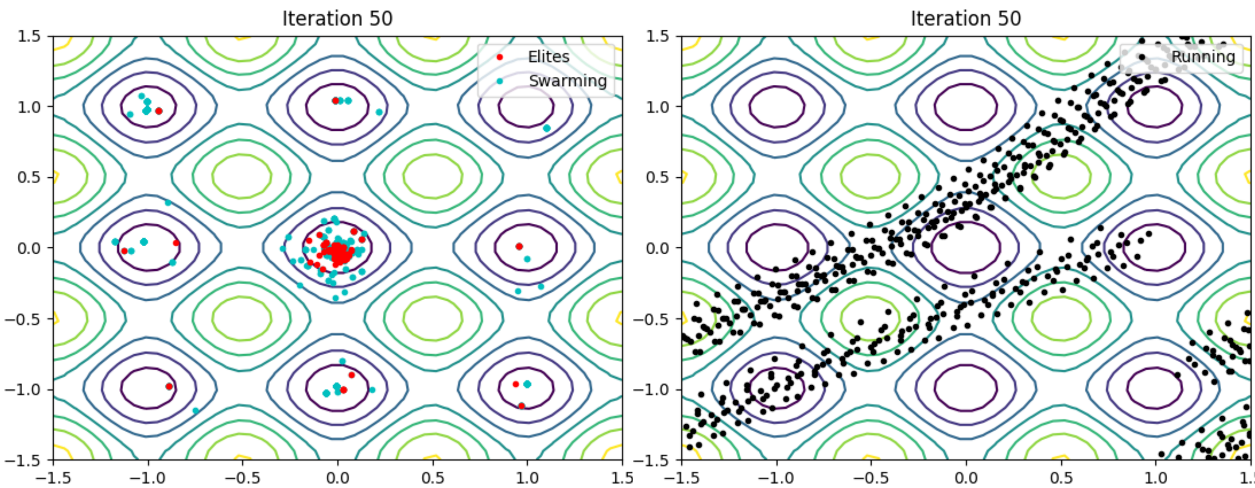
$$\boldsymbol{run\ direction}_{i+1} = \boldsymbol{run\ direction}_i \cdot random(0.9, 1.1) \tag{4.5}$$

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \boldsymbol{run\ direction}_{i+1} \tag{4.6}$$

Where:

- $\boldsymbol{run\ direction}$ is the run direction vector,
- D is a dimension,
- i is current iteration,
- ub and lb are the upper and lower boundaries of the search space,
- and $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i+1}$ represent a running group member and its updated position.

This exploration implementation crosses the borders quite often. The study *Border Strategies of the Bison Algorithm* [22] examined the most feasible border strategy and recommended using the hypersphere strategy, connecting the dimensional upper and lower boundaries.



(a) Swarming group (b) Running group

Fig. 4.3 Cumulative movement of the swarming group (a) and the running group (b) on the 2-dimensional Rastrigin's Function.

4.6 Modifications of the Bison Algorithm

The Bison Algorithm was developed with an escape mechanism from local containment. Simulating these behavior patterns in two groups based on their objective function value separately founded the first version of the algorithm [23].

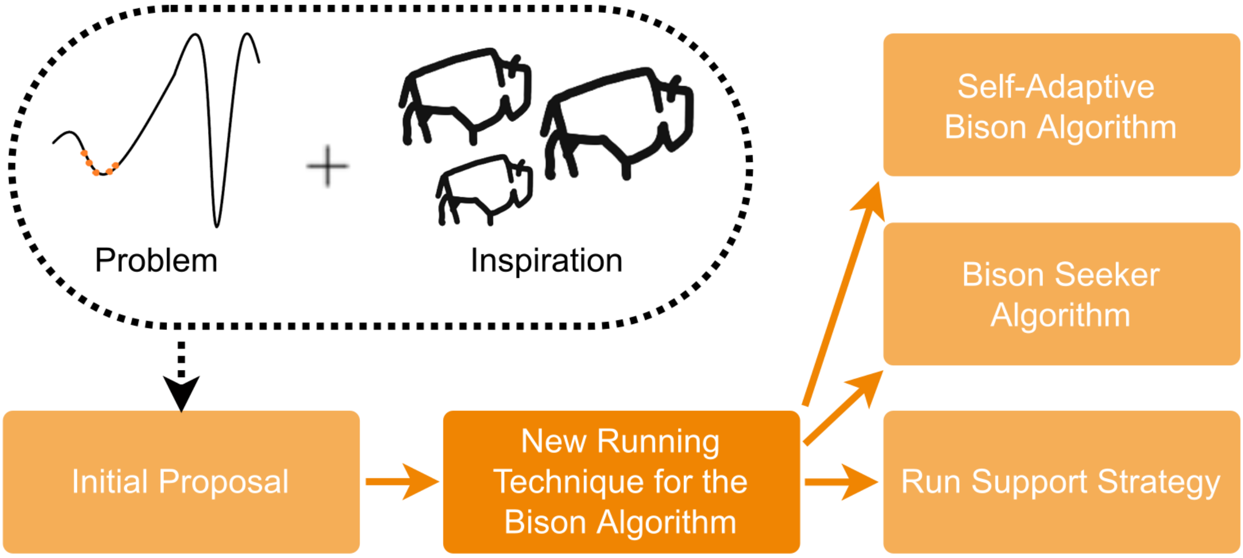


Fig. 4.4 Pipeline of the Bison Algorithm development process.

Further variations investigated the options for how to advance the explorative capacity of the algorithm. The first modification tested the benefits of coherent exploration, which was ultimately employed in all the modifications that followed [21].

The Bison Seeker modification investigated the behavior change of the explorative solutions when discovering an attractive solution [20]. *The Run Support Strategy* added a new parameter to support the utilization of the discovered solutions [25]. Finally, in 2021 was developed a *Self-Adaptive Variant of the Bison Algorithm* based on multi-population scheme with various parameters configuration, which is available at TBU's A.I.Lab GitHub repository[6]. The development process is mapped in Figure 4.4.

[6] `https://github.com/TBU-AILab/Bison-Algorithm-OOP`, accessed 02/06/2022

5 PERFORMANCE OF THE BISON ALGORITHM

5.1 Comparison with Other Metaheuristics

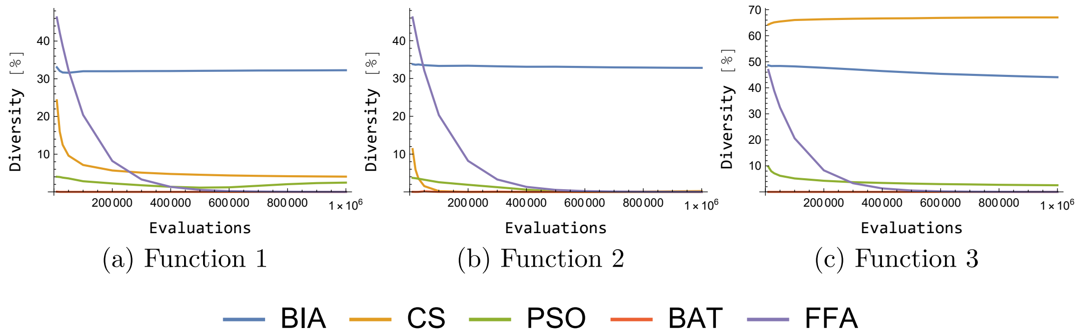
The Bison Algorithm was compared to other swarm algorithms on the benchmark sets of IEEE CEC 2015 and 2017 in 10, 30, 50, and 100 dimensions. Following the evaluation recommendations [1], each experiment took 51 independent runs, consisting of $10,000 \times D$ evaluations of the objective function.

The selected algorithms were: the Bison Algorithm (BIA), the Cuckoo Search (CS), Particle Swarm Optimization (PSO), the Bat Algorithm (BAT), and the Firefly Algorithm (FFA). They were implemented from the EvoloPy optimization library [12] and are available at TBU's AILab GitHub repository[7]. The control parameters were tuned for each algorithm separately in [26].

To explore the inner dynamics of the optimizers diversity of the algorithms was examined using the diversity computation metric from [36]. Figure 5.1 shows the mean diversity on the first three problems of the CEC 2015 test set, however the Bison Algorithm maintained the same diversity rate on the rest of the tested problems as well. Figure 5.2 shows all the error value convergences on IEEE CEC'17 F4 problem. The sudden drops indicate an escape from local optima, when the running group found a promising solution. When compared to other optimizers, these sudden drops happened more frequently.

The most common method of performance analysis is to investigate the error values of the final solutions. Table 5.1 and Table 5.2 summarize the Wilcoxon Rank-Sum test results ($p<0.05$). The tests count the number of problems where one algorithm performed significantly better than all the remaining algorithms. The problems were compared with the Friedman Rank test ($p<0.05$) in Figure 5.3. This test ranks the algorithms and sets the significance threshold as the Nemenyi critical distance; the algorithms over the distance performed significantly worse than the first-ranked algorithm.

[7] `https://github.com/TBU-AILab/Bison-Algorithm-OOP`, accessed 02/06/2022



(a) Function 1 (b) Function 2 (c) Function 3

Fig. 5.1 Mean diversity convergences example (IEEE CEC 2015 in 100 dimensions.

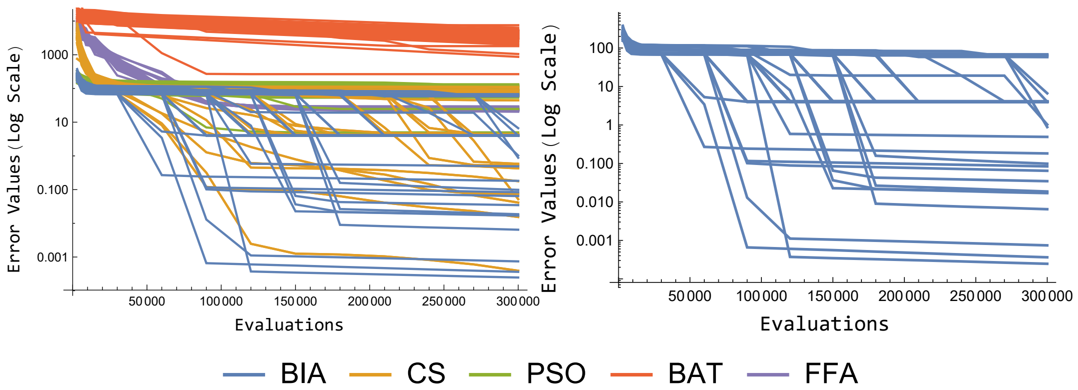


Fig. 5.2 Convergence of all runs of the swarm algorithms compared on IEEE CEC 2017 Function F4 in 30 dimensions.

Tab. 5.1 Winning Algorithms on CEC 2015 (Wilcoxon Rank-Sum test, p<0.05).

	None	**BIA**	**CS**	**PSO**	**BAT**	**FFA**
10 dimensions	6	3	**5**	0	1	0
30 dimensions	4	**5**	3	0	1	2
50 dimensions	5	**4**	3	0	1	2
100 dimensions	4	**5**	0	0	1	5
Sum of wins	19	**17**	11	0	4	9

Tab. 5.2 Winning Algorithms on CEC 2017 (Wilcoxon Rank-Sum test, p<0.05).

	None	**BIA**	**CS**	**PSO**	**BAT**	**FFA**
10 dimensions	7	7	**13**	1	0	2
30 dimensions	3	**14**	6	1	0	6
50 dimensions	7	**10**	5	1	0	7
100 dimensions	6	**11**	2	1	0	10
Sum of wins	23	**42**	26	4	0	25

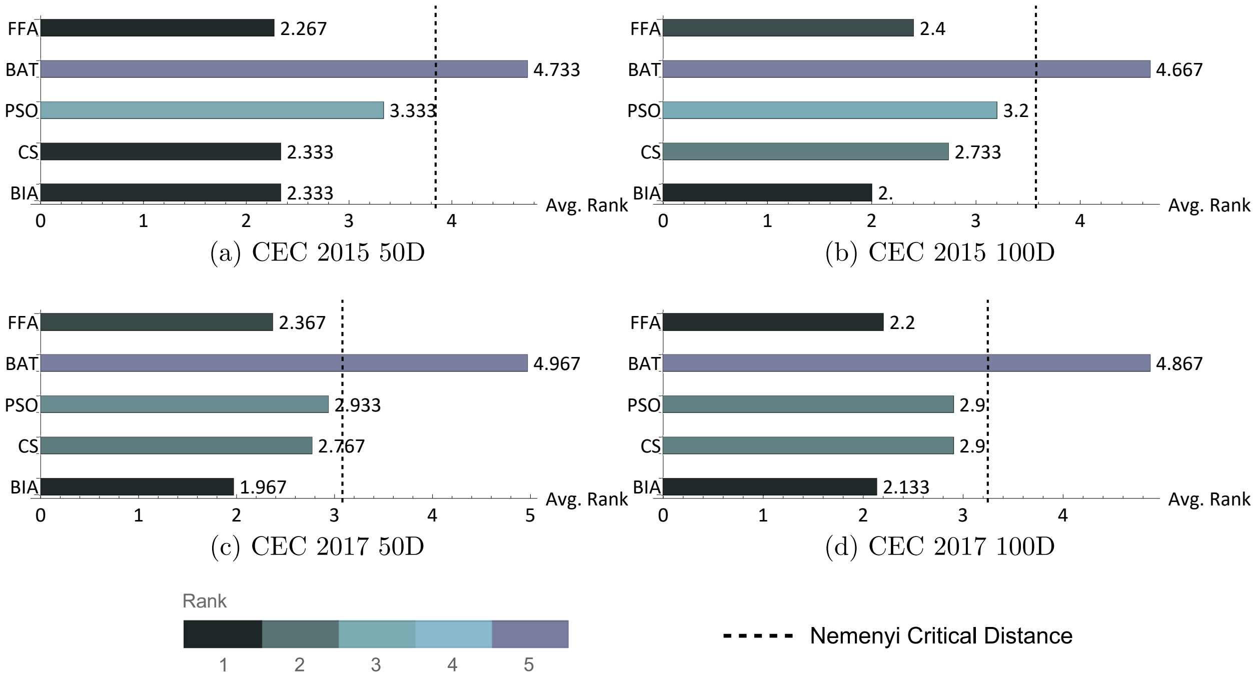


Fig. 5.3 Rank comparison of the BIA, CS, PSO, BAT, and FFA on benchmarks CEC 2015 and CEC 2017 (Friedman Rank Test, p<0.05).

The Bison Algorithm was generally successful when statistically compared to other swarm optimizers in most tested scenarios. However, despite the success against swarm optimizers selection, the proposed algorithm was also compared to the competition winners of the examined benchmark test sets - SPS L SHADE[8] and EBO with CMAR[9], and did not outperform them.

Yet the ultimate goal of the experiment was not to prove the Bison Algorithm's superiority to the state-of-the-art optimizers. The main focus of the development was to create a tool to fight the local optimum containment. Hence, the next Section analyses, how the algorithm deals with the local optimum containment challenge in practice.

[8] `https://github.com/P-N-Suganthan/CEC2015-Learning-Based` accessed 13/06/2022
[9] `https://github.com/P-N-Suganthan/CEC2017-BoundContrained`, accessed 13/06/2022

5.2 Local Optimum Containment Challenge

The algorithms were compared with respect to the characteristics of the solved problems. Based on the benchmark definitions [1], the problems from CEC 2017 benchmark test set were classified into selected classes. The results were then examined with the Wilcoxon Rank-Sum test (Figure 5.4) on the corresponding test sets across all dimensions.

The Bison Algorithm was especially successful when solving multimodal problems, problems with many local optima and problems with second best optimum far from the global best. Interestingly, all of these attributes point to a higher risk of the local optimum containment challenge, which the Bison Algorithm surpassed.

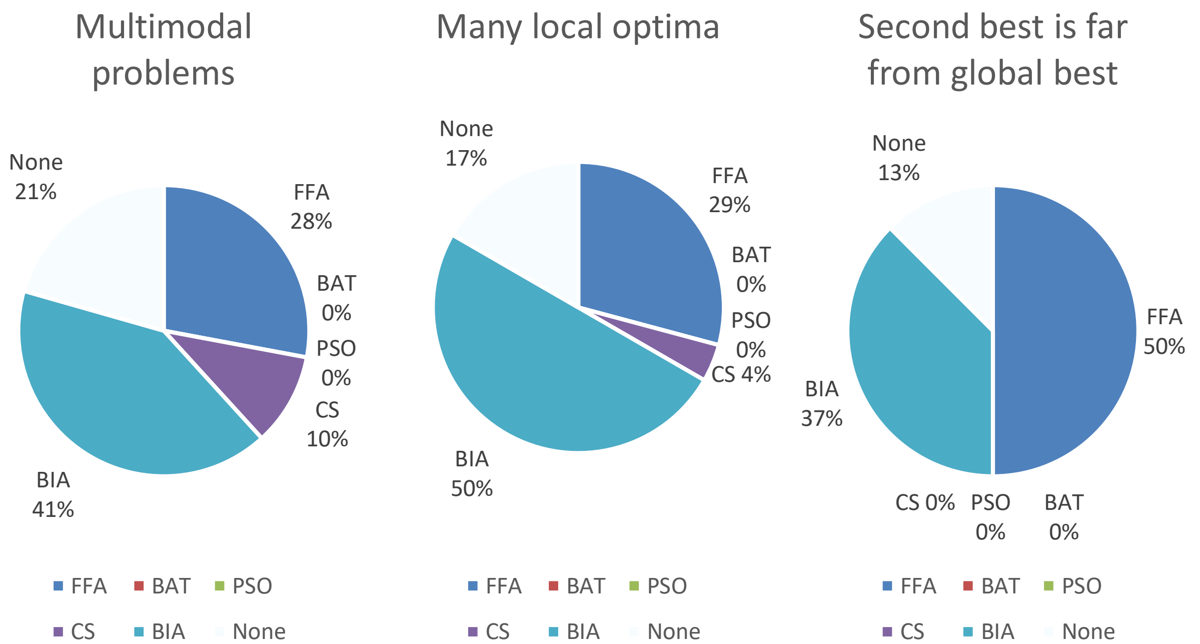


Fig. 5.4 Performance measure comparing the number of problems with a significantly better solution on the set with characteristic feature (Wilcoxon p<0.05).

6 CONTRIBUTION TO SCIENCE AND PRACTICE

6.1 Does the World Need Yet Another Swarm Algorithm?

The excess of metaphor-based algorithms has been clearly stated as a major risk factor of current metaheuristics. More than 320 algorithms escalated into various troubles like the duality of algorithms, scepticism of any *"novel"* technique, or open disdain for metaheuristics [34]. How could creating a new algorithm help?

It is important to acknowledge that even reviewers' scowls cannot ultimately end the development of new algorithms; very likely, new metaheuristics would arise despite generic opposition. Nevertheless, to prevent substandard production, it is vital to find a way of avoiding the recurrent mistakes that are often connected to new metaheuristic proposals. That is the goal of this work.

For meaningful development, the Author proposed rules for future design of novel swarm and metaheuristic algorithms and created the Bison Algorithm a proof of concept of these guidelines. Following the recommendations, the development was aimed at a common yet poignant problem of local optimum containment. As a result, an algorithm was born with interesting results for solving problems with many local optima.

One without the other would not make any difference. Suggesting a set of recommendations without their application would be just another invitation to improve metaheuristic practice with minimal impact. Developing a new algorithm without good scientific practice would, in fact, only reinforce the inappropriate "substandard" label on novel metaheuristics. But together, these two concepts may benefit new metaheuristics that are yet to be created. Ultimately, to answer the title question, reflecting the reservations and trying raise the standard of metaheuristic proposals is what the world certainly needs.

6.2 Applications of the Bison Algorithm

The most valuable contribution of swarm algorithms lies in quick solutions to complex real-life problems, including transportation, energy, logistics, or social networks [9]. Real-life problems need efficient real-time solutions. Although the metaheuristic approach does not guarantee finding the exact optimum, "good" solutions are often sufficient.

The Bison Algorithm was successfully used to optimize 3 PID controllers – the water tank test, mass spring damper, DC motor, and their cascade versions in [24]. The problems were defined to maintain the desired water level $\dot{h}$ in the tank by changing the water inflow, to maintain the desired position $s*$ of mass m_1 by managing the control force F, and to maintain the desired motor speed $\omega*$ by managing input voltage F.

The experiment compared five optimizers: the Genetic Algorithm (GA), Differential Evolution (DE), Particle Swarm Optimization, the Cuckoo Search, and the Bison Algorithm, and examined the differences between standard versus cascade PID controllers. Figure 6.1 summarizes the time consumption of each algorithm, and the best and worst result statistics of the optimizers: on how many of the six examined problem scenarios one algorithm deliver superior or inferior results to all the others. The paper concluded that the Bison Algorithm delivered top results (in a 5% range from the best-found results) in the majority of the tested problems.

The Bison Seeker Algorithm was applied as a hybrid method of symbolic regression in [30]. The algorithm outclassed basic symbolic regression even with a non-standard parameter setting of very few iterations and small populations. Figure 6.2 shows the percentual success rate of various instances of hybrid Bison Seeker Algorithm Symbolic Regression compared to standard Symbolic Regression.

Fig. 6.1 Metaheuristic comparison: a) average time consumption of each algorithm during the experiment, b) how often one algorithm delivered a final solution of quality in the top 5%, c) how many times the algorithm delivered the worst results.

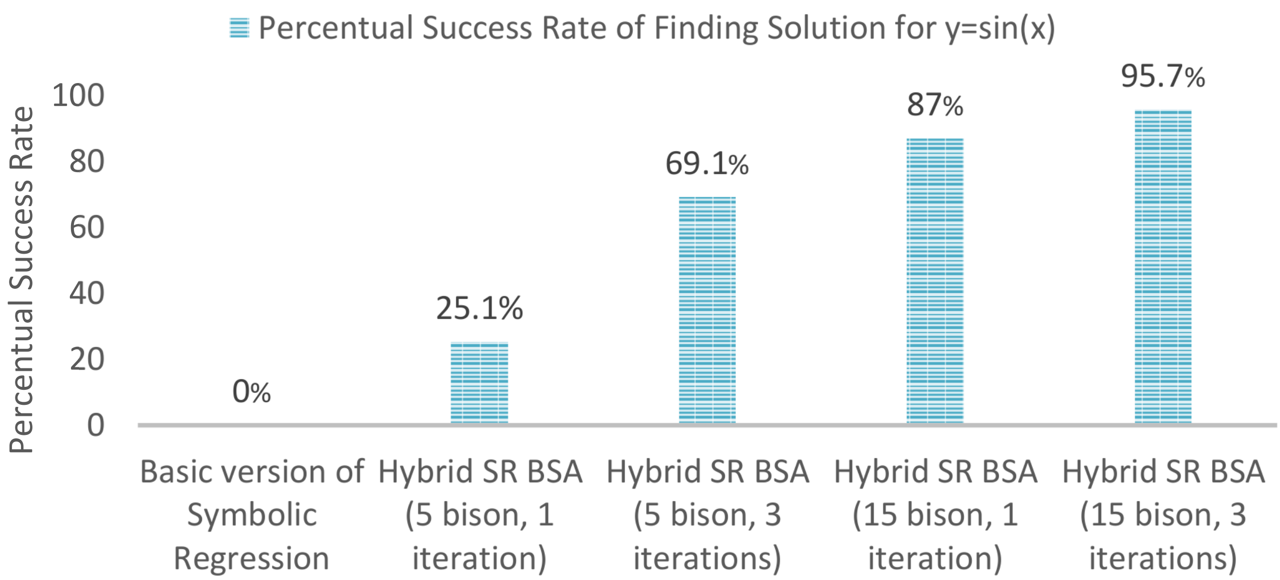


Fig. 6.2 Percentual success rate of finding the solution for $y = sin(x)$ comparing basic symbolic regression and several parameter variations of hybrid Bison Seeker Algorithm Symbolic Regression.

6.3 Dissertation Goal Fulfillment

This section describes the steps taken to fulfill the dissertation goals, which were set as follows:

1. **Map the current scene** of modern swarm algorithms, its trends, and challenges.
2. **Investigate** the methods addressing the weaknesses of swarm algorithms.
3. **Propose** a set of recommendations for new metaheuristics creation.
4. **Proof of concept testing**: Implement the proposed recommendations and methods in a new swarm algorithm.
5. **Evaluate the benefits** of the proposed algorithm for applied sciences.

There are multiple challenges in the development and modification of bio-inspired swarm algorithms. Mapping the current scene of modern metaheuristics and current trends revealed a variety of both optimization and existential problems. To answer the former, Section 3.1 named the methods addressing the optimization problems, while Section 3.2 proposed guidelines to avoid the existential ones. Furthermore, the recommendations for novel metaheuristics development were applied to prove the concept, as a new swarm-based algorithm was proposed and tested. Section 4.6 introduced several algorithm modifications, including a self-adaptive variant, as a modern modification trend representative. Finally, Section 6 evaluated the benefits of the proposed algorithm and its applications.

7 CONCLUSION

Nowadays, most new metaheuristics go round in circles repeating the same mistakes and facing prejudicial disrespect regardless of the actual quality of the presented method. Many researchers, scientists, and practitioners stand up against common malpractice and try to influence future metaheuristics towards a better standard. Most recently, at the turn of 2020/2021, a great number of publications dedicated to benchmarking issues and fairness in comparison, were published. However, advising a better approach, or pointing out others' mistakes, is not as powerful as applying the change proposed.

This work describes the current scene of the swarm algorithms, the state-of-the-art optimization techniques, modification trends, and reservations about the pitfalls of novel metaheuristic development. Detecting two types of struggles: optimization problems like stagnation or premature convergence, and existential problems connected to the criticism mentioned above, the Author proposes a new standard for developing future metaheuristics. But most importantly, these recommendations are applied to a showcase development project of a new swarm-based algorithm.

Following the recommendations led to the creation of an algorithm designed to tackle local optimum containment. The Bison Algorithm proposes a systematical scanning of the search space independently of the exploitation process. The suggested technique offers a way out of stagnation caused by local optimum confinement. Yet, it should be easy to implement for all kinds of problems from discrete, continuous, to large-scale, or other optimizations.

The algorithm was thoroughly examined, tested, and compared to other swarm optimization methods on the sum of 45 functions of IEEE CEC 2015 and 2017. The results show that the proposed algorithm is exceptionally competent when solving problems with many local optima.

Solving the recurrent problems of metaheuristic optimization may open the way for new challenges. The future might hold exciting discoveries like algorithm similarity detection systems, automatically assembled AI-based optimizers, neuroevolution, or new unexplored methods to tackle ubiquitous optimization problems.

This work did not aim to disclaim the No Free Lunch Theorem. It did not attempt to create a superlative optimizer that would solve every known possible problem. In fact, it aimed even higher. By setting preliminary rules and leading the way, this work presents one of many steps towards a meaningful development of metaheuristics yet to be created.

8 CURRICULUM VITAE

Anežka Kazíková

INFORMATION

Born 9th May 1990
Czech Republic

kazikova@utb.cz
+420 776 021 238

ADDRESS

Nová Lhota 282
696 74

LANGUAGES

Czech • maternity language
English • intermediate

SKILLS

PROGRAMMING

> Advanced knowledge
Python • C • C++

> Basic knowledge
C# • Java • PHP • MySQL
MatLab • Mathematica

WEB TECHNOLOGIES

HTML • CSS • JavaScript

INTEREST

Artificial Intelligence
Swarm Algorithms
Optimization

EDUCATION

2015 - NOW

Tomas Bata University in Zlín
/ Faculty of Applied Informatics
Doctoral studies: Engineering Informatics
Member of A.I. laboratory (ailab.fai.cz)

2013 - 2015

Tomas Bata University in Zlín / Faculty of Applied Informatics
Field of study: Engineering Informatics

2010 - 2013

Tomas Bata University in Zlín / Faculty of Applied Informatics
Field of study: Information and Control Technologies

WORK EXPERIENCE

2015 - NOW

Freelance programming (NetArk.cz s.r.o.)

2010 - 2016

Website development

ACHIEVEMENTS

2013 • EdHouse Award for Bachelor's Thesis
2015 • Dean's Award for Master's Thesis
2015 • Passed with honors

REFERENCES

[1] Awad, N. H., Ali, M. Z., Suganthan, P. N., Liang, J. J. and Qu, B. Y. Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization. *Technical Report, Nanyang Technological University, Singapore*. 2016, 2016, pp. 34.

[2] Bartz-Beielstein, T. et al. Benchmarking in Optimization: Best Practice and Open Issues. *arXiv:2007.03488 [cs, math, stat]*. Dec 2020. Available from: `<http://arxiv.org/abs/2007.03488>`. arXiv: 2007.03488.

[3] Berman, R. *American Bison*. Lerner Publications, Sep 2008. ISBN 978-0-8225-7513-9.

[4] Bernardino, H. S. and Barbosa, H. J. C. *Artificial Immune Systems for Optimization*, pp. 389–411. Studies in Computational Intelligence. Springer, 2009. doi: 10.1007/978-3-642-00267-0_14. Available from: `<https://doi.org/10.1007/978-3-642-00267-0_14>`. ISBN 978-3-642-00267-0.

[5] Biswas, A., Mishra, K. K., Tiwari, S. and Misra, A. K. Physics-Inspired Optimization Algorithms: A Survey, Jun 2013. ISSN 2356-752X. Available from: `<https://www.hindawi.com/journals/jopti/2013/438152/>`. DOI: https://doi.org/10.1155/2013/438152.

[6] Chakraborty, A. and Kar, A. K. *Swarm Intelligence: A Review of Algorithms*, 10, pp. 475–494. Springer International Publishing, 2017. doi: 10.1007/978-3-319-50920-4_19. Available from: `<http://link.springer.com/10.1007/978-3-319-50920-4_19>`. ISBN 978-3-319-50919-8.

[7] Csébfalvi, A. and Csébfalvi, G. Fair Comparison of Population-based Heuristic Approaches - The Evils of Competitive Testing. In *IJCCI 2012 - Proceedings of the 4th International Joint Conference on Computational Intelligence*, pp. 306–309, Dec 2012. doi: 10.5220/0004168403060309. ISBN 978-989-8565-33-4.

[8] Armas, J., Lalla-Ruiz, E., Tilahun, S. L. and Voss, S. Similarity in metaheuristics: a gentle step towards a comparison methodology. *Natural Computing*. Feb 2021. ISSN 1572-9796. doi: 10.1007/s11047-020-09837-9. Available from: `<https://doi.org/10.1007/s11047-020-09837-9>`.

[9] Del Ser, J., Osaba, E., Molina, D., Yang, X.-S., Salcedo-Sanz, S., Camacho, D., Das, S., Suganthan, P. N., Coello, C. A. C. and Herrera, F. Bio-inspired computation: Where we stand and what's next. *Swarm and Evolutionary Computation*. 2019, 48, pp. 220–250.

[10] Dorigo, M., Maniezzo, V. and Colorni, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*. 1996, 26, 1, pp. 29–41. doi: 10.1109/3477.484436.

[11] Eberhart, R. and Kennedy, J. New optimizer using particle swarm theory. In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39–43, 1995.

[12] Faris, H., Aljarah, I., Mirjalili, S., Castillo, P. A. and Merelo, J. J. EvoloPy: An Open-source Nature-inspired Optimization Framework in Python:. In *Proceedings of the 8th International Joint Conference on Computational Intelligence*, pp. 171–177. SCITEPRESS - Science and Technology Publications, 2016. doi: 10.5220/0006048201710177. Available from: `<https://www.scitepress.org/Papers/2016/60482/60482.pdf>`. ISBN 978-989-758-201-1.

[13] Felipe, C. and Claus, A. *EC Bestiary: A bestiary of evolutionary, swarm and other metaphor-based algorithms.* Zenodo, Jun 2018. doi: 10.5281/zenodo.1293352. Available from: `<https://doi.org/10.5281/zenodo.1293352>`.

[14] Holland, J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* A Bradford Book, reprint edition edition, Apr 1992. ISBN 978-0-262-58111-0.

[15] Hooker, J. N. Testing heuristics: We have it all wrong. *Journal of heuristics*. 1995, 1, 1, pp. 33–42.

[16] Ince, D. C., Hatton, L. and Graham-Cumming, J. The case for open computer programs. *Nature*. 2012, 482, 7386, pp. 485–488.

[17] Islam, M. R., Saifullah, C. M. K. and Mahmud, M. R. Chemical reaction optimization: survey on variants. *Evolutionary Intelligence*. Sep 2019, 12, 3, pp. 395–420. ISSN 1864-5917. doi: 10.1007/s12065-019-00246-1.

[18] Karaboga, D. An idea based on honey bee swarm for numerical optimization. Technical report, Technical report-tr06, Erciyes university, Engineering Faculty, 2005.

[19] Kazikova, A., Pluhacek, M. and Senkerik, R. Tuning of the Bison Algorithm Control Parameters. In Noelle, L., Burger, A., Tholen, C., Werner, J. and Wellhausen, J. (Ed.) *32ND European Conference on Modelling and Simulation (ECMS 2018)*, pp. 156–162, 2018. ISBN 978-0-9932440-7-0.

[20] Kazikova, A., Pluhacek, M. and Senkerik, R. Regarding the Behavior of Bison Runners Within the Bison Algorithm. *MENDEL*. Jun. 2018, 24, 1, pp. 63–70. doi: 10.13164/mendel.2018.1.063. Available from: <https://mendel-journal.org/index.php/mendel/article/view/24>.

[21] Kazikova, A., Pluhacek, M., Viktorin, A. and Senkerik, R. New Running Technique for the Bison Algorithm. In Rutkowski, L., Scherer, R., Korytkowski, M., Pedrycz, W., Tadeusiewicz, R. and Zurada, J. M. (Ed.) *Artificial Intelligence and Soft Computing*, Lecture Notes in Computer Science, pp. 417–426. Springer International Publishing, 2018. doi: 10.1007/978-3-319-91253-0_39. ISBN 978-3-319-91253-0.

[22] Kazikova, A., Oplatkova, Z. K., Pluhacek, M. and Senkerik, R. Border Strategies of the Bison Algorithm. In Iacono, M., Palmieri, F., Gribaudo, M. and Ficco, M. (Ed.) *Proceedings of the 33rd International ECMS Conference on Modelling and Simulation (ECMS 2019)*, 33 /

Communications of the ECMS, pp. 43–49, 2019. ISBN 978-3-937436-65-4.

[23] Kazikova, A., Pluhacek, M., Senkerik, R. and Viktorin, A. Proposal of a New Swarm Optimization Method Inspired in Bison Behavior. In Matoušek, R. (Ed.) *Recent Advances in Soft Computing*, Advances in Intelligent Systems and Computing, pp. 146–156. Springer International Publishing, 2019. doi: 10.1007/978-3-319-97888-8_13. ISBN 978-3-319-97888-8.

[24] Kazikova, A., Łapa, K., Pluhacek, M. and Senkerik, R. Cascade PID Controller Optimization Using Bison Algorithm. In Rutkowski, L., Scherer, R., Korytkowski, M., Pedrycz, W., Tadeusiewicz, R. and Zurada, J. M. (Ed.) *Artificial Intelligence and Soft Computing*, Lecture Notes in Computer Science, pp. 406–416. Springer International Publishing, 2020. doi: 10.1007/978-3-030-61401-0_38. ISBN 978-3-030-61401-0.

[25] Kazikova, A., Pluhacek, M., Kadavy, T. and Senkerik, R. Introducing the Run Support Strategy for the Bison Algorithm. In Zelinka, I., Brandstetter, P., Trong Dao, T., Hoang Duy, V. and Kim, S. B. (Ed.) *AETA 2018 - Recent Advances in Electrical Engineering and Related Sciences: Theory and Application*, Lecture Notes in Electrical Engineering, pp. 272–282. Springer International Publishing, 2020. doi: 10.1007/978-3-030-14907-9_27. ISBN 978-3-030-14907-9.

[26] Kazikova, A., Pluhacek, M. and Senkerik, R. Why Tuning the Control Parameters of Metaheuristic Algorithms Is So Important for Fair Comparison? *MENDEL*. Dec 2020, 26, 2, pp. 9–16. doi: 10.13164/mendel.2020.2.009.

[27] Kennedy, J. and Eberhart, R. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, 4, pp. 1942–1948 vol.4, Nov 1995. doi: 10.1109/ICNN.1995.488968.

[28] LaTorre, A., Molina, D., Osaba, E., Del Ser, J. and Herrera, F. Fairness in Bio-inspired Optimization Research:

A Prescription of Methodological Guidelines for Comparing Metaheuristics. *arXiv:2004.09969 [cs]*. Apr 2020. Available from: <http://arxiv.org/abs/2004.09969>. arXiv: 2004.09969.

[29] Lee, K. and Geem, Z. A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering*. 2005, 194, 36–38, pp. 3902–3933. doi: 10.1016/j.cma.2004.09.007.

[30] Merta, J. Hybrid Symbolic Regression with the Bison Seeker Algorithm. *MENDEL*. Jun 2019, 25, 1, pp. 79–86. ISSN 2571-3701. doi: 10.13164/mendel.2019.1.079.

[31] Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*. 2015, 89, pp. 228–249. doi: 10.1016/j.knosys.2015.07.006.

[32] Mirjalili, S. and Lewis, A. The Whale Optimization Algorithm. *Advances in Engineering Software*. 2016, 95, pp. 51–67. doi: 10.1016/j.advengsoft.2016.01.008.

[33] Mirjalili, S., Mirjalili, S. M. and Lewis, A. Grey Wolf Optimizer. *Advances in Engineering Software*. Mar 2014, 69, pp. 46–61. ISSN 0965-9978. doi: 10.1016/j.advengsoft.2013.12.007.

[34] Molina, D., Poyatos, J., Del Ser, J., Garcia, S., Hussain, A. and Herrera, F. Comprehensive Taxonomies of Nature- and Bio-inspired Optimization: Inspiration versus Algorithmic Behavior, Critical Analysis and Recommendations. *arXiv:2002.08136 [cs]*. Feb 2020. Available from: <http://arxiv.org/abs/2002.08136>. arXiv: 2002.08136.

[35] Passino, K. M. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*. Jun 2002, 22, 3, pp. 52–67. ISSN 1941-000X. doi: 10.1109/MCS.2002.1004010.

[36] Polakova, R., Tvrdík, J., Bujok, P. and Matoušek, R. Population-size adaptation through diversity-control mechanism for differential evo-

lution. In *MENDEL, 22th International Conference on Soft Computing*, pp. 49–56, 2016.

[37] Ramos, A., Petit, O., Longour, P., Pasquaretta, C. and Sueur, C. Collective decision making during group movements in European bison, Bison bonasus. *Animal Behaviour*. Nov 2015, 109, pp. 149–160. ISSN 0003-3472. doi: 10.1016/j.anbehav.2015.08.016.

[38] Ruiz, R. Heuristic Scheduling: Running away from the bio-inspired tsunami. Available from: https://github.com/fcampelo/EC-Bestiary/blob/master/bestiary_references/Presentation_RRuiz.pdf, 2012.

[39] Savsani, P. and Savsani, V. Passing vehicle search (PVS): A novel metaheuristic algorithm. *Applied Mathematical Modelling*. 2016, 40, 5–6, pp. 3951–3978. doi: 10.1016/j.apm.2015.10.040.

[40] Sörensen, K. Metaheuristics—the metaphor exposed. *International Transactions in Operational Research*. 2015, 22, 1, pp. 3–18.

[41] Sörensen, K. Metaphors and metaheuristics: A match made in hell. Available from: https://github.com/fcampelo/EC-Bestiary/blob/master/bestiary_references/Sorensen2018%20-%20plenary%20slides.pdf, 2018.

[42] Storn, R., Price, K. and others. Differential evolution–a simple and efficient adaptive scheme for global optimization over continuous spaces: technical report TR-95-012. *International Computer Science, Berkeley, California*. 1995.

[43] Weyland, D. A rigorous analysis of the harmony search algorithm: How the research community can be misled by a novel methodology. *International Journal of Applied Metaheuristic Computing (IJAMC)*. 2010, 1, 2, pp. 50–60.

[44] Yang, X.-S. Firefly algorithms for multimodal optimization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artifi-*

cial Intelligence and Lecture Notes in Bioinformatics). 2009, 5792 LNCS, pp. 169–178. doi: 10.1007/978-3-642-04944-6_14.

[45] Yang, X.-S. A New Metaheuristic Bat-Inspired Algorithm. *arXiv:1004.4170 [physics]*. Apr 2010. Available from: <http://arxiv.org/abs/1004.4170>. arXiv: 1004.4170.

[46] Yang, X.-S. and Deb, S. Cuckoo Search via Levy flights. In *2009 World Congress on Nature Biologically Inspired Computing (NaBIC)*, pp. 210–214, Dec 2009. doi: 10.1109/NABIC.2009.5393690.

[47] Yang, X.-S., Deb, S., Fong, S., He, X. and Zhao, Y.-X. From swarm intelligence to metaheuristics: nature-inspired optimization algorithms. *Computer*. 2016, 49, 9, pp. 52–59.

[48] Zelinka, I. *SOMA—self-organizing migrating algorithm*, pp. 167–217. Springer, 2004.

PUBLICATIONS OF THE AUTHOR

Journals

[1] 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. [accessed 2022-06-08]. ISSN 2169-3536. Available from: `https://ieeexplore.ieee.org/document/9378523`.

[2] KAZIKOVA, A., PLUHACEK, M. and SENKERIK, R. Why tuning the control parameters of metaheuristic algorithms is so important for fair comparison?. *Mendel*. 2020, vol. 26, iss. 2, pp. 9-16. [accessed. 2022-06-08]. ISSN 1803-3814. Available from: `https://mendel-journal.org/index.php/mendel/article/view/120`.

[3] KAZIKOVA, A., PLUHACEK, M. and SENKERIK, R. Regarding the behavior of bison runners within the Bison algorithm. *Mendel*. 2018, vol. 24, iss. 1, pp. 63-70. [accessed 2022-06-08]. ISSN 1803-3814. Available from: `https://mendel-journal.org/index.php/mendel/article/view/24`.

[4] PLUHACEK, Michal, KAZIKOVA, A., KADAVY, T., VIKTORIN, A. and SENKERIK, R. Relation of neighborhood size and diversity loss rate in particle swarm optimization with ring topology. *Mendel* . 2021, vol. 27, iss. 2, pp. 74-79. [accessed 2022-06-08]. ISSN 1803-3814. Available from: `https://mendel-journal.org/index.php/mendel/article/view/151/162`.

Conference Proceedings

[5] KAZIKOVA, A., PLUHACEK, M., SENKERIK, R. and VIKTORIN, A. Proposal of a new swarm optimization method inspired in bison behavior. In: *Advances in Intelligent Systems and Computing*. Brno: Springer Verlag, 2019, pp. 146-156. [accessed 2022-06-08]. ISSN

2194-5357. Available from: https://link.springer.com/chapter/10.1007/978-3-319-97888-8_13.

[6] KAZIKOVA, A., Krystian ŁAPA, PLUHACEK, M. and SENKERIK, R. Cascade PID controller optimization using bison algorithm. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* . Zakopane: Springer Science and Business Media Deutschland GmbH, 2020, pp. 406-416. [accessed 2022-06-08]. ISSN 0302-9743. Available from: https://link.springer.com/chapter/10.1007/978-3-030-61401-0_38.

[7] KAZIKOVA, A., KOMINKOVA OPLATKOVA, Z., PLUHACEK, M. and SENKERIK, R. Border strategies of the bison algorithm. In: *Proceedings of the 33rd International ECMS Conference on Modelling and Simulation (ECMS 2019)* . Napoli: European Council for Modelling and Simulation, 2019, pp. 43-49. [accessed 2022-06-08]. ISSN 2522-2414. Available from: http://www.scs-europe.net/dlib/2019/2019-0043.htm.

[8] KAZIKOVA, A., PLUHACEK, M., KADAVY, T. and SENKERIK, R. Introducing the run support strategy for the bison algorithm. In: *Lecture Notes in Electrical Engineering* . Ostrava: Springer Verlag, 2020, pp. 272-282. [accessed 2022-06-08]. ISSN 1876-1100. Available from: https://link.springer.com/chapter/10.1007/978-3-030-14907-9_27.

[9] KAZIKOVA, A., PLUHACEK, M., VIKTORIN, A. and SENKERIK, R. New running technique for the bison algorithm. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Zakopane: Springer Verlag, 2018, pp. 417-426. [accessed 2022-06-08]. ISSN 0302-9743. Available from: https://link.springer.com/chapter/10.1007/978-3-319-91253-0_39.

[10] KAZIKOVA, A., PLUHACEK, M. and SENKERIK, R. Performance of the bison algorithm on benchmark IEEE CEC 2017. In: *Advances*

in Intelligent Systems and Computing. Springer Verlag, 2019, pp. 445-454. [accessed 2022-06-08]. ISSN 2194-5357. Available from: `https://link.springer.com/chapter/10.1007/978-3-319-91189-2_44`.

[11] KAZIKOVA, A., PLUHACEK, M. and SENKERIK, R. Tuning of the bison algorithm control parameters. In: *Proceedings - European Council for Modelling and Simulation, ECMS*. Wilhelmshaven: European Council for Modelling and Simulation, 2018, pp. 156-162. [accessed 2022-06-08]. ISSN 2522-2414. Available from: `http://www.scs-europe.net/dlib/2018/2018-0156.htm`.

LIST OF FIGURES

Fig. 3.1 The number of metaheuristic proposals in the years 1973-2018. 11
Fig. 3.2 Number of publications citing the Harmony Search Algorithm versus the exposure publication in 2005-2020. 13
Fig. 4.1 Simplified flowchart of the Bison Algorithm. 18
Fig. 4.2 Flowchart of swarming movement. 20
Fig. 4.3 Cumulative movement of the swarming group (a) and the running group (b) on the 2-dimensional Rastrigin's Function. . . 21
Fig. 4.4 Pipeline of the Bison Algorithm development process. . . . 22
Fig. 5.1 Mean diversity convergences example (IEEE CEC 2015 in 100 dimensions. 24
Fig. 5.2 Convergence of all runs of the swarm algorithms compared on IEEE CEC 2017 Function F4 in 30 dimensions. 24
Fig. 5.3 Rank comparison of the BIA, CS, PSO, BAT, and FFA on benchmarks CEC 2015 and CEC 2017 (Friedman Rank Test, $p<0.05$). 25
Fig. 5.4 Performance measure comparing the number of problems with a significantly better solution on the set with characteristic feature (Wilcoxon $p<0.05$). 26
Fig. 6.1 Metaheuristic comparison: a) average time consumption of each algorithm during the experiment, b) how often one algorithm delivered a final solution of quality in the top 5%, c) how many times the algorithm delivered the worst results. 29

Fig. 6.2 Percentual success rate of finding the solution for $y = sin(x)$ comparing basic symbolic regression and several parameter variations of hybrid Bison Seeker Algorithm Symbolic Regression. . 29

LIST OF TABLES

Tab. 3.1 Swarm-inspired metaheuristics sorted by the number of citations of the original proposal paper (accessed 27/01/2021). . 10
Tab. 5.1 Winning Algorithms on CEC 2015 (Wilcoxon Rank-Sum test, p<0.05). 24
Tab. 5.2 Winning Algorithms on CEC 2017 (Wilcoxon Rank-Sum test, p<0.05). 24

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
BAT	Bat Algorithm
BIA	Bison Algorithm
BSA	Bison Seeker Algorithm
CEC	Congress on Evolutionary Computation
CS	Cuckoo Search
D	Dimension
DE	Differential Evolution
EC	Evolutionary Computation
EG	Elite group
FFA	Firefly Algorithm
GA	Genetic Algorithm
IEEE	Institute of Electrical and Electronics Engineers
NP	Population size
PSO	Particle Swarm Optimization
RG	Running group
SG	Swarming Group
SR	Symbolic Regression

Vývoj a modifikace moderních bio-inspirovaných hejnových algoritmů

Development and Modification of Modern Bio-Inspired Swarm Algorithms

Doctoral Thesis Summary

Published by: Tomas Bata University in Zlín

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

Edition: published electronically

Typesetting by: Ing. Anezka Kazikova, Ph.D.

This publication has not undergone any proofreading or editorial review.

First Edition

Publication year: 2022

ISBN 978-80-7678-124-5