A Proposed Framework for the Design and Analysis of Metaheuristics

by

Iannick GAGNON

MANUSCRIPT-BASED THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE IN PARTIAL FULFILLMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, MARCH 3, 2026

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Professor Alain April, Thesis Supervisor
Department of Software and IT Engineering at École de technologie supérieure

Professor Alain Abran, Thesis Co-Director
Department of Software and IT Engineering at École de technologie supérieure

Professor Antoine Tahan, Chair, Board of Examiners
Department of Mechanical Engineering at École de technologie supérieure

Professor Sylvie Ratté, Member, Board of Examiners
Department of Software and IT Engineering at École de technologie supérieure

Ms. Cherifa Mansoura, External Independent Examiner
Business Architect at TEKsystems

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON DECEMBER 15, 2025

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

FOREWORD

The completion of this doctoral thesis has been both intellectually demanding and deeply fulfilling. My research has centered on the analysis of methodological practices in metaheuristics, a field that has captivated my interest since the early stages of my graduate studies. Presented as a collection of peer-reviewed publications, this work represents the culmination of several years of sustained effort, inquiry, and reflection.

My engagement with metaheuristics began during the preliminary phase of my doctoral work. Initially, my intention was to employ optimization algorithms to automate the architectural design of convolutional neural networks for control chart pattern recognition, which was the topic I explored in my master's dissertation. I quickly recognized that metaheuristics offered an appealing balance between solution accuracy and computational efficiency for this purpose. As I deepened my understanding of the field, two persistent questions emerged:

1. Why does this algorithm work?
2. How is this algorithm different from other algorithms?

Over time, it became increasingly clear to me that, within the literature, a depth of understanding was often sacrificed in favor of quantity and a form of tokenism (i.e., diversity for its own sake), particularly in the way analyses were conducted and presented. This realization marked the beginning of my journey to demonstrate how even modest methodological adjustments, combined with a sharper focus, can yield valuable and previously overlooked scientific insights in metaheuristics research.

The first article, "A Critical Analysis of the Bat Algorithm", arose from my desire to scrutinize a widely cited metaheuristic by revealing that its underlying methodology failed to distinguish its constituent mechanisms. Specifically, we show that it is in fact a hybrid of Particle Swarm Optimization and Simulated Annealing, both of which are long-established canonical methods. Furthermore, we demonstrate how rigorous

comparative analyses can refute claims of its general superiority over competing algorithms.

The second article, “An Investigation of the Effects of Chaotic Maps on the Performance of Metaheuristics”, shifts attention from a single algorithm to a commonly employed algorithmic component: chaotic maps. Although widely believed to enhance performance, the evidence for such claims is limited. This article challenges the notion of universal benefit, proposes that sequence effects explain much of the reported impact, and provides empirical support for this interpretation.

The third article, “A Quantitative Evaluation of Statistical Practices in Metaheuristics Research”, extends the line of inquiry established in the preceding studies. It assesses the current state of methodological practices and underscores the seriousness of the issues identified.

It is my hope that the findings and perspectives offered in this thesis will not only advance the methodological rigor of metaheuristics research but also encourage future scholars to approach the field with the same blend of critical analysis and curiosity that has guided my own work.

ACKNOWLEDGEMENTS

Throughout the writing of this dissertation, I have received invaluable encouragement and assistance. I would like to express my deepest gratitude to my thesis supervisor, Professor Alain April, and my co-director, Professor Alain Abran, for their exceptional guidance and support. Their expertise and insights have been instrumental in shaping the direction and quality of this work.

I am grateful to my wife, Yasaman Dorraji, whose support and encouragements have been an enduring source of strength throughout my doctoral journey. During the course of this work, we welcomed our three children: Jacob, Arielle, and our newborn son, Zacharie, into our lives. Amid the demands of parenthood and the challenges of research, her presence provided the motivation that made the completion of this work possible.

Finally, I extend my heartfelt appreciation to my mother, whose sacrifices and work ethic instilled in me have been a driving force in bringing this dissertation to completion.

"For since the fabric of the universe is most perfect and the work of a most wise Creator, nothing at all takes place in the universe in which some rule of maximum or minimum does not appear."

— Leonhard Euler

Un nouveau cadre d'applications pour la conception et l'analyse des métaheuristiques

Iannick GAGNON

RÉSUMÉ

Cette thèse de doctorat propose une évaluation critique et une amélioration des pratiques méthodologiques dans la recherche sur les métaheuristiques. Ces algorithmes stochastiques de type « boîte noire » sont largement utilisés pour résoudre des problèmes NP-complets, tels que l'optimisation de tournées de véhicules, l'entraînement de réseaux neuronaux ou la conception de structures aérospatiales. Malgré leur popularité, la littérature révèle un manque de rigueur méthodologique, de transparence et de reproductibilité, compromettant la validité scientifique des résultats publiés.

Pour répondre à ces lacunes, la recherche s'appuie sur trois études complémentaires : (1) une analyse critique de l'algorithme de la chauve-souris, démontrant ses similitudes structurelles avec des méthodes antérieures et l'absence de preuves robustes de sa supériorité; (2) une investigation des effets des cartes chaotiques, révélant que les gains de performance rapportés dans la littérature sont souvent dus à des effets de séquence plutôt qu'à des propriétés intrinsèques; (3) une évaluation quantitative des pratiques statistiques dans articles scientifiques récents, mettant en évidence des insuffisances systématiques dans l'usage des tests d'hypothèse, des tailles d'effet et des intervalles de confiance.

À partir de ces constats, la thèse propose le Metaheuristics Design and Analysis Framework (MDAF), un cadre méthodologique structuré qui intègre des lignes directrices pour la conception expérimentale, l'analyse statistique et la présentation des résultats. L'adoption du MDAF vise à renforcer la qualité, la reproductibilité et la pertinence des recherches en métaheuristiques, ouvrant la voie à des solutions d'optimisation plus robustes et scientifiquement fiables.

Mots-clés: métaheuristiques, optimisation mathématique, optimisation stochastique, pratiques méthodologiques

A proposed framework for the design and analysis of metaheuristics

Iannick GAGNON

ABSTRACT

This doctoral thesis critically evaluates and seeks to improve methodological practices in metaheuristics research. These black-box stochastic optimization algorithms are widely applied to NP-complete problems, including vehicle routing, neural network training, and aerospace structure design. Despite their popularity, the literature reveals significant shortcomings in methodological rigor, transparency, and reproducibility, which undermine the scientific validity of reported results.

To address these issues, the research is structured around three complementary studies: (1) a critical analysis of the Bat Algorithm, revealing its structural similarities to earlier methods and the lack of robust evidence supporting its claimed superiority; (2) an investigation into the use of chaotic maps, showing that reported performance gains often stem from sequence effects rather than intrinsic properties; (3) a quantitative assessment of statistical practices in 70 recent articles, highlighting systematic deficiencies in hypothesis testing, effect size reporting, and confidence interval usage.

Building on these findings, the thesis introduces the Metaheuristics Design and Analysis Framework (MDAF) as a structured methodological framework integrating guidelines for experimental design, statistical analysis, and transparent reporting. Adoption of the MDAF aims to enhance the quality, reproducibility, and practical relevance of metaheuristics research, ultimately contributing to more robust and scientifically credible optimization solutions.

Keywords: metaheuristics, mathematical optimization, stochastic optimization, methodological practices

TABLE OF CONTENTS

Page

INTRODUCTION 27

CHAPTER 1 LITERATURE REVIEW 31
1.1 Historical overview 31
1.1.1 Early period 31
1.1.2 Proliferation period 34
1.1.3 Metaphor-centric period 35
1.2 Theoretical shortcomings 36
1.3 Methodological shortcomings 37
1.3.1 Transparency and reproducibility 38
1.3.2 Experimental design 38
1.3.3 Statistical methodology 40
1.4 Frameworks 42

CHAPTER 2 OVERVIEW OF PUBLISHED ARTICLES 47
2.1 A Critical Analysis of the Bat Algorithm (Gagnon, April & Abran, 2020) 47
2.1.1 Aim 488
2.1.2 Methodology 48
2.1.3 Key results 50
2.1.4 Discussion 50
2.2 An Investigation of the Effects of Chaotic Maps on the Performance of Metaheuristics (Gagnon, April & Abran, 2020) 52
2.2.1 Aim 52
2.2.2 Methodology 53
2.2.3 Key results 53
2.2.4 Discussion 54
2.3 A Quantitative Evaluation of Statistical Practices in Metaheuristics Research (Gagnon, Abran & April, 2025) 555
2.3.1 Aim 55
2.3.2 Methodology 55
2.3.3 Key results 56
2.3.4 Discussion 56
2.4 Conclusion 58
2.4.1 Contributions 58
2.4.2 Future directions 59

CHAPTER 3 A CRITICAL ANALYSIS OF THE BAT ALGORITHM 61
3.1 Introduction 62
3.2 The particle swarm optimization metaheuristic 63
3.3 The bat algorithm metaheuristic 66

3.4 Comparative analysis 69
3.5 Methodology 73
3.5.1 Test set 74
3.5.2 Parameter selection 80
3.5.3 Experiments and nonparametric statistical tests 82
3.5.3.1 Investigating the research hypotheses 82
3.5.3.2 The Wilcoxon-Mann-Whitney Test 83
3.6 Results 84
3.6.1 Sensitivity to initial conditions 84
3.6.2 BA with roulette wheel selection instead of pulse rate 85
3.6.3 BA without the loudness mechanism 87
3.6.4 Additional observations 87
3.7 Conclusion 90

CHAPTER 4 AN INVESTIGATION OF THE EFFECTS OF CHAOTIC MAPS ON THE PERFORMANCE OF METAHEURISTICS 93
4.1 Introduction 94
4.1.1 Chaotic maps 95
4.1.2 Particle swarm optimization and simulated annealing 101
4.2 Literature review 103
4.3 Methodology 104
4.3.1 Definition of performance 104
4.3.2 Null hypothesis statistical testing and effect size. 105
4.3.3 Parameter selection for PSO 105
4.3.4 Test set 106
4.4 Results and analysis for particle swarm optimization 106
4.5 Results and analysis for simulated annealing 108
4.6 Limitations of this study 109
4.7 Conclusion 110

CHAPTER 5 A QUANTITATIVE EVALUATION OF STATISTICAL PRACTICES IN METAHEURISTICS RESEARCH 113
5.1 Introduction 114
5.2 Methodology 115
5.3 Results and discussion 126
5.4 Conclusion 141

CONCLUSION 143

APPENDIX I THE MDAF 145

APPENDIX II MINIMUM RESEARCH RECEIVABILITY CRITERIA 205

APPENDIX III PUBLICATION FIGURES AND TABLES CHECKLIST 207

APPENDIX IV ANALYSIS CHECKLIST 209

APPENDIX V DESCRIPTIVE STATISTICS CHECKLIST.....211

BIBLIOGRAPHY 213

LIST OF TABLES

Page

Table 2.1 Mapping of published articles to research objectives 47

Table 3.1 BA parameter settings for grid search 81

Table 3.2 Parameter settings for BA and PSO 81

Table 3.3 Impact of initialization scheme on median FHT for BA 84

Table 3.4 Impact of initialization scheme on median FHT for PSO 85

Table 3.5 Effect of replacing pulse rate with roulette wheel selection 86

Table 3.6 Effect of initialization on BA with roulette wheel selection 67

Table 3.7 Effect of removing the loudness mechanism 87

Table 4.1 Parameters for PSO and CPSO 106

Table 4.2 Median FHT for PSO and CPSO 107

Table 4.3 Wilcoxon Rank-Sum test p-values for CPSO 107

Table 4.4 PSO and CPSO median FHT differences 95% CI 108

Table 4.5 Median FHT for SA and CSA 109

Table 4.6 Wilcoxon Rank-Sum test p-values for CSA 109

Table 5.1 Criteria for descriptive statistics 119

Table 5.2 Criteria for null-hypothesis statistical testing 123

LIST OF FIGURES

Page

Figure 1.1 Relative rise in publications between 2000 and 2023 35

Figure 3.1 Effect of γ on pulse rate 68

Figure 3.2 Effect of α on loudness 68

Figure 3.3 Relative rise in publications between 2000 and 2023 70

Figure 3.4 Empirical correlation matrix of the five benchmark functions 76

Figure 3.5 Dataset with a correlation coefficient of 0.25 76

Figure 3.6 Sphere function 77

Figure 3.7 Rosenbrock function 77

Figure 3.8 Step function 78

Figure 3.9 Noisy quartic 79

Figure 3.10 Shekel's foxholes 79

Figure 3.11 BA centroids on f_1 88

Figure 3.12 BA centroids on f_2 88

Figure 3.13 BA centroids on f_3 89

Figure 3.14 BA centroids on f_4 89

Figure 3.15 HappyCat function with $\alpha = 1/8$ on $x_i \in [-2, 2]$ 90

Figure 4.1 Number of yearly publications on optimization with chaos 95

Figure 4.2 Chaotic behaviour of $x_{t+1} = 4x_t\,(1 - x_t)$ 96

Figure 4.3 Difference between chaotic maps with different initial conditions 96

Figure 4.4 Chaotic maps 97

Figure 4.5 Chebyshev map EPDF 98

Figure 4.6 Circle map EPDF 98
Figure 4.7 Gauss map EPDF 99
Figure 4.8 Logistic map EPDF 100
Figure 4.9 Sine map EPDF 100
Figure 4.10 Tent map EPDF 101
Figure 5.1 Evaluation process 117
Figure 5.2 Data generation process 118
Figure 5.3 Compilation of coverage criteria in category A 126
Figure 5.4 Coverage for criterion A.1 127
Figure 5.5 Coverage for criterion A.2 128
Figure 5.6 Coverage for criterion A.3 129
Figure 5.7 Coverage for criterion A.4 130
Figure 5.8 Coverage for criterion A.5 130
Figure 5.9 Coverage for criterion A.6 131
Figure 5.10 Coverage for criterion A.7 131
Figure 5.11 Coverage for criterion A.8 132
Figure 5.12 Coverage for criterion A.9 133
Figure 5.13 Coverage for criterion A.10 133
Figure 5.14 Coverage for criterion A.11 134
Figure 5.15 Compilation of coverage of criteria in category B 135
Figure 5.16 Coverage for criterion B.1 135
Figure 5.17 Coverage for criterion B.2 136
Figure 5.18 Coverage for criterion B.3 137

Figure 5.19 Coverage for criterion B.4 138

Figure 5.20 Coverage for criterion B.5 138

Figure 5.21 Coverage for criterion B.6 139

Figure 5.22 Coverage for criterion B.7 139

Figure 5.23 Coverage for criterion B.8 140

Figure 5.24 Coverage for criterion B.9 140

LIST OF ALGORITHMS

Page

Algorithm 3.1 Pseudocode for minimization with PSO 65

Algorithm 3.2 Pseudocode for minimization with BA 67

Algorithm 3.3 Pseudocode of component 2 of BA 71

Algorithm 3.4 Pseudocode of component 3 of BA 72

Algorithm 4.1 Pseudocode for minimization with SA 106

LIST OF ABBREVIATIONS AND ACRONYMS

ACO	Ant Colony Optimization
AMA	American Medical Association
APA	American Psychological Association
BCa	Bias-corrected and Accelerated
CEC	Congress on Evolutionary Computation
CI	Confidence interval
CLT	Central limit theorem
CPSO	Chaotic Particle Swarm Optimization
CPU	Central processing unit
CSA	Chaotic Simulated Annealing
DRY	Don’t Repeat Yourself
DOE	Design of experiment
DOI	Digital object identifier
ES	Evolution strategies
FDR	False discovery rate
FHT	First hitting time
FWER	Familiwise Error Rate
GA	Genetic Algorithm
GLP	Good Laboratory Practice
GUI	Graphical user interface
GLP4OPT	Good Laboratory Practice for Optimization

HSD	Honestly Significant Difference
IQR	Interquartile range
MDAF	Metaheuristics Design and Analysis Framework
NHST	Null hypothesis statistical testing
OSF	Open Science Framework
PDF	Probability density function
PSO	Particle Swarm Optimization
RO	Research objective
RQ	Research question
SA	Simulated Annealing
SAMPL	Statistical Analyses and Methods in the Published Literature
SN-FHT	Success-normalized first hitting time
SUT	System under test
TS	Tabu Search
TSP	Traveling salesman problem
VCS	Version control system
VRP	Vehicle routing problem

INTRODUCTION

Metaheuristics are a powerful class of optimization algorithms that have transformed the approach to complex, high-dimensional problems across science, engineering, and industry. Yet, despite their success, concerns about methodological rigor have become increasingly prominent. This chapter introduces the research by placing it within its broader scientific and methodological context. It then defines the research aim, presents the research questions, and outlines the objectives that guide the study. Finally, it describes the methodological approach used to achieve these objectives.

Context

Metaheuristics have become increasingly prominent for their ability to address complex, high-dimensional optimization problems under the practical constraints of time and computational resources, offering a deliberate balance between performance and cost. Over recent decades, the literature has expanded to include hundreds of metaphor-based algorithms, each aiming to balance exploration and exploitation across the search space.

However, recent critiques contend that, despite the field's rapid growth, its methodological standards have declined markedly (Tzanetos & Dounias, 2020; Camacho-Villalón et al., 2023). While early contributions often combined strong theoretical foundations with rigorous empirical evaluation, the current research culture tends to prioritize novelty over scientific reliability (Hooker, 1995). This shift has led to the proliferation of methods with questionable originality and limited theoretical and empirical support (Sörensen, 2015).

In response, this thesis seeks to strengthen the methodological foundations of metaheuristics research by establishing clearer, more robust standards and by providing tools to support the scientific design and analysis of metaheuristic algorithms.

Aim

The aim of this research is to strengthen the methodological foundations of metaheuristics research by identifying recurring weaknesses in current practices and developing a structured framework to address them. This aim is pursued through two guiding research questions (RQs). To answer these questions, the work is organized around three research objectives (ROs). A mapping between the three peer-reviewed articles and the ROs is provided in Table 2.1.

RQ1 - What are the most significant methodological deficiencies in current metaheuristics research, and how do they impact the scientific quality of results?

This question seeks to establish a clear diagnosis of the field's methodological landscape. It involves a critical review of published literature to identify common shortcomings in areas such as experimental design, performance measurement, statistical analysis, and reporting transparency. By clarifying how these deficiencies affect the credibility and reproducibility of results, RQ1 provides the empirical basis for developing targeted methodological improvements.

RQ2 - How can a structured framework be defined to address these deficiencies and promote more rigor, transparency, and reproducibility?

This question focuses on the design of the Metaheuristics Design and Analysis Framework (MDAF). The framework integrates prescriptive guidelines for experimental design, statistical principles, and reporting standards, aiming to provide a standardized approach for metaheuristics research. RQ2 translates the findings from RQ1 into actionable methodological recommendations.

RO1 - Diagnose deficiencies in current practice

Identify and critically analyze methodological weaknesses in metaheuristics research through three targeted peer-reviewed studies, each providing concrete evidence of specific

deficiencies. Together, these studies expose systematic issues that affect the rigor, transparency, and reproducibility of the results.

RO2 - Develop a standardized framework

Create the MDAF to address the deficiencies identified in RO1. The framework integrates prescriptive guidelines that promote rigor, transparency, and reproducibility in metaheuristics research. This is accomplished by providing positive examples of sound methodological practices, offering clear statistical and experimental design recommendations, as demonstrated throughout the three peer-reviewed studies.

RO3 - Empirically validate the framework

Apply the MDAF developed for RO2 to specific case studies via controlled experiments to assess its practical utility. This RO tests whether the framework improves research quality and enables fairer, more meaningful comparisons between algorithms.

Methodology and structure

To attain the previous ROs, this thesis adopts a multifaceted methodological approach that integrates theoretical and empirical dimensions consisting of the following elements:

- **Literature review:** A review of existing literature to identify prevailing practices, recurring limitations, and methodological gaps is provided in and prefaced by a historical contextualization of the field's research practices.
- **Case studies:** In-depth examinations of specific algorithms that define and illustrate how the proposed framework translates into more nuanced and robust results through three peer-reviewed research articles are presented.
- **Framework:** A structured presentation of the proposed framework's guidelines for experimental design, analysis and reporting of metaheuristics research is provided.

Together, these elements provide a broad but cohesive foundation for evaluating current research practices, identifying their limitations, and demonstrating how structured improvements can advance the field in a scientifically robust manner.

CHAPTER 1

LITERATURE REVIEW

This chapter provides a comprehensive review of the historical development, current state, and critical evaluation of metaheuristics research. It covers the evolution of metaheuristics, from their early inception, and the ensuing proliferation, until the present. Research gaps are identified, including the lack of theoretical foundations and suboptimal research practices are highlighted. The role of metaheuristics frameworks in advancing the field is also discussed.

1.1 Historical overview

Metaheuristics are a class of high-level, stochastic optimization algorithms designed to find approximate solutions to complex optimization problems that are otherwise computationally infeasible to solve exactly. There have been significant developments since their inception and metaheuristics have been mainly used to obtain approximate solutions to NP-complete problems such as the Traveling Salesman Problem (TSP), the Knapsack Problem, the Graph Colouring problem and the Vehicle Routing Problem (VRP), all of which are canonical examples.

These problems are characterized by large, complex search spaces where finding exact solutions becomes impractical as problem size increases. Metaheuristics offer practical workarounds by providing high-quality approximate solutions in a reasonable amount of time, making them indispensable tools in a wide range of real-world applications.

1.1.1 Early period

The development of metaheuristics began in the 1950s with the study of stochastic optimization algorithms, which use random variables to blend the explorative and greedy components of iterative search procedures. The introduction of randomness also confers the following benefits:

1. Facilitates the escape from local optima.
2. Enables the exploration of a wider range of potential solutions.
3. Reduces the sensitivity to small perturbations or noise.

These advantages accelerate progress and improve robustness at the cost of exactness. It is especially useful for the problems previously listed that rapidly become intractable as the problem instances grow. For example, the TSP involves finding the shortest possible route that visits each city exactly once and returns to the original city. Finding the exact solution is computationally intractable for large numbers of cities, but algorithms like genetic algorithms (GAs), simulated annealing (SA), and ant colony optimization (ACO) can provide near-optimal solutions in relatively short times depending on problem size. Another example is the Knapsack Problem, where the task is to determine the number of items from different categories to include in a collection so that the total weight is within a given limit and the total value is maximized. Exact solutions are challenging to obtain for large item sets. For example, the Graph Coloring problem involves assigning colors to the vertices of a graph (in the mathematical sense) so that no two adjacent vertices share the same color while also using the minimum number of colors. Lastly, the Vehicle Routing Problem (VRP) requires determining the optimal set of routes for a fleet of vehicles who make deliveries to a given set of customers. Exact solutions to these problems rapidly become intractable for large instances, but heuristic and metaheuristic approaches can yield near-optimal solutions in relatively short times.

The 1960s marked a significant milestone in the history of metaheuristics with the introduction of Evolution Strategy (ES) algorithms. Originating from the work of Ingo Rechenberg (Rechenberg, 1965) and Hans-Paul Schwefel (Schwefel, 1965) in the mid-1960s, ES were among the first optimization techniques to leverage the principles of natural evolution. These algorithms utilized mechanisms such as mutation, recombination, and selection to evolve a population of candidate solutions, increasing their quality or *fitness* over successive generations. The rigorous analyses accompanying ES provided valuable insights into their behavior and laid the groundwork for future developments in the field.

Building on the principles of ES, John Holland introduced Genetic Algorithms (GAs) in the 1970s (Holland, 1975). Inspired by the process of natural selection and genetics, GAs employed similar operators such as crossover, mutation, and selection to evolve candidate solutions. The flexibility and robustness of GAs allowed them to be applied to a wide range of problems, from engineering design to artificial intelligence. Holland's work was pivotal in establishing the theoretical foundations of GAs and demonstrated their potential to solve complex optimization problems effectively.

SA, introduced by Kirkpatrick, Gelatt, and Vecchi in the 1980s (Kirkpatrick, Gelatt & Vecchi, 1983), drew inspiration from the physical annealing process in metallurgy. This algorithm probabilistically accepted worse solutions to escape local minima, gradually decreasing the probability of such acceptances as the search progresses to eventually converge at a fixed point. SA's capacity to probabilistically escape local optima made it especially effective for navigating rugged or noisy search landscapes. The mathematical rigor underlying SA's acceptance criteria and cooling schedule provided a strong theoretical basis for its application across various domains.

While these early metaheuristics varied in inspiration, they were often supported by formal models and well-defined mechanisms. Their development prioritized interpretability and theoretical soundness, offering a level of methodological clarity that contrasts with many modern metaphor-driven variants. At the same time, this formative period was marked by the absence of standardized evaluation protocols, which hindered the comparability and cumulative advancement of results across studies. Notably, several recurring structural patterns began to emerge, such as population-based exploration and acceptance criteria for non-improving solutions, suggesting the potential value of a component-based perspective for metaheuristic design.

1.1.2 Proliferation period

In the late 1980s, Fred Glover introduced Tabu Search (TS), an optimization technique that uses memory structures to avoid revisiting previously explored inferior solutions. By maintaining a list of forbidden or "tabu" moves, TS enhances the exploration capabilities of

the search process, preventing cycles and encouraging the exploration of new areas in the solution space. The strategic use of memory and adaptive mechanisms in TS distinguished it from other metaheuristics and contributed to its success in solving difficult optimization problems.

The 1990s saw the introduction of several other influential metaheuristic algorithms, including ACO by Marco Dorigo (Dorigo, 1992) and Particle Swarm Optimization (PSO) by James Kennedy and Russell Eberhart (Kennedy & Eberhart, 1995). ACO, inspired by the foraging behavior of ants, utilized a memory structure based on evaporating pheromone trails to guide the search process, while PSO, inspired by the social behavior of birds and fish, relied on the collective intelligence of a swarm of generically named “particles” to find optimal solutions. Both algorithms further expanded the repertoire of metaheuristic techniques and demonstrated the potential of biologically inspired approaches.

This period was marked by a surge in interest and publication volume as researchers increasingly sought inspiration from natural and social systems. While many of these methods introduced novel metaphorical narratives, their methodological foundations varied considerably in depth and rigor. The emphasis began to shift from theoretical analysis toward performance demonstration, often based on benchmark results without standardized evaluation criteria.

As the number of proposed algorithms grew, so did the need for comparative analysis. However, a lack of shared benchmarks, reproducibility standards, and statistically grounded methodologies limited the interpretability and generalizability of results. This proliferation period laid the groundwork for both the expansion of the field and the emergence of critical methodological concerns that continue to shape current research practices.

1.1.3 Metaphor-centric period

Taking the year 200 as baseline, the number of publications for which the topic is labelled as “metaheuristics” increased by a factor of more than 40 compared to a factor of less than 10 for publications whose topic is labelled as “optimization” (Figure 1.1). This trend has

been critically examined in the literature, notably by (Sörensen, 2015), (Tzanetos & Dounias, 2020) and (Aranha et al., 2022), who argue that the metaphorization of algorithm development has, in many cases, supplanted rigorous scientific reasoning.

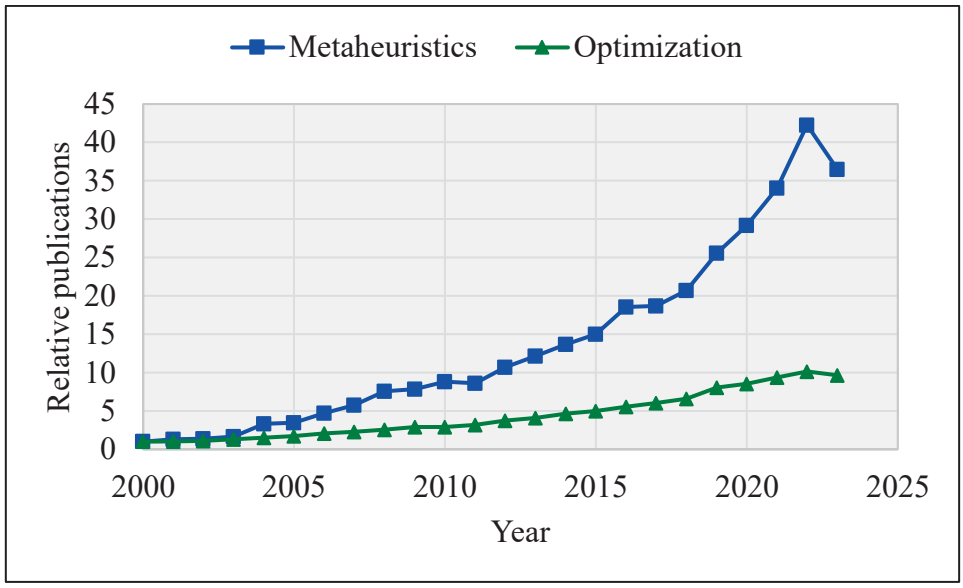

Figure 1.1 Relative rises in publications between 2000 and 2023

Contrasting perspectives also exist, such as (Fausto et al., 2019) for example, who commend the use of diversified metaphors, exemplified by titles like “From ants to whales: metaheuristics for all tastes,” which is expressed without any negative connotation.

Throughout this period, the field of metaheuristics experienced substantial growth, fueled by imaginative algorithmic proposals rooted in increasingly abstract or symbolic inspirations. This surge in creativity yielded a wide array of techniques, many of which contributed to solving challenging problems across disciplines. Yet, the influx of metaphor-based contributions was not always matched by a corresponding investment in methodological rigor. The shift from theory-driven development to narrative-driven design introduced a growing disconnect between algorithmic novelty and empirical or analytical substantiation. This emerging gap between conceptual innovation and scientific validation will be explored in the following sections.

1.2 Theoretical shortcomings

The shift towards metaphor-based algorithms highlighted a significant issue in metaheuristics research practices: the tendency to prioritize quantity over depth of understanding as detailed in (Sörensen, 2015), (Sörensen, Sevaux & Glover, 2017), (Tzanetos & Dounias, 2020) and (Aranha et al., 2022). These authors argue that many new algorithms are introduced with minimal empirical validation and mathematical description. As a result, the field has seen a proliferation of methods accompanied by unsubstantiated and often unverifiable claims of superiority.

In many cases, contributions differ only superficially from existing methods, with minor parameter changes or symbolic reinterpretations repackaged under new metaphors. This practice not only dilutes the perceived novelty of genuine innovations but also contributes to a fragmented and inconsistent research landscape. The emphasis on publication volume, sometimes driven by pressures within the academic and funding ecosystems, has further incentivized the rapid production of algorithms without adequate methodological rigor.

Such conditions have made it increasingly difficult to assess which algorithms truly offer meaningful advances. Without standardized benchmarks, transparent experimental protocols, and proper statistical validation, comparative claims lack scientific credibility. These shortcomings hinder the cumulative progress of the field and obscure valuable insights that might otherwise emerge through more disciplined and reproducible research practices.

One of the primary concerns with many metaphor-based metaheuristics is the lack of a clear theoretical foundation. Unlike classical algorithms such as Genetic Algorithms (Holland, 1975) and Simulated Annealing (Kirkpatrick et al., 1983), which are grounded in well-understood natural and physical processes, many new algorithms are inspired by obscure or loosely related metaphors. For example, algorithms based on the foraging behavior of bats (Yang, 2010) or the movement patterns of social animals often fail to establish a direct link between the metaphor and the algorithm's functional mechanisms. This lack of grounding makes it difficult to analyze and understand the algorithm's behavior and performance.

The field's excessive focus on metaphors has contributed to the acceptance and publication of work that lacks originality, despite the metaphorical appeal. For instance, Weyland (2010, 2015) demonstrates that the Harmony Search algorithm (Geem, 2001) is a special case of Evolutionary Strategies, which was introduced 35 years earlier in (Rechenberg, 1965). At the time of its publication, Google Scholar indicates that Harmony Search had approximately 500 citations, a number that has since grown to over 7000. A similar trend is observed with the Firefly Algorithm (Yang, 2008), the Bat Algorithm (Yang, 2010), the Grey Wolf Optimizer (Mirjalili, Mirjalili & Lewis, 2014), the Ant Lion Optimizer (Mirjalili, 2015), Moth-Flame Optimization (Mirjalili, 2015), and the Whale Optimization Algorithm (Mirjalili, 2016), which have been shown to be only marginally different from Particle Swarm Optimization (Camacho-Villalón, Dorigo & Stützle, 2023). This issue is also noted by (Tzanetos and Dounias, 2020), albeit without the rigorous mathematical analysis provided by the former study.

In conclusion, the lack of theoretical foundations in many metaphor-based metaheuristics represents an important challenge to the field. This deficiency obscures the underlying mechanisms of these algorithms and leads to the proliferation of superficially novel approaches that cause harm in the form of duplication and reduced cooperation. Addressing this issue requires a shift towards more rigorous theoretical and empirical validation, ensuring that new metaheuristic algorithms are both original and grounded in a solid theoretical framework. Only through this shift can the field establish a more cumulative, coherent, and scientifically credible body of knowledge.

1.3 Methodological shortcomings

Beyond the issue of weak theoretical foundations, metaheuristics research has also been hindered by suboptimal methodological practices that compromise the reliability and scientific value of published work. Chief among these concerns is the insufficient empirical validation of newly proposed algorithms. Many studies present novel metaheuristics without extensive testing, often relying on limited benchmark sets, shallow comparisons, or inadequate mathematical and/or statistical analysis. In some cases, flawed or outdated

evaluation techniques are used, leading to misleading conclusions about performance or generalizability (Gagnon, Abran & April, 2025).

1.3.1 Transparency and reproducibility

A closely related issue is the widespread lack of transparency and reproducibility in metaheuristics research. Descriptions of experimental setups, parameter tuning procedures, and algorithmic implementations are frequently vague or incomplete. As a result, replicating experiments or validating results becomes difficult, if not impossible. This issue is compounded by the common use of unpublished or proprietary code, which restricts the broader scientific community's ability to reproduce results, identify potential flaws, or build upon prior work.

The absence of reproducibility not only undermines the credibility of individual studies but also hampers the cumulative progress of the field. Without open and transparent research practices, meaningful comparison across algorithms is compromised, making it difficult to separate substantive advances from noise. Addressing this issue will require a cultural shift toward the adoption of shared benchmarks, open-source implementations, and detailed methodological reporting.

1.3.2 Experimental design

One of the earliest published critiques of evaluation practices in computational science is the seminal article titled "Testing Heuristics: We Have It All Wrong" (Hooker, 1995). The author challenges the then-dominant practice of evaluating metaheuristics based on best-found values, a method he calls "competitive testing". He argues that this approach is fundamentally unscientific because it fails to explain *why* one algorithm performs better than another. Moreover, it shifts focus away from controlled investigation and toward performance competition, which he deems anti-intellectual and ultimately counterproductive to genuine algorithmic insight.

Hooker's critique echoes the sentiments of earlier pioneers like (Hamming, 1962) who famously stated that "the purpose of computing is insight, not numbers," while (Geoffrion, 1976) applied a similar view to mathematical programming. All three emphasize that although numerical results are necessary, their ultimate value lies in the insights they yield about methods, structures, and underlying problems. In this view, computation and experimentation are not endpoints but tools for advancing understanding, decision-making, and effective problem-solving.

That same year, (Barr, Glover & Klingman, 1995) proposed a more structured methodology for evaluating metaheuristics. They advocated for rigorous, well-documented experimental procedures and outlined a framework still relevant today. Drawing parallels with best practices in design of experiments (Montgomery, 2021), they proposed five essential steps:

1. Define the goals of the experiment.
2. Select performance measures and experimental factors.
3. Design and execute the experiment.
4. Analyze the data and interpret the results.
5. Report the experiment's methods and results in full detail.

They further emphasized the importance of criteria beyond raw performance, such as simplicity, generalizability, robustness, and insight, as well as transparent reporting. Their perspective reinforces the idea that metaheuristics research should not only produce results but also deepen understanding.

(Gent et al., 1997) added a practical dimension with their technical report titled "How Not To Do It", which provided a detailed list of empirical best practices and common pitfalls. They emphasized reproducibility, systematic parameter tuning, robust statistical analysis, and careful experimental design. Their contribution remains a cornerstone for conducting meaningful empirical studies of algorithms.

Building on these efforts, (Rardin & Uzsoy, 2001) stressed the importance of deriving transferable insights from metaheuristics research. They promoted the use of exploratory

studies or pilot studies to identify non-influential factors, enabling more efficient and focused experimental designs. Their view reinforces a broader theme: the purpose of experimentation is not simply to compare, but to understand.

Despite these early and enduring calls for methodological rigor, many contemporary metaheuristics studies continue to deviate from these principles. It remains common to find evaluations based on single-run reporting, inadequate parameter tuning disclosure, or selective benchmark usage (Velasco et al., 2023). These practices hinder reproducibility and undermine scientific validity. These trends mirror concerns raised in broader meta-scientific discussions, such as those surrounding the replication crisis in psychology and biomedical research (Open Science Collaboration, 2015). In these fields, as in metaheuristics, calls for transparency, statistical robustness, and replicable experimentation have led to the rise of open science initiatives like the Open Science Framework (OSF).

To address these concerns, researchers are increasingly encouraged to adopt modern tools and platforms that facilitate reproducibility. Version control systems such as Git, executable documents like Jupyter Notebooks, and containerization technologies such as Docker allow for full transparency of code, data, and experimental environments. Embracing these tools can help metaheuristics research align with best practices in computational science and enhance the credibility and impact of future contributions.

1.3.3 Statistical methodology

Following early critiques of empirical practices in metaheuristics, several methodological studies have emerged advocating for more rigorous statistical foundations. For instance, (Johnson, 2002) and (Chiarandini, Paquette, Preuss & Ridge, 2005) emphasized the importance of structured experimental design and statistically grounded analysis. Yet, the persistence of basic debates, such as whether to rely on mean or best-obtained results for comparison (Birattari & Dorigo, 2007), suggests that widespread adoption of best practices remains limited.

In 2015, two notable collaborative efforts addressed the standardization of metaheuristics research. First, (Swan et al., 2015) brought together 26 researchers to propose methodological guidelines for both descriptive clarity and statistical evaluation. Among their recommendations is the use of statistical resampling methods, such as bootstrap procedures for estimating p-values, in addition to careful empirical procedure selection.

Later that year, (Kendall et al., 2015) extended this line of work by adapting the Good Laboratory Practice (GLP) framework to the context of optimization research, branding it as GLP4OPT (Good Laboratory Practices for Optimization). Their proposal is prescriptive in nature and covers experimental design, statistical methods, documentation standards, and reproducibility protocols. However, despite the breadth of their proposal, statistical methodology occupies less than 2% of the document, revealing its secondary status in the overall hierarchy of research concerns.

More recently, (Halim, Ismail & Das, 2021) provided an extensive overview of statistical techniques, particularly emphasizing null hypothesis significance testing (NHST). While their review includes over a dozen procedures and a visual flowchart for guidance, it stops short of prescribing specific tests. Crucially, the absence of statistical power analysis, confidence intervals, and effect size measures reflects a recurring pattern in the literature: the statistical component is often addressed but not deeply integrated.

A partial response to this gap is offered by (LaTorre et al., 2021), who provide detailed guidelines covering benchmarking, validation, algorithm design and tuning, and reporting. Their statistical methodology recommendations include:

- Use established statistical methods.
- Prefer Bayesian analysis over NHST.
- Apply proper control of error rates in multiple comparisons.
- Use visual tools to support algorithm comparison.

In their discussion, the authors clarify that Bayesian methods are not a replacement but a complement to NHST, especially when classical significance testing yields inconclusive

results. They advocate for the integration of *a priori* power analysis, hypothesis validation, and post hoc analyses, emphasizing that statistical tools should serve the broader aim of producing robust and interpretable results.

Overall, while literature reflects an awareness of statistical rigor, its practical implementation remains inconsistent. Methodological calls for power analysis, effect sizes, and transparent reporting have not yet achieved widespread integration into the evaluation pipelines of metaheuristics research.

1.4 Frameworks

In a general context, a framework is an abstract structure designed to support the execution of various tasks. They aim to enforce best practices and streamline workflows, thereby alleviating common difficulties encountered during the execution of complex procedures. A useful analogy is to view frameworks as structured templates or instruction sets designed to guide practitioners toward achieving high-quality, reproducible results.

When narrowed to the domain of software development, frameworks provide reusable, modular components that can be assembled and integrated into larger applications. These frameworks often include standardized libraries, tools, and conventions that simplify the development process. By offering pre-built functionality, they reduce boilerplate code and increase efficiency, while still leaving room for customization.

The conveniences offered by frameworks have led to widespread adoption across various software communities. For example, according to Stack Overflow's 2023 Developer Survey, Facebook’s React framework for web application design is used by approximately 40% of web developers, based on responses from more than 70,000 participants. This widespread adoption underscores the practical value of well-designed frameworks in accelerating development, fostering best practices, and enhancing collaboration.

In contrast to conventional software frameworks, frameworks used in metaheuristics research often take the form of methodological guidelines rather than specific software tools.

There are no universally accepted reference frameworks in this domain, but a variety of approaches proposed through individual research publications (Kendall et al., 2015) or books (McGeoch, 2012). However, some software tools, referred to as Metaheuristics Software Frameworks (MSFs) or Metaheuristics Optimization Frameworks (MOFs), have been developed to support the design, implementation, and evaluation of metaheuristic algorithms.

MSFs abstract the key components of metaheuristics research into discrete, modular elements, such as algorithms, operators, problem definitions, and analysis tools. For instance, the initial version of the C#-based HeuristicLab framework was organized around three primary object-oriented classes: Algorithm, Problem, and Solution (Wagner, 2009). The jMetal framework, implemented in Java, follows a similar model with its own class-based architecture (Durillo, Nebro & Alba, 2010). ParadisEO, a C++ framework tailored for evolutionary algorithms, uses specialized constructs such as Evolving Objects and Moving Objects (Durillo, Nebro & Alba, 2010), highlighting the diversity in architectural approaches taken across different platforms.

A 2011 review identified ten actively maintained MSFs, including HeuristicLab, jMetal, and ParadisEO (Parejo et al., 2011). More recently, a follow-up review reaffirmed these three while also referencing a lesser-known C++ framework, EMILI (Camacho-Villalón, Stützle & Dorigo, 2023). The authors noted, however, that EMILI appears to have limited uptake and maintenance—possibly due to its dependence on the domain expertise of one of its original developers, who also co-authored the review.

Both (Parejo et al., 2011) and (Camacho-Villalón, Stützle & Dorigo, 2023) identify limited built-in analysis capabilities as a key limitation of MSFs. In fact, four out of the ten (40%) frameworks reviewed by Parejo et al. lacked statistical analysis tools entirely, including ParadisEO and HeuristicLab. As of this writing, both still offer only elementary statistical functions. This is a notable shortcoming, given that many of the most persistent criticisms of metaheuristics research relate directly to methodological rigor.

Another frequently cited challenge is the high barrier to entry posed by the programming expertise required to effectively use most MSFs. According to (Parejo et al., 2011), 60% of the frameworks required proficiency in Java, 30% in C++, and 10% in C#. Although HeuristicLab is a GUI-based application that alleviates some of this burden, it presents a steep learning curve due to its complexity. Moreover, the skills required to use HeuristicLab are not easily transferable to other contexts, a drawback of its highly specialized, no-code approach.

A further limitation relates to language selection. As (Campelo & Aranha, 2021) observe, the metaheuristics community tends to prioritize application-driven research over foundational theory. In this context, prototyping languages, particularly Python, may be better suited for rapid development and iterative experimentation. Python also benefits from a rich ecosystem of scientific libraries and tools; many inherited from its widespread use in artificial intelligence. These are among the reasons that led to the release of jMetalPy, a Python-based adaptation of jMetal (Benítez-Hidalgo et al., 2019).

In summary, MSFs provide structured, reusable components that facilitate the development, implementation, and analysis of metaheuristic algorithms. These frameworks offer a modular approach that enhances code reusability, maintainability, and standardization, which are critical factors for advancing research in this domain. However, despite their utility, significant gaps remain in their built-in analysis capabilities and the extensive programming knowledge required for their effective use.

One of the most prominent limitations is the insufficient support for statistical analysis within existing MSFs. Robust statistical methods are essential for validating algorithm performance and ensuring reliable, reproducible results. In the absence of integrated statistical tools, researchers must often resort to external software, which adds complexity and introduces opportunities for inconsistency or error. Enhancing statistical capabilities within MSFs could simplify workflows and elevate the quality of empirical research.

Moreover, the prevailing preference within the metaheuristics community for application-oriented research over theoretical contributions highlights the practical value of prototyping

languages. Python stands out for its extensive ecosystem of scientific libraries, intuitive syntax, and widespread use in AI research. Its accessibility makes it an especially effective language for the rapid development and iterative refinement of metaheuristic algorithms.

The persistence of methodological gaps, despite the availability of both software frameworks and prescriptive guidelines, can largely be attributed to the disconnect between tools and processes. Bridging this gap remains a pressing challenge. Integrating methodological guidance directly into MSFs could foster a more cohesive and user-friendly environment, supporting more consistent adherence to best practices and thereby improving research quality and reproducibility.

The separation between MSFs and methodological guidelines creates a persistent gap between tools and processes. This disconnect may help explain the continued presence of methodological shortcomings in metaheuristics research, despite the availability of both high-quality software and robust guidelines (Bartz-Beielstein et al., 2020). This thesis addresses the gap through the MDAF, which unifies prescriptive and technical components. The MDAF consists of a structured set of methodological guidelines and an envisioned open-source software library. The guidelines are presented in this work, while the implementation of the software library is left for future research.

CHAPTER 2

OVERVIEW OF PUBLISHED ARTICLES

This chapter contextualizes each article in relation to the research objectives (RO) presented in the introduction and presents their methodological approaches, principal results and scholarly contributions. Finally, section synthesizes their cumulative insights.

Table **Erreur! Source du renvoi introuvable.** shows the contributions of the three published articles to the development and validation of the MDAF by mapping each of them to the research objectives.

Table 2.1 Mapping of published articles to research objectives

No.	Research Objective	Article 1	Article 2	Article 3
1	Diagnose deficiencies in current practice	✓	✓	✓
2	Specify the MDAF	✓	✓	✓
3	Empirically validate the MDAF through targeted studies	✓	✓	

2.1 Critical Analysis

The first article centers on the Bat Algorithm (BA), a widely cited and influential metaheuristic that exemplifies many of the challenges facing the field. By scrutinizing an algorithm that has received broad academic endorsement, the study aims to uncover systemic methodological issues in metaheuristics research.

2.1.1 Aim

This article presents a critical assessment of the BA, a popular nature-inspired metaheuristic, by analyzing its underlying structure and empirically testing the performance claims of general superiority made by (Zhang, 2010). As stated in the article, its purpose is to demonstrate that inadequate research methodology has led the research community to accept BA as an original contribution when it is in fact a hybrid variant of previously known algorithms, namely PSO and SA. Through a concrete case study, it remonstrates that this misperception stems from (1) the lack of structural analysis, (2) insufficiently rigorous benchmarking, and (3) the superficial hybridization of existing strategies obfuscated by metaphorical language.

It also demonstrates that widespread scholarly interest, as indicated by high publication and citation counts, does not preclude the presence of significant conceptual or methodological flaws. Thus, it serves not only as an isolated critique but as a reflection of deeper methodological flaws within the metaheuristics research community.

2.1.2 Methodology

The article's methodological approach is introduced in section 4 of the article, where a component-based mathematical analysis of the BA is conducted. This analysis yields the following hypotheses which are explored in subsequent sections:

1. The BA exhibits sensitivity to initial conditions, primarily due to its hill-climbing behavior in the early stages of the optimization process;
2. The pulse rate mechanism, as described using metaphorical terminology, performs no better than random sampling;
3. The loudness mechanism, similarly presented in metaphorical terms, is functionally equivalent to the probabilistic acceptance strategy found in SA.

Hypothesis no.1 is explored by experimenting with the two following initialization strategies:

1. Using a uniform random distribution over the entire domain (i.e., including the known global minimum);
2. Using a uniform random distribution over a restricted domain that does not contain the global minimum.

Hypothesis no.2 is explored by comparing the performance distributions of the two following versions of the BA:

1. The canonical BA with its original pulse rate mechanism;
2. A variant where the pulse rate mechanism is replaced by pure random sampling.

Hypothesis no.3 is explored by comparing the performance distributions of the two following versions of the BA:

1. The canonical BA with its original loudness mechanism;
2. A variant where the loudness mechanism is removed.

To ensure fair comparisons, algorithms were reimplemented in a unified public codebase (https://github.com/iangagn/ENG-2019-11-084) and evaluated under identical experimental conditions. The primary measure of performance was the number of function evaluations required to reach an ε-neighborhood ($\varepsilon = 10^{-2}$) of the known global optimum (section 5). Performance was assessed on five well-established De Jong benchmark functions, each selected to expose different optimization challenges: smooth convexity, narrow valleys, plateaus, ruggedness, and deceptive multimodality (section 5.1). Parameter settings were selected through a multidimensional grid search, using the median first-hitting time (FHT) as the optimization criterion and adjusting for hit rate to avoid bias from failed runs (section 5.2). Each algorithm-function pair was run independently $n = 10^3$ times to generate robust empirical distributions. The resulting distributions were analyzed using nonparametric statistical tests, namely the Kolmogorov-Smirnov test to assess distributional differences and the Wilcoxon-Mann-Whitney test to compare medians without assuming normality. Finally, study uses a graphical approach in section 6.4 as additional support to the exploration of hypothesis no.2.

2.1.3 Key results

The study confirmed hypothesis no.1 which proposed that the BA is highly sensitive to initial conditions compared to PSO, with statistically significant large increases in FHT when the initialization region did not include the global minimum. This limitation is particularly consequential in real-world optimization contexts, where the global minimum is typically unknown, thereby exposing a critical weakness in the algorithm's practical applicability.

The study also confirmed hypothesis no.2 which proposed that the pulse rate mechanism is not superior to random sampling by showing that performance *increases* significantly when it is replaced by the latter. This finding contradicts the notion that the BA consistently outperforms PSO and related algorithms and additionally highlights that the algorithm's metaphor-based components serve to obscure its limitations rather than contribute to functional improvement.

The study partially confirmed hypothesis no.3 which proposed that the loudness mechanism is functionally equivalent to SA's probabilistic acceptance mechanism by showing that removing it only exacerbates its underlying behaviour (i.e., the effects of its other components). Further studies were suggested to compare an optimally parameterized version of the BA without the loudness mechanism to an equally well parameterized version of canonical PSO.

2.1.4 Discussion

The results discussed in the previous section provide evidence against the notion that the BA is a generally superior alternative to PSO and other related metaheuristics. Moreover, it presents conclusive theoretical and empirical evidence that the BA is a hybrid between PSO and SA, which go back to 1995 and 1983 respectively, and that this is deliberately or inadvertently obscured by metaphorical language. A later study made by (Camacho-Villalón, Dorigo & Stützle, 2023) supports this exact conclusion based on a similar mathematical analysis. This aligns with the call to action made in (Piotrowski & Napiorkowski, 2018) to avoid the unnecessary complexification of metaheuristics by

carelessly adding progressively more components (e.g., the pulse rate and the loudness mechanisms). These results show that metaphorical language lacks explanatory power and contributes to the propagation of algorithms like the BA and Harmony Search (see Weyland, 2010) whose novelty is superficial based on unsubstantiated claims (RO1). Another noteworthy contribution was made by the study's mathematical analysis and subsequent demonstration of the swarm collapse behavior (sections 4 and 6.4) which explained the observations made in (Suárez et al, 2019) that BA-controlled robots would begin the exploration process by running into each other – a detail that was brought to the attention of the authors during the peer-review process. This exemplifies how rigorous structural analysis can clarify and contextualize even long-standing anomalies in applied studies (RO2 and RO3).

These results were enabled by the application of a rigorous methodological framework, which included the following key elements (RO2 and RO3):

1. A detailed mathematical analysis of the algorithm's components, leading to the development of explicit and testable hypotheses;
2. The selection of a machine-independent performance measure, specifically the FHT;
3. The implementation of fair comparison procedures through careful parameter tuning via grid search and bias mitigation strategies such as adjusting the median FHT by the convergence rate;
4. The use of large sample sizes and appropriate statistical practices, including distribution-aware measures of central tendency and nonparametric hypothesis testing;
5. The inclusion of supplementary visualizations when beneficial to interpretation.

In summary, this article critically re-evaluates both the purported novelty and the alleged performance superiority of the BA, while also exposing systemic methodological issues prevalent in metaheuristics research. By integrating formal mathematical analysis with methodical empirical validation, it reveals how metaphor-oriented approaches obscure the

absence of substantive contributions. The results affirm the imperative for more transparent reporting and methodologically sound evaluation practices moving forward.

2.2 An Investigation of the Effects of Chaotic Maps on the Performance of Metaheuristics

The second article investigates the integration of chaotic maps into metaheuristic algorithms, a practice frequently justified by the authors based on the widespread but unverified assumption, often explicit, that it generally enhances performance. By systematically and rigorously evaluating the performance of both swarm-based and single-state metaheuristics augmented with chaotic maps, the study aims to test whether these claims are made under controlled experimental conditions. Notably, it demonstrates that when the sequence of chaotic values is randomized, the observed performance differences vanish, thereby highlighting the role of sequence effects rather than the inherent benefits of chaos.

2.2.1 Aim

This article presents an empirical investigation into the performance effects of replacing standard pseudo-random number generators (PRNGs) with chaotic maps in two representative metaheuristics: PSO and SA. The study questions the methodological foundations of prior research in this domain, which frequently reports performance improvements from chaotic maps based on inadequate methodology. The study sets itself apart by employing statistical tests based on large sample sizes, conducting comparisons using empirically optimized parameter configurations, reporting effect sizes, and ensuring transparency through the public release of all data and source code.

Its main purpose is to assess whether chaotic maps produce consistent, statistically significant and practically meaningful improvements in performance, and whether any observed improvements can really be attributed to the statistical properties of chaotic maps themselves. In doing so, the study introduces the concept of sequence effects, or the notion that the specific order in which the values are generated is responsible for the performance differences, to suggest the possibility that structured sampling patterns influence outcomes

independently of their statistical distribution. By critically analyzing both the assumptions and the empirical practices surrounding chaotic maps, the article challenges the narrative that chaotic maps are inherently beneficial in metaheuristics research.

2.2.2 Methodology

The experimental design is detailed in section 3 of the article. Two algorithms, canonical PSO and SA, were modified to create chaotic variants (CPSO and CSA) by replacing the standard PRNGs with six well-known chaotic maps: Chebyshev, Circle, Gauss, Logistic, Sine, and Tent. Each modified algorithm was evaluated using the same benchmark functions as in the first article presented in section **Erreur! Source du renvoi introuvable.**, which represent a range of optimization challenges including convexity, multimodality, ruggedness, discontinuity, and noise.

To ensure fair comparisons, algorithms were implemented in a unified public codebase (https://github.com/iangagn/ENG-2019-12-0887). Parameters for PSO/CPSO and SA/CSA were optimized using a full-factorial grid search based on the rank-sum of median FHT across all test functions. The FHT measures the number of objective function evaluations required to reach an ε-neighborhood ($\varepsilon = 10^{-2}$) of the global optimum and was selected to provide a balanced measure of both quality and computational cost (section 4). Each experiment was replicated $n = 10^3$ times to generate robust empirical distributions.

Performance distributions were analysed using nonparametric statistical methods. The Wilcoxon Rank-Sum test was used to compare medians, while Levene's test verified homogeneity of variance. When significance was observed, effect sizes and confidence intervals were estimated using bootstrapping. The study also examined whether performance differences disappeared when chaotic sequences were randomly shuffled to isolate potential sequence effects using the same comparison methodology.

2.2.3 Key results

The study found that chaotic maps did not generally improve the performance of PSO. Statistically significant improvements were observed in only one case: PSO on a noisy convex function with the Chebyshev map. Subsequent testing revealed that this improvement disappeared when the sequence was randomly shuffled, supporting the presence of sequence effects. In four out of five test functions, the use of chaotic maps led to statistically significant *decreases* in performance, with FHT increases ranging from 58% to 444%.

For SA, no statistically significant performance differences were found between the canonical and chaotic variants across all benchmark functions. This result suggests that chaotic maps may have limited or context-dependent effects on single-state metaheuristics, with no statistically significant improvements observed under the specific conditions tested in this article.

2.2.4 Discussion

The results discussed in the previous section provide evidence against the alleged benefits of chaotic maps in metaheuristics research. While prior studies have often reported improvements, they typically lacked rigorous statistical methods, large sample sizes, adequately parameterized baseline algorithms, reporting of effect sizes, and/or transparency via source code availability (RO1). This article demonstrates that when these standards are upheld (RO2 and RO3), the performance effects of chaotic maps either disappear or reverse. The results suggest that improvements observed in earlier works may have been caused by sequence effects or biased experimental setups rather than genuine algorithmic advantages.

By introducing and testing the concept of sequence effects, the study makes a novel contribution to the understanding of the effects of randomness in metaheuristics. The disappearance of performance improvements upon shuffling chaotic sequences underscores the need to critically evaluate the structure, not just the distribution, of random number generators in algorithm design.

In summary, this article calls into question a decade-long trend in the metaheuristics literature by showing that chaotic maps do not lead to consistent or generalizable performance improvements. Instead, it advocates for more careful experimental design, transparent reporting, and methodological rigor. These results strengthen the need for a more disciplined empirical foundation in the study of metaheuristics.

2.3 A Quantitative Evaluation of Statistical Practices in Metaheuristics Research

The third article addresses widespread concerns about the lack of methodological rigor in metaheuristics research. While articles 1 and 2 focus on empirical evaluation of claims made about specific algorithms or their components, this article adopts a broader perspective by quantitatively evaluating the methodological practices of the metaheuristics research community with an emphasis on statistical practices.

2.3.1 Aim

This article investigates the statistical practices of the metaheuristics research community by systematically analyzing a random sample of 70 peer-reviewed articles published in the recent literature. The study is motivated by a recurring pattern of methodological concerns, namely the absence of key elements such as statistical hypothesis testing, effect size reporting, and confidence intervals, which are standard expectations in more mature scientific disciplines like medicine and psychology. Using a quantitative evaluation framework derived from best practices codified by the American Psychological Association (APA) and the American Medical Association (AMA), the study aims to determine the extent to which the metaheuristics research community adheres to these norms. It provides a structured baseline for identifying methodological gaps and justifies the need for methodological frameworks such as the MDAF.

2.3.2 Methodology

The study employs a random sampling approach, using a custom Python script to extract article metadata from Google Scholar. From the pool of results, 70 articles were randomly selected for evaluation. The analysis framework comprises two sets of criteria:

- **Category A**: descriptive statistical practices (e.g., central tendency, variability, sample size adequacy);
- **Category B**: inferential statistical practices (e.g., hypothesis testing, power analysis, effect size reporting, correction for multiple comparisons).

Each article was evaluated against a checklist of 20 criteria, resulting in a binary classification (met/not met) per item. The results were quantified as coverage rates and presented alongside their respective confidence intervals.

2.3.3 Key results

The study revealed pervasive deficiencies in the statistical practices employed within the metaheuristics research community. Notably, among the 70 randomly selected articles, not even one concurrently reported statistical significance, confidence intervals, and effect size, despite these being widely recognized as core components of rigorous inferential analysis. Individually, null hypothesis significance testing was used in only 43% of articles (95% CI [38.0, 48.0]), while confidence intervals and effect sizes were reported in just 1% (95% CI [0.9, 1.1]) and 3% (95% CI [2.6, 3.4]) of articles, respectively.

Another critical issue identified is the lack of code availability. Only 8% (95% CI [7.1, 8.9]) of the articles made their source code publicly accessible, with 92% (95% CI [81.2, 100.0]) failing to do so.

2.3.4 Discussion

These results underscore a critical disconnect between the methodological norms expected from a mature scientific field and the prevailing practices in metaheuristics research. While some progress has been made, such as the growing use of nonparametric tests, key elements of statistical rigor remain largely absent. The overreliance on descriptive statistics, the failure to report uncertainty through confidence intervals, and the omission of effect sizes significantly undermine the reliability and interpretability of published results (RO1). This is especially concerning in an applied field such as metaheuristics, where understanding the

magnitude and uncertainty of observed effects is essential for translating research into practice.

Notably, the study revealed that not a single article in the sample simultaneously reported the three core components of inferential analysis: statistical significance, confidence intervals, and effect size. This aligns with broader criticisms in the literature concerning the lack of statistical robustness underlying widely cited results. By quantifying these issues through a replicable, data-driven evaluation framework, the study substantiates the need for methodological reform and provides an empirical basis for the prescriptive components of the MDAF (RO2).

Importantly, the criteria evaluated in this article are neither complex nor resource-intensive. They are widely adopted in fields such as psychology, medicine, and pharmacology, which have successfully improved reporting standards and now demonstrate near-complete adherence to these basic methodological norms. This demonstrates that the metaheuristics research community is not constrained by technical barriers, but rather by a lack of consensus on best practices and insufficient editorial or institutional pressure to adopt them.

The absence of publicly available source code in 92% (95% CI [81.2, 100.0]) of the surveyed articles further highlights a major obstacle to replicability and long-term scientific progress. This severely limits the reproducibility of results and likely contributes to the rarity of replication studies in the field. As suggested in prior literature, this shortcoming could be effectively addressed by enforcing open science policies at the level of journals, institutions, and funding agencies.

The evaluation framework proposed in this article is intentionally designed to be lightweight, interpretable, and adaptable. It can be used not only by researchers to self-assess their methodological rigor but also by peer reviewers and editors as a standardized quality control tool. Its utility extends beyond metaheuristics research and could serve as a diagnostic tool in adjacent fields such as computer science, software engineering, and applied artificial intelligence.

In summary, this article contributes to the literature by systematically quantifying the methodological deficiencies that articles 1 and 2 demonstrate through focused case studies. The results confirm that the metaheuristics field has not yet adopted the baseline statistical practices now standard in more mature scientific fields. By providing a structured and actionable blueprint for reform, the study reinforces the urgency of adopting methodological frameworks like the MDAF to ensure the scientific validity of research.

2.4 Conclusion

This thesis addresses longstanding methodological deficiencies in the field of metaheuristics research by developing and empirically validating the MDAF. While metaheuristics are widely used in engineering, computer science, and artificial intelligence, their development and evaluation have often been undermined by inadequate methodology. This work contributes to the advancement of the field by proposing an integrated framework grounded rigor and transparency.

2.4.1 Contributions

The three published peer-reviewed articles included in this thesis converge to form a coherent body of evidence supporting the development and necessity of the MDAF. Each article addresses distinct but complementary aspects of the metaheuristics research landscape, collectively advancing the research objectives stated in section 0.

Article 1 (section CHAPTER 2) reveals that even highly cited algorithms such as the Bat Algorithm can achieve broad academic legitimacy despite resting on weak theoretical foundations and misleading empirical claims. Through mathematical decomposition and controlled experiments, the article illustrates how superficial metaphor-based innovations can obscure structural redundancy with existing algorithms. The case study underscores the critical importance of rigorous methodology and transparent reporting which are explicitly encoded in the MDAF.

Article 2 (section 2.2) extends this critique to the widespread but unsubstantiated practice of replacing standard pseudo-random number generators with chaotic maps. The study

shows that while such modifications are often assumed to improve exploration capabilities, any performance gains are inconsistent and rarely statistically or practically significant. The article's rigorous experimental design demonstrates how the MDAF's prescriptive guidelines can lead to more scientifically sound results.

Article 3 (section 2.3) generalizes the concerns raised in articles 1 and 2 by quantifying the methodological state of the field at large. By evaluating 70 randomly selected peer-reviewed studies, the article reveals widespread deficiencies in statistical reporting, effect size estimation, and code availability, among other methodological elements. These results validate the need for a framework like the MDAF.

Taken together, the three studies support the threefold purpose of this work:

- Diagnosing methodological deficiencies (RO1) through case studies and audits;
- Specifying the MDAF (RO2) by demonstrating what rigorous research looks like in practice;
- Empirically validating the framework (RO3) by showing that adherence to its principles leads to more reliable outcomes and enables more comprehensive analyses.

These results also offer broader implications for the field. First, they illustrate that the methodological weaknesses in metaheuristics research are not isolated to poorly cited or marginal work. Second, they emphasize the role of methodological rigor not merely as formalities but as essential safeguards against misleading conclusions and wasted research efforts. Finally, they demonstrate that reform is feasible: the standards promoted by the MDAF are attainable with current tools, commonly practiced in adjacent fields, and necessary for the continued maturation of metaheuristics as a credible scientific discipline.

In conclusion, the synthesis of results across the three articles affirms both the relevance and the applicability of the MDAF.

2.4.2 Future directions

Several avenues for future work emerge from this research. First, the MDAF could be extended to address methodological challenges specific to multi-objective optimization, constrained search, and real-time or online applications. Second, additional tools and checklists could be developed to support automated adherence to MDAF principles for self-audit or during peer review. Third, integrating the MDAF into educational curricula and peer-review guidelines could promote cultural change and help establish a new norm of empirical rigor in the field.

In closing, this thesis provides not only a critical evaluation of current practices in metaheuristics but also a constructive path forward. By promoting methodological discipline and transparency, the MDAF supports the long-term goal of establishing metaheuristics research as a credible scientific discipline.

CHAPTER 3

A CRITICAL ANALYSIS OF THE BAT ALGORITHM

Iannick Gagnon[a], Alain April[a] and Alain Abran

[a] Department of Software and IT Engineering, École de Technologie Supérieure,
1100 Rue Notre-Dame West, Montreal, Quebec, Canada, H3C1K3

Paper published in *Engineering Reports*, May 2020

Abstract

This article presents an analysis of the Bat Algorithm based on elementary mathematical analysis and statistical comparisons of the first hitting time performance metric distributions obtained on a test set comprising five carefully selected objective functions. The findings show that the Bat Algorithm is not an original contribution to the metaheuristics literature and that it is not generally superior to the Particle Swarm Optimization algorithm when fair comparisons are made. It is also shown that some components of the Bat Algorithm can be either replaced by simpler alternatives or be removed entirely to increase performance. Finally, the results suggest that the best version of the Bat Algorithm is in fact a simple hybrid between Particle Swarm Optimization and Simulated Annealing. To encourage more transparency in metaheuristics research, the entirety of the MATLAB code used in this article is available in a GitHub repository for suggestions and/or corrections.

Keywords

Metaheuristics, bat algorithm, particle swarm optimization

3.1 Introduction

Metaheuristics are universal stochastic optimization algorithms used to solve problems where objective functions are unknown or infeasible to solve with limited computational resources. Such problems include the Quadratic Assignment and Traveling Salesman

problems which are well known in the Operations Research community. More recent examples include the training of convolutional neural networks (Rere, Fanany & Arymurthy, 2015) and on-line load sharing management for web servers (Kalra & Singh, 2015) among others. This article compares and constrasts two specific metaheuristics, namely the Bat Algorithm (BA) (Yang, 2010) and Particle Swarm Optimization (PSO) (Eberhart & Kennedy, 1995).

The Bat Algorithm is proposed in (Yang, 2010) as a novel metaheuristic inspired by the echolocation mechanism of bats. The original article makes two main claims. The first is that BA is a novel metaheuristic and the second is that BA is generally much superior to Particle Swarm Optimization and Genetic Algorithms (GA) based on experiments with 10 benchmark functions. There is no a priori reason to question the bat metaphor given that other metaheuristics like ACO (Colorny, Dorigo & Maniezzo, 1992) really are inspired by biology (Dorigo & Stützle, 2004). The scientific literature on bat echolocation does not discredit the metaphor, but it shows that it is shallow. BA parameterizes true bat echolocation based on frequency ($\boldsymbol{f}$), pulse rate ($\boldsymbol{r}$) and loudness ($\boldsymbol{A}$) while real bats use the former plus duration, directionality and a complex motor control system to home in on prey (Jakobsen, Brinkløv & Surlykke, 2013). To be fair, the fact that BA is not a perfect model of true bat behaviour is not a problem per se, but if the research community accepts metaphorical descriptions, it should insist that other possible analogies be differentiated. Case in point, bats are not the only species that use echolocation for navigation and hunting. Some bird species employ a similar strategy and so do certain species of dolphins and whales. Any one of these animals could have been chosen to create the Whale Algorithm for echolocation-based optimization instead of BA. This creates unnecessary confusion since Whale Optimization Algorithm (WOA) (Mirjalili & Lewis, 2016) does exist and it is not based on echolocation. To repeat one of Kenneth Sörensen's points in (Sörensen, 2015), the language of metaphors obscures what is truly novel about an algorithm other than its name and says very little if anything about *why* it works. The reliance on metaphors instead of mathematical descriptions is but one of the methodological flaws commonly observed in the metaheuristics literature.

The purpose of this article is to show that inadequate research methodology has led the research community to believe that BA is an original contribution to the metaheuristics literature. The research hypothesis is that BA is an underperforming variant of PSO instead of the other way around. The reason for singling out BA over other metaheuristics is that BA has obtained more than 10,000 citations since it was published in 2010 as is reported in Web of Science as of November 2019. This makes BA a good candidate for demonstrating that the growing concern that bad methodology is widespread in the metaheuristics research literature (Sörensen, 2015; Barr, Kelly, Resende & Stewart, 1995; Boussaïd, Lepagnot & Siarry, 2013; Hooker, 1995; McGeoch, 1996; Brownlee, 2007) is in fact supported by evidence obtained by following good methodological practices.

The rest of this article is organized as follows: sections 3.2 and 3.3 describe PSO and BA respectively, section 3.4 contains a detailed mathematical analysis of BA, section 3.5 contains a detailed description of the methodology used to generate and analyze the results of benchmark function minimization discussed in section 3.6 and finally section 3.7 contains a summary of the findings and proposes future research avenues.

3.2 The particle swarm optimization metaheuristic

In historical terms, PSO is the third swarm intelligence metaheuristic after Stochastic Diffusion Search (SDS) (Bishop, 1989) and ACO (Colorni, Dorigo & Maniezzo, 1992). It is relevant to our discussion to point out that it precedes BA by a decade and a half. PSO was designed by social psychologist James Kennedy and electrical engineer Russell Eberhart (Eberhart & Kennedy, 1995). The initial purpose was to simulate the social behavior of humans through the imagery of the movements of bird flocks and fish schools. It was quickly realized that the resulting paradigm was well-suited for mathematical optimization and thereafter was used to optimize highly non-convex functions and neural network training.

Kennedy and Eberhart did not use names like Bird Flock Optimization and Fish School Optimization precisely because they systematically simplified PSO by removing some of the original nature-inspired components of the algorithm and found that it worked just as well.

As such, they decided to use the neutral imagery of particles as defined in (Millonas, 1993) and (Reeves, 1983). Others like (Duman, Uysal & Alkaya, 2012) with Migrating Birds Optimization (MBO) and (Filho, Neto, Lins, Nascimento & Lima, 2008) with Fish School Search (FSS) were apparently not as concerned with the impact of using metaphors.

The characteristic equations of PSO are given in Algorithm 3.1. The position ($\vec{x}_i^t$) at time step t of the i-th particle are its coordinates in the search space and the velocities ($\vec{v}_i^t$) are a weighed sum of the relative position vectors between the particles, their personal historical best positions ($\vec{p}_i^t$) and the current global best position ($\vec{x}^*$). The velocity term is weighed by what is called the inertia factor (ω). The effect of inertia is to accumulate velocity in a "good" direction over time. The primary difference with BA is the use of the i-th particle's personal best-known location. The parameters c_1 and c_2 are constants that weigh the relative importance between the global best and the personal best positions, respectively. The parameters r_1 and r_2 are random variables drawn from a uniform distribution generally with support $[0, 1]$ represented by $U(0,1)$. The randomness introduced by these parameters causes the particles to overshoot global best and personal best positions with proportions $1/c_1$ and $1/c_2$ respectively when t is large. The net effect is to prevent premature convergence by providing what Kennedy and Eberhart called a stochastic "kick" in (Eberhart & Kennedy, 1995). The key idea behind PSO is that at every time step, the particle is pushed – accelerated in PSO terminology – in the direction of a weighted average of the particle's personal best-known location and the entire swarm's best-known position. The implied assumption is that regions of high fitness are close to other regions of high fitness (Clerc, 2015).

Algorithm 3.1 Pseudocode for minimization with PSO

Global minimization procedure with PSO

Input: The maximum number of iterations t_{max}, The number of particles n, and PSO's coefficients ω, c_1 and c_2.

Output: An array corresponding to the best-found position $\vec{x}^*$.

initialize $\vec{x}$, $\vec{v}$, $\vec{x}^*$ and $\vec{p}$ with zeros
while $t < t_{max}$ **do**
  for $i = 1 : n$
    $\vec{v}_i^{t+1} \leftarrow \omega\vec{v}_i^t + c_1 r_1(\vec{x}^* - \vec{x}_i^t) + c_2 r_2(\vec{p}_i - \vec{x}_i^t)$
    $\vec{x}_i^{t+1} \leftarrow \vec{x}_i^t + \vec{v}_i^{t+1}$
    if $\vec{x}_i^{t+1} < \vec{x}^*$ **then**
      $\vec{x}^* \leftarrow \vec{x}_i^{t+1}$
    end if
    if $\vec{x}_i^{t+1} < \vec{p}_i$ **then**
      $\vec{p}_i \leftarrow \vec{x}_i^{t+1}$
    end if
end for
  end while
  return $\vec{x}^*$

There are hundreds of variants of PSO some of which were proposed by the original authors. The original PSO given in (Yang, 2010) is the one used in this article despite there being a standard reference PSO algorithm proposed in (Clerc, 2012). The reason for this choice is that the authors believe that BA is only marginally different from the original PSO, therefore Original PSO is a more appropriate basis of comparison.

3.3 The bat algorithm metaheuristic

BA is also a population metaheuristic. It works on several solutions called bats at every time step $\boldsymbol{t}$. Each of the $\boldsymbol{n}$ bats is represented by a position in parameter space denoted by the vector $\vec{\boldsymbol{x}}_{\boldsymbol{i}}^{\boldsymbol{t}} = \langle \boldsymbol{x}_{\boldsymbol{i},\boldsymbol{1}}^{\boldsymbol{t}},\ \boldsymbol{x}_{\boldsymbol{i},\boldsymbol{2}}^{\boldsymbol{t}}, \dots, \boldsymbol{x}_{\boldsymbol{i},\boldsymbol{d}}^{\boldsymbol{t}} \rangle$ just like PSO. The characteristic equations of BA are the following:

$$f_i = f_{min} + \beta(f_{max} - f_{min}) \tag{3.1}$$

$$\vec{v}_i^{t+1} = \vec{v}_i^t + f_i(\vec{x}^* - \vec{x}_i^t) \tag{3.2}$$

$$\vec{x}_i^{t+1} = \vec{x}_i^t + \vec{v}_i^{t+1} \tag{3.3}$$

The frequency of the sonar f is bounded by f_{min} and f_{max} which are generally set to 0 and 1 respectively. The parameter β is a random number drawn from a uniform distribution with support $[0, 1]$. The position is updated at each iteration by moving it by an amount $\vec{v}_i^{t+1}$ called velocity. Thus far, there is no practical difference with PSO except that the personal best position is omitted. This is not insignificant, because the authors of PSO explicitly reported in (Eberhart & Kennedy, 1995) that this component increased performance. It is therefore implicitly stated that the new mechanisms proposed in Algorithm 3.2 more than make up for this exclusion. Three other parameters are introduced in BA: pulse rate r, loudness A and step size δ. This last parameter controls the size of the random step taken around the global best position when the conditional statement $U(0,1) > r_i$ is true.

Pulse rate and loudness are used as parameters for the two probabilistic acceptance mechanisms of lines 8 and 11. This idea is borrowed from SA (Kirkpatrick, Gelatt & Vecchi, 1983) which introduced the notion that non-improving moves should be accepted following some probabilistic mechanism so that the algorithm can escape local minima and go on to search for other promising regions of the fitness landscape. While SA uses what is called a cooling schedule to gradually decrease the probability of accepting non-improving moves, BA uses pulse rate and loudness. Pulse rate (Equation 3.4) is a function of the initial value r_0 and γ which controls speed of convergence as shown in figure (1). A value of $r_0 = 1$ was used for convenience since $r^{t+1} \longrightarrow r_0$ as $t \longrightarrow \infty$.

$$r^{t+1} = r_0[1 - e^{-\gamma t}] \tag{3.4}$$

Algorithm 3.2 Pseudocode for minimization with BA

Global minimization procedure with BA

Input: The maximum number of iterations t_{max}, The number of particles / bats n, BA's coefficients β, γ, α, and δ, and a function to optimize f.

Output: An array corresponding to the best-found position $\vec{x}^*$.

initialize $\vec{\boldsymbol{x}}$, $\vec{\boldsymbol{v}}$, $\vec{\boldsymbol{x}}^*$, $\vec{\boldsymbol{r}}$ and $\vec{\boldsymbol{A}}$
while $t < t_{max}$ **do**
  $f = f_{min} + \beta(f_{max} - f_{min})$
  $\vec{v}^{t+1} \leftarrow \vec{v}^t + f(\vec{x}^* - \vec{x}^t)$
  $\vec{x}_{temp} \leftarrow \vec{x}^t + \vec{v}^{t+1}$
  for $i = 1 : n$
    if $U(0,1) > r_i^t$ **then**
      $\vec{x}_i^{t+1} \leftarrow \vec{x}^* + \delta \cdot \text{Normal}(0,1)$
    end if
    if $U(0,1) < A_i^t$ **and** $f(\vec{x}_{temp,i}) < f(\vec{x}^*)$ **then**
      $\vec{x}^* \leftarrow \vec{x}_{temp,i}$
      $\vec{x}^{t+1} \leftarrow \vec{x}_{temp}$
      $r_i^{t+1} \leftarrow r_0[1 - e^{-\gamma t}]$
      $A_i^{t+1} \leftarrow \alpha A_i^t$
    end if
end for
  end while
  return $\vec{x}^*$

Loudness is calculated at each time step using the following equation:

$$A^{t+1} = \alpha A^t \tag{3.5}$$

Equation 3.5 introduces the parameter α which controls the rate of exponential decrease as shown in Figure 3.2. Possible values of A are bounded by $[0, 1]$ and α is bounded by $[0, 1)$. Since $t \in \mathbb{Z}^+$, the continuous curves in Figure 3.1 and Figure 3.2 are shown for illustrative purposes.

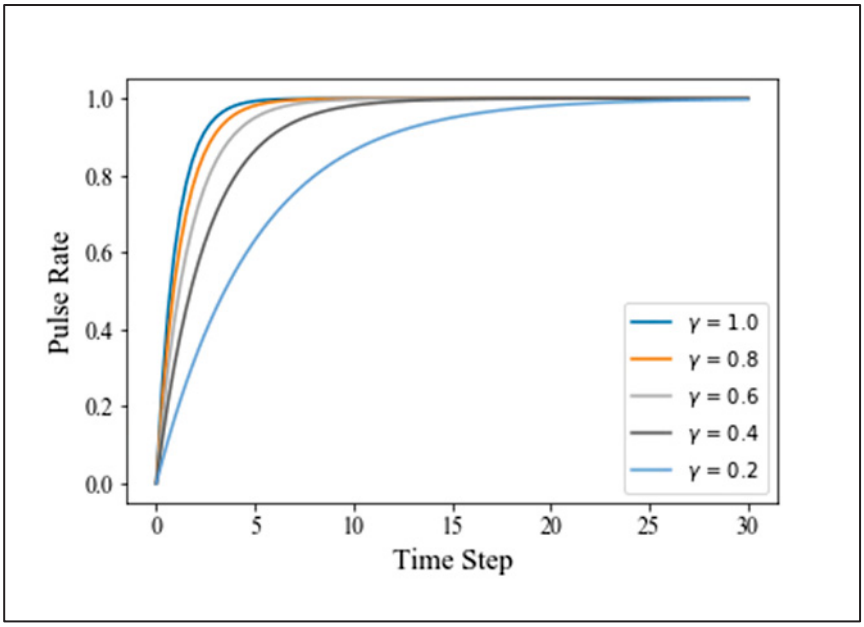

Figure 3.1 Effect of γ on pulse rate

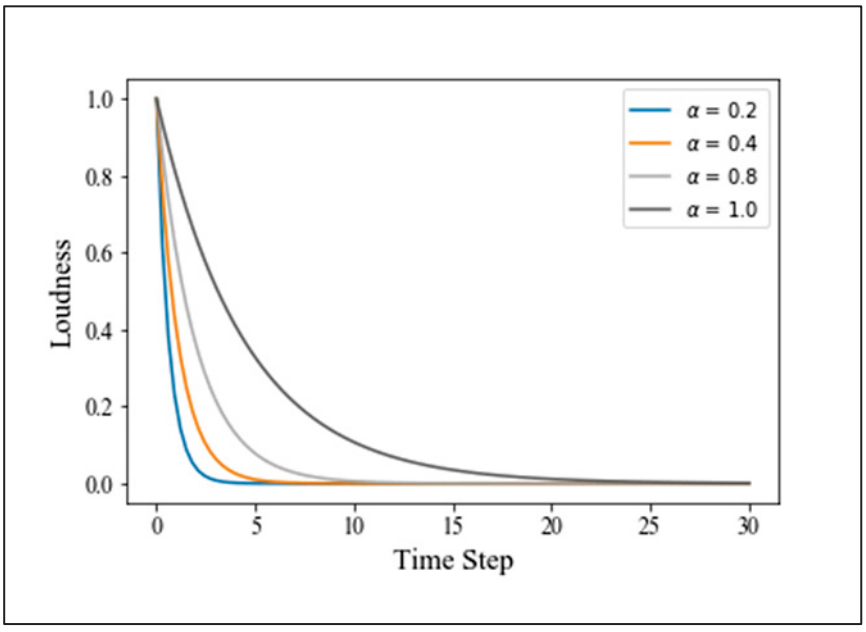

Figure 3.2 Effect of α on loudness

3.4 Comparative analysis

In (Yang, 2010), PSO is said to be a particular case of BA when the effects of pulse rate and loudness are removed. This is incorrect since Original BA does not use PSO's so-called autobiographical memory (Eberhart & Kennedy, 1995) (i.e., $\boldsymbol{c_2}$) to track each particle's personal best position and it also does not use inertia. To better understand the differences between BA and PSO, it is helpful to do a few elementary algebraic manipulations. The first component is represented by lines 4 to 6 of Algorithm 3.2.

Equation 3.1 can be rewritten as:

$$f_i = U(f_{min}, f_{max}) \tag{3.6}$$

Combining equations 3.1 and 3.2 gives:

$$\vec{v}_i^{t+1} = \vec{v}_i^t + U(f_{min}, f_{max}) \cdot (\vec{x}^* - \vec{x}_i^t) \tag{3.7}$$

Combining equations 3.3 and 3.7 gives:

$$\vec{x}_i^{t+1} = \vec{x}_i^t + \vec{v}_i^t + U(f_{min}, f_{max}) \cdot (\vec{x}^* - \vec{x}_i^t) \tag{3.8}$$

The same logic applied to lines 4 and 5 of PSO in Algorithm 3.1 gives:

$$\vec{x}_i^{t+1} = \vec{x}_i^t + \omega\vec{v}_i^t + U(0, c_1)(\vec{x}^* - \vec{x}_i^t) + U(0, c_2)(\vec{p}_i - \vec{x}_i^t) \tag{3.9}$$

From equations 3.8 and 3.9, the first component of BA is shown to be a special case of its counterpart in PSO when $\omega = 1$, $c_2 = 0$ and c_1 is drawn from the following probability density function (PDF) derived from the ratio of two uniform distributions:

$$f_X(z) = \begin{cases} \frac{1}{2} & 0 < z \leq 1 \\ \frac{1}{2z^2} & z > 1 \end{cases} \quad \text{with} \quad z = \frac{U(f_{min}, f_{max})}{U(0,1)} \tag{3.10}$$

The effect is that when an agent finds a better solution than the current global best, there is a one hundred percent probability that it will be accepted until t reaches a value of t_* when it starts to decrease exponentially until it is practically zero. As pointed out in (Yang, 2010), α plays a similar role as the cooling factor in SA (Kirkpatrick, Gelatt & Vecchi, 1982). The resemblance becomes clear when it is realized that α^t is equivalent to be^{ct} for values of $\alpha = e^{[\ln(b)-ct\,]/t}$ which is a form of SA cooling schedule (Nourani & Andresen, 1998).

The resulting PDF is shown in Figure 3.3 for $f_{min} = 0$ and $f_{max} = 1$. The area in blue covers $\int_0^1 1/2 \; dz = 1/2$ and the orange area covers $\int_1^\infty 1/(2z^2)\, dz = \; 1/2$ so that $\int_0^\infty f_X(x)\, dx = 1$. This shows that BA uses a random number generated from the same uniform distribution as PSO about half of the time and an exponentially decreasing function for the rest. In proportional terms, BA takes larger random steps in the direction of the global best position than PSO about three quarters of the time.

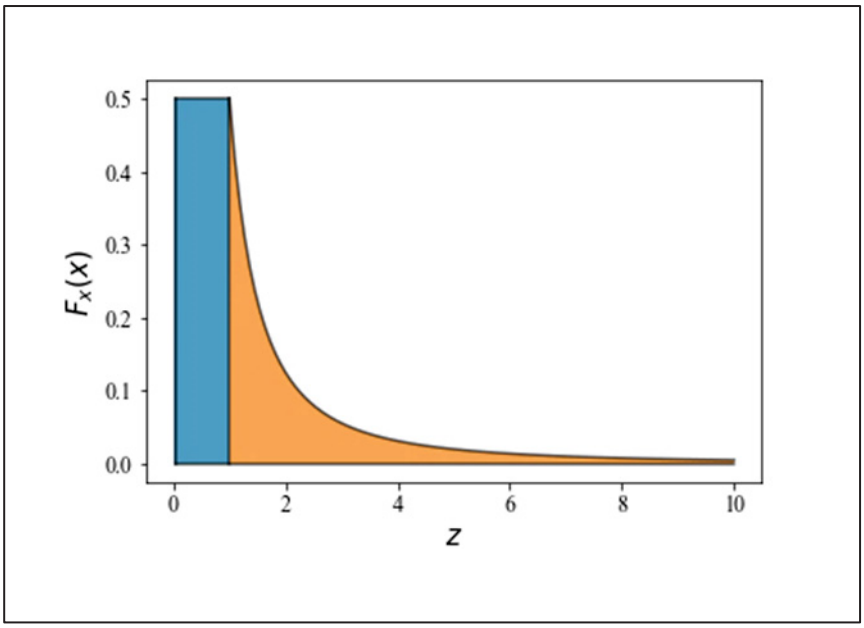

Figure 3.3 PDF of c_1 when $f_{min} = 0$ and $f_{max} = 1$ to match BA

The previous conclusion holds for $f_{min} = 0$ and $f_{max} = 1$, but it can be generalized. Given $f_{min} = 0$ and $f_{max} \geq 1$, BA will, on average, take larger steps than PSO in the direction of the global best position a fraction of the time equal to $[1 - 1/(4f_{max})]$. This also means that

BA overshoots the global best position by larger amounts than PSO. More precisely, the resulting long-tailed distribution causes the overshoot to sometimes be a large multiple of the distance between an agent's current position and the global best position. For example, if the distance $\|\vec{x}^* - \vec{p}_i\|$ is 1 unit, then the resulting push in the direction of the global best will be larger than 5 units 10% of the time, i.e. $\int_5^\infty 1/(2z^2)\ dz = 1/10$, causing an overshoot of 4 units. In practice, it is similar to increasing c_1 and ω in PSO, so the difference is likely to be of no consequence due to the effects of parameter selection.

Component 2 covers lines 8 to 10 of Algorithm 3.2 reproduced in Algorithm 3.3:

Algorithm 3.3 Pseudocode component 2 of BA

Partial global minimization procedure with BA

```
 8    if U(0,1) > r_i^t then
 9        x_i^{t+1} ← x* + δ · Normal(0,1)
10    end if
```

Equation 3.4 and Figure 3.1 show that the pulse rate is an exponentially increasing function whose rate is governed by the parameter γ. Technically, since $t \in \mathbb{Z}^+$, pulse rate goes from 0 to r_0 geometrically fast instead of exponentially, but the distinction is abandoned hereafter. The main effect of the second component is that it separates BA into two phases. The first phase is when the algorithm starts. When t is small, the second component is almost surely always applied since r begins at zero and $U(0,1) > r_i^t$ is likely to be true for most agents. This means that the second component makes the swarm converge on the initial global best value x_{init}^* and makes every agent take random steps around it in the beginning. It was pointed out during the peer-review process that this behaviour was observed in (Suárez, Iglesias & Gálvez, 2019) with BA-controlled robots. This makes the values of almost every agent's personal best $\vec{p}_i$ equal to the global best $\vec{x}^*$ in the first few iterations. This is similar to stochastic hill climbing because the swarm is centred around a single agent and is effectively sampling its immediate neighbourhood for improving positions. As such, the next following position has to be in its immediate neighbourhood. The direct consequence of this

finding is that BA is expected to perform relatively well on convex objective functions and less well on multimodal ones because "bad" initial positions will cause BA to converge to local extrema from the beginning.

The second phase of the BA optimization process is when t becomes larger. If r is equal to r_0 when it reaches β percent of its final value, then at $t = \lceil -\ln(1-\beta)/\gamma \rceil$, the probability that the second component is applied is simply $1 - r_0$. For example, if $\beta = 99\%$ and $\gamma = 0.5$, then it would be true starting at $t = 10$. Typical values of γ range from 0.5 to 0.9 (Xue, Cai, Cao, Cui & Li, 2015), so the preceding result could be as low as $t = 6$. This is a small value considering that the number of epochs is commonly observed in the 10^3 to 10^4 range. Larger values of r_0 cause the second component to be applied less often.

The third component covers lines 11 to 16 of Algorithm 3.2 reproduced in Algorithm 3.4:

Algorithm 3.4 Pseudocode component 3 of BA

Partial global minimization procedure with BA

```
11    if U(0,1) < A_i^t and f(x⃗_temp,i) < f(x⃗*) then
12       x⃗* ← x⃗_temp,i
13       x⃗^(t+1) ← x⃗_temp
14       r_i^(t+1) ← r_0[1 − e^(−γt)]
15       A_i^(t+1) ← αA_i^t
16    end if
```

Loudness is governed by the exponentially decreasing function given in Equation 3.5. The initial loudness for every agent is the same and is represented by A^0 without the i subscript. The probability that the loudness condition is true is given by Equation 3.11.

$$P(U(0,1) < A_i^t) = \begin{cases} 1, & 0 \le t \le t_* \\ \alpha^t A_i^t, & t > t_* \end{cases} \quad \text{with} \quad t_* = \lceil -\ln(A^0)/\ln(\alpha) \rceil \tag{3.11}$$

To summarize, it is clear from Equation 3.9 and the preceding analysis that BA does not meet the traditional criteria for being considered an original contribution. The so-called

traditional criteria refer to the ones given by the Canadian Intellectual Property Office (CIPO), the European Patent Convention (EPC) and the United States Patent and Trademark Office (USPTO) for patentability. BA uses mechanisms that (1) have been discovered before, (2) can be simplified and (3) can be considered obvious by most individuals with an average understanding of stochastic optimization algorithms. It is believed that this is the fruit of inadequate methodological standards and that the use of metaphors obscures these findings which would have been obvious otherwise. To support these conclusions, the following items will be investigated experimentally:

1. It is hypothesized that BA is particularly sensitive to the initial positions of the swarm. It appears that performance would decrease significantly if the swarm was initialized in a region that did not contain the global minimum.
2. It appears likely that the pulse rate mechanism (line 8, Algorithm 3.4) is no better than roulette wheel selection since it is asymptotically equivalent.
3. It appears likely that that component 3 of BA (Algorithm 3.5) is no better than the probabilistic acceptance mechanism of Simulated Annealing.

3.5 Methodology

This section describes the methodological approach used in this research. The goal is to provide all of the essential information to make it reproducible and easy to understand. This is accomplished by clearly stating the research assumptions. In this subsection, the following elements will be discussed briefly:

1. The first hitting time performance metric.
2. The notion of ε-neighbourhood.

First, the performance metric used in this research is the FHT which is defined in statistics as the first time a process reaches a specific subset of the state space. In this research, it is defined as the number of objective function evaluations (i.e., a measure of time) required to reach an ε-neighbourhood (i.e., a subset of the state space) of the known best solution of a benchmark function. Convergence is an interesting property, but it is often of no practical

importance if the swarm converged to the global minimum or not provided the best-found solution is kept in memory throughout the optimization process.

Second, the notion of "hit" is clarified by defining what an ε-neighbourhood is. An algorithm is considered to have hit target when it finds a position whose fitness is within $\varepsilon = 10^{-2}$ or less of the global minimum for a given benchmark. It is assumed that there is no need to make a distinction between 10^{-2} and 10^{-3} or 10^{-30} because the basin of attraction of the global minimum has almost certainly been found with the former. Some judgement is required.

Finally, section 3.5.1 contains details about the test set selection process and supports the use of a small number of test functions by emphasizing that selection should be based on problem characteristics rather than the size of the test set. The results obtained on the test set will be compared to the results on two more benchmarks to demonstrate the strength of this approach for future reference. Section 3.5.2 contains information about the parameter selection process by pointing to the appropriate literature and by describing the grid search procedure employed to find reasonable parameters for fair comparison. Section 3.5.3 explains how the research hypotheses stated in section 3.4 will be explored and describes the nonparametric statistical tests used to compare FHT distributions. The data that support the findings of this study are openly available in https://github.com/iangagn/ENG-2019-11-084.

3.5.1 Test set

The standard practice for the investigation and comparison of metaheuristics is to compare their performances on a series of benchmark problems. This is partly due of the No Free Lunch Theorem (NFLT) (Wolpert & MacReady, 1997) which says that the ideal universal optimizer does not exist and that good performance on a set of problems comes at the detriment of others. The result is that researchers often use large sets of test functions in order to find what problems the studied algorithm is good for. This approach is incomplete and redundant for the following reasons:

1. It does not attempt to explain why the algorithms perform the way they do on specific problems or problem attributes, so it largely fails to provide generalizable knowledge.
2. Choosing benchmarks with no reason other than to reach a predetermined number almost guarantees that some functions will provide redundant information already provided by other functions of the set.

To avoid these problems, this research aims to characterize the performance of the studied algorithms when facing the common difficulties encountered in optimization problems and to define a performance profile for each. This is achieved by making use of the well-known De Jong test functions (De Jong, 1975). This choice is motivated by the following items:

1. The small size of the test set (i.e., 5 functions) makes the results more tractable.
2. It was designed specifically to measure the strengths and weaknesses of optimization algorithms when confronted with specific difficulties such as ridges, discontinuities, noise and multiple peaks.

This decision is also supported by the fact that the average empirical linear correlation coefficient (ρ) as defined in (Maulana, 2018) for the five test functions is only 0.258 (Figure 3.4). An instance of two variables that have a correlation coefficient of 0.25 is shown in Figure 3.5 for reference. Functions f_1 and f_4 are highly correlated ($\rho = 0.91$), but this is explained by the fact that they both have bowl-like shapes and that their global minima coordinates are the same (i.e., the origin). The empirical linear correlation coefficient matrix is an imperfect measure of test set diversity, but it confirms that the functions "look" different without having to manually inspect every pair of graphs.

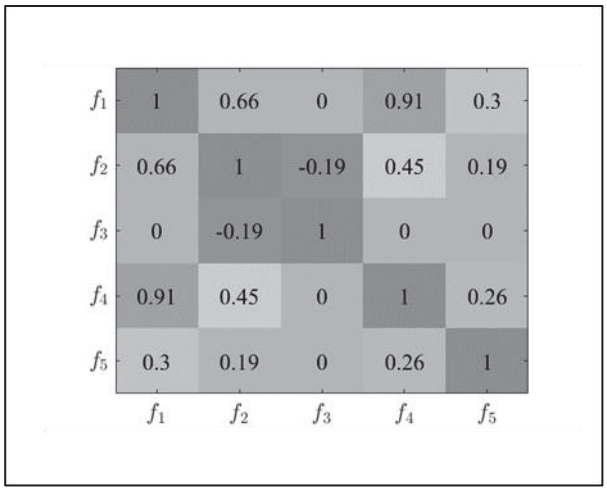

Figure 3.4 Empirical correlation matrix of the five benchmark functions

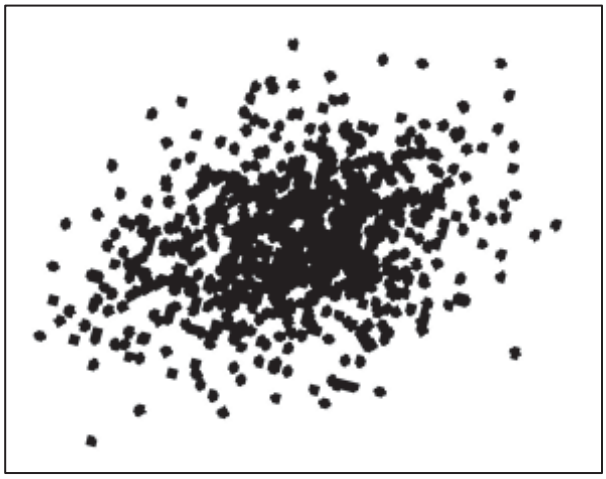

Figure 3.5 Dataset with a correlation coefficient of 0.25

The first benchmark is the unimodal convex function called Sphere (Figure 3.6). It is the least challenging an optimizer can face and simply serves to measure its general efficiency. Trajectory methods (i.e., algorithms that only use one solution per time step) always perform well on this kind of function, because the gradient always points in the direction of the global extremum (i.e. there are no local extrema).

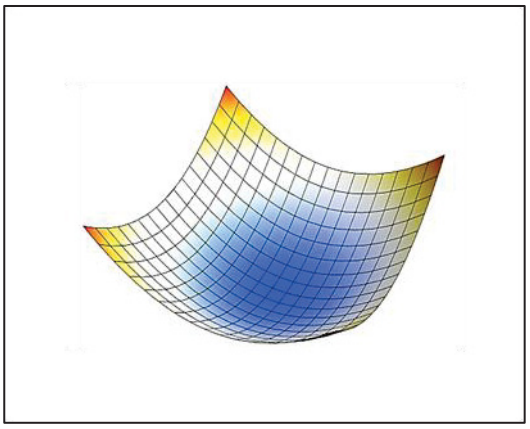

Figure 3.6 Sphere function

$$f_1(\vec{x}) = \sum_{i=1}^{n} x_i^2$$
$$-5.12 \leq x_i \leq 5.12 \tag{3.12}$$
$$\min(f_1) = f_1(0, \dots, 0) = 0$$

The second function is a generalization of the Rosenbrock function (Figure 3.7). The minimum is located inside a banana-shaped valley where optimizers can get stuck due to the scarcity of improving positions (i.e., low gradient).

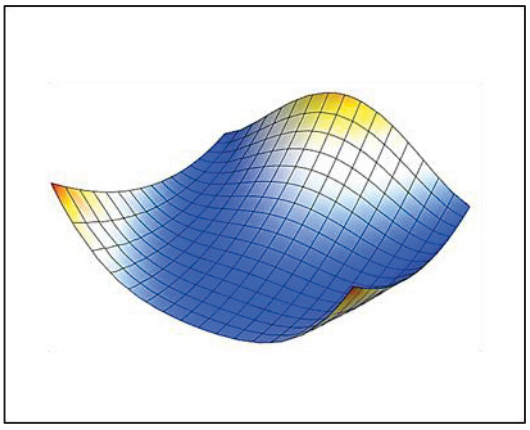

Figure 3.7 Rosenbrock function

$$f_2(\vec{x}) = \sum_{i=1}^{n-1} [100 \cdot (x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$
$$-5.12 \leq x_i \leq 5.12 \tag{3.13}$$
$$\min(f_2) = f_2(1, \dots, 1) = 0$$

The third function is called Step (Figure 3.8) and the main difficulty it poses is that it is made up of many plateaus (i.e., where $\nabla f = 0$). Optimizers fail when their neighbourhood topologies are fixed and/or are too small to search outside the plateaus where they are trapped.

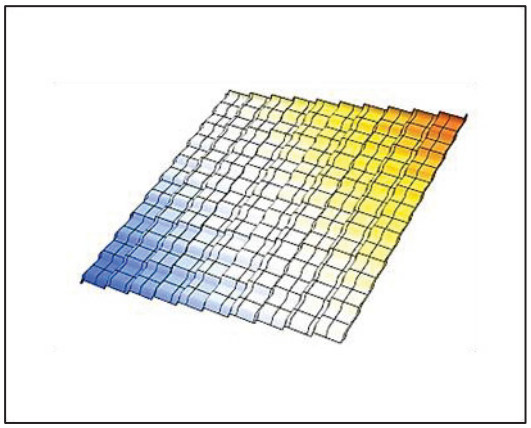

Figure 3.8 Step function

$$f_3(\vec{x}) = 6n + \sum_{i=1}^{n} \lfloor x_i \rfloor$$
$$-5.12 \leq x_i \leq 5.12 \tag{3.14}$$
$$min(f_3) = f_3\big([-5.12, -5), \dots, [-5.12, -5)\big) = 0$$

The fourth function is a quartic function with Gaussian noise (Figure 3.9) represented by $N(0,1)$. Noisy functions have the property of being rugged. Optimizers can fail when local minima appear to exist where they do not. Note that Figure 3.9 shows an inverted (i.e., multiplied by -1) version of the noisy quartic function for visualization purposes.

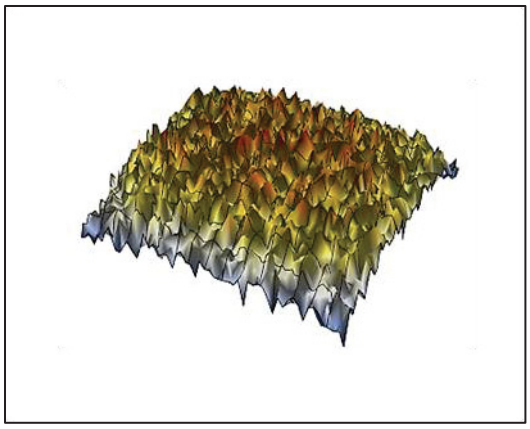

Figure 3.9 Noisy quartic

$$f_4(\vec{x}) = \sum_{i=1}^{n} (ix_i^4 + N(0,1)) \qquad (3.15)$$

$$-1.28 \leq x_i \leq 1.28$$

$$\min(f_4) = f_4(0, \ldots, 0) = 0$$

The fifth function is called Shekel's Foxholes (Figure 3.10). It is a plateau filled with steep basins where optimizers often cannot escape. Note that the figure is inverted for visualization purposes.

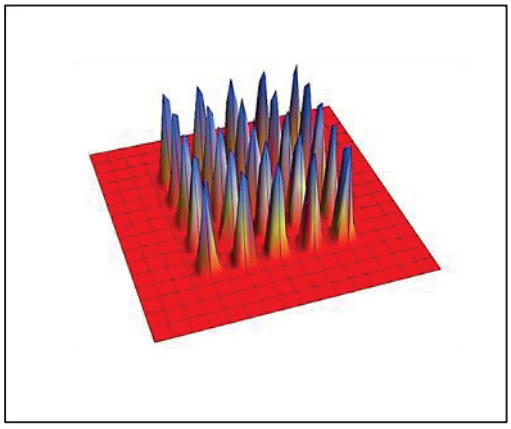

Figure 3.10 Shekel's foxholes

$$f_5(\vec{x}) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6} \right)^{-1}$$

$$\boldsymbol{a} = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & \dots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & \dots & 32 & 32 & 32 \end{bmatrix} \tag{3.16}$$

$$-65.536 \leq x_i \leq 65.536$$

$$\min(f_5) = f_5(-32, \dots, -32) = 0.9980$$

It is important to note that the domains of equations 3.12 to 3.16 do not have any special meaning and that they could just as well all be $[-100, 100]$. The values were taken directly from (De Jong, 1975). These functions are found in the referenced GitHub repository under the names sphere.m, rosenbrock.m, linear_step.m, noisy_quartic.m and foxholes.m.

3.5.2 Parameter selection

In the current metaheuristics literature, the parameter selection process is often left unmentioned which can leave the impression that the values were either chosen at random or carefully selected to produce a desired outcome. The purpose of this section is to explain how the parameters were chosen for BA and PSO. Since this research is mainly concerned with exploring the claim that BA is *generally* superior PSO, it is believed that it is not necessary to compare both algorithms on the elusive best possible parameter settings. This is because if general superiority really existed, it should not require extensive parameter tuning to observe. In other words, the dominance pattern should be apparent. Despite this, efforts were made to find good parameter settings to make the comparisons as fair as possible. The parameters for BA were chosen by measuring median FHT for 100 runs on a multidimensional grid of parameter combinations and selecting the combination whose average ranks is the best on the test set. The ranges for the BA parameters are based on a meta-analysis (Xue et al, 2015) of historical parameter settings. The parameter settings studied are given in Table 3.1.

Table 3.1 BA parameter settings for grid search

Parameter	**Symbol**	**Values**
Frequency	f	$[0, 1], [0, 2]$
Pulse rate	r_0	$0.1, 0.4, 0.7, 1$
Loudness	A	$0.5, 1, 1.4, 1.9$
Alpha	α	$0.5, 1, 1.4, 1.9$
Gamma	γ	$0.1, 0.5, 0.9$

This results in $2 \times 4^3 \times 3 = 384$ combinations and $384 \times 100 = 38{,}400$ runs in total. Each run is limited to 2,500 epochs for a total maximum number of epochs of 96×10^6 epochs. Since the number of epochs is limited, sometimes BA never hits the target. This is accounted for by adjusting the median FHT values of each configuration by dividing by the hit rate. For example, if the median FHT is 500 function evaluations and the hit rate is 90%, then the actual FHT used for comparisons is $500/0.9 = 555.\overline{5}$ function evaluations. This is to prevent a bad configuration from winning by having a small number of good hits with low hit rate. The resulting best parameter set for BA is $f \in [0{,}2], r_0 = 0.7, A = 1, \alpha = 1.9, \gamma = 0.1$. These results can be reproduced or improved upon using the following MATLAB file in the referenced GitHub repository: ENG2019110845_Parameter_Selection_BA.m. A similar exercise with PSO yielded $\omega = c_1 = c_2 = 0.7$ and corresponds to ENG2019110845_Parameter_Selection_PSO.m.

Table 3.2 Parameter settings for BA and PSO

BA		**PSO**	
Parameter	**Value**	**Parameter**	**Value**
f	$[0, 2]$	ω	0.7
r_0	0.7	c_1	0.7
A	1	c_2	0.7
α	1.9	–	–
γ	0.1	–	–

It is interesting to note that the best α value is 1.9 since this causes $U(0,1) < A_i^t$ to always be true (i.e., because $A^0 = 1$ implies that $A_i^t \geq 1$ for all values of t and i). This suggests that line 11 of Algorithm 3.2 can be simplified. This finding will be discussed in the results section considering the hypothesis stated in section 4 that the loudness mechanism is inferior or equal to the SA probabilistic acceptance mechanism.

3.5.3 Experiments and nonparametric statistical tests

This section describes the experimental procedures used to investigate the hypotheses formulated in section 3.4 . section 3.5.3.1 describes how the research hypotheses will be explored and section 3.5.3.2 describes Wilcoxon-Mann-Whitney test to compare medians.

3.5.3.1 Investigating the research hypotheses

This section describes the experimental procedures used to investigate the research hypotheses which are repeated here for convenience:

1. BA is suspected of being highly sensitive to initial positions.
2. The pulse rate mechanism is suspected of being no better than roulette wheel selection.
3. The loudness mechanism is suspected of being inferior or equal to the probabilistic acceptance mechanism of Simulated Annealing.

To measure the effect of different initial positions on BA's performance (hypothesis 1), the two following schemes will be used:

1. Random uniform over the entire domain.
2. Random uniform in a restricted domain that does not contain the global minimum.

The domains are restricted to exclude a unit square centered on the global minimum for f_1, f_2 and f_3. Since the domain of f_4 is only 2.56×2.56, the excluded square is 0.25×0.25 and 5×5 for f_5 because of its larger domain.

To compare the pulse rate mechanism with roulette wheel selection (hypothesis 2), a variant of BA (ba_roulette_wheel.m) will be used to generate FHT distributions. A uniform probability roulette wheel with $\mathbb{P}(\text{true}) = \mathbb{P}(\text{false}) = 0.5$ will be used as shown in Algorithm 3.3. The line numbers on the left-hand side of Algorithm 3.3 correspond to the line numbers of Algorithm 3.2.

To measure the effects of the loudness mechanism (hypothesis 3), a variant of BA with no loudness mechanism (ba_no_loudness.m) as shown in Algorithm 7 will be used to generate FHT distributions.

The resulting FHT distributions will be analyzed and compared using the nonparametric tests described in the next two sections. The reason why nonparametric tests are used is because the normality assumption does not hold for FHT distributions since FHT is naturally greater than or equal to 1.

3.5.3.2 The Wilcoxon-Mann-Whitney Test

To compare performance between BA and PSO, FHT distributions are generated for both algorithms and the one with the smallest median FHT is the winner for a given benchmark. The differences in medians confirmed to a 0.05 confidence level using the Wilcoxon-Mann-Whitney test with MATLAB's built-in ranksum function in the Statistics and Machine Learning Toolbox (https://www.mathworks.com/help/stats/ranksum.html). The technical details of the Wilcoxon-Mann-Whitney nonparametric test (WMW hereafter) are given in (Sheskin, 2003).

In short, WMW takes two FHT populations, say X_1 and X_2 of size n_1 and n_2 respectively and tests the following:

$$\boldsymbol{H_0}: \quad \boldsymbol{P(X_1 > X_2) = P(X_2 > X_1)} \tag{3.17}$$

$$\boldsymbol{H_a}: \quad \boldsymbol{P(X_1 > X_2) \neq P(X_2 > X_1)} \tag{3.18}$$

If $\boldsymbol{H_0}$ is true and $\boldsymbol{S_{X_1}}$ is the sum of the ranks of the $\boldsymbol{n_1}$ elements in $\boldsymbol{X_1 \cup X_2}$, then it can be shown that the even $\boldsymbol{S_{X_1} = S_{X_2}}$ approximately follows $\boldsymbol{N\left(\mu = \frac{n_1 n_2}{2}, \sigma^2 = \frac{n_1 n_2 (n_1+n_2+1)}{12}\right)}$ provided that $\boldsymbol{n_1 > 20}$ and $\boldsymbol{n_2 > 20}$. From this, the inverse normal distribution can be used to reject the null hypothesis if the test statistic $\boldsymbol{\epsilon = \left|S_{X_1} - \mu\right|/\sigma}$ is greater than 1.96 for a 0.05 confidence level or 2.58 for a 0.01 confidence level.

3.6 Results

This section contains the results obtained by following the protocol given in section 3.5.3. Section 3.6.1 contains the results of the effect of two different initialization schemes, section 3.6.2 contains the results of the effect of replacing the pulse rate mechanism with roulette wheel selection, section section 3.6.3 contains the results of the effect of removing the loudness mechanism and 3.6.4 contains additional observations.

3.6.1 Sensitivity to initial conditions

The FHT distributions were evaluated for BA and PSO with random uniform initialization on the full domain (Adjusted Median FHT A) and random uniform initialization on a restricted domain (Adjusted Median FHT B). The adjusted medians are calculated, and statistical significance is assessed using WMW. The results are presented in Table 3.3 and Table 3.4.

Table 3.3 Impact of initialization scheme on median FHT for BA

Function	Adjusted Median FHT A	Adjusted Median FHT B	Difference	Statistically significant?
f_1	608	816	+ 34.2 %	Yes
f_2	3,604	9,843	+ 173.1 %	Yes
f_3	452,870	487,360	+ 7.6 %	No
f_4	2,670	3,340	+ 25.1 %	Yes
f_5	76,000	55,789	− 26.5 %	No

Table 3.4 Impact of initialization scheme on median FHT for PSO

Function	Adjusted Median FHT A	Adjusted Median FHT B	Difference	Statistically significant?
f_1	280	360	+ 28.6 %	Yes
f_2	4,220	4,960	+ 17.7 %	No
f_3	11,099	11,808	+ 6.4 %	No
f_4	1,740	2,240	+ 28.7 %	Yes
f_5	8,191	9,710	+ 18.5 %	Yes

The results for BA on f_3 and f_5 are consistent with those obtained during the parameter selection process. BA severely underperforms PSO on these functions and initialization does not appear to play a statistically significant role. It is believed that BA mostly finds the global minima for f_3 and f_5 by chance. The mean difference in adjusted median FHT values for f_1, f_2 and f_4 is 77.5% for BA and 25% for PSO. Both algorithms are similarly affected by the restricted initialization scheme on f_1 and f_4, so it could be argued that BA's adjusted median FHT difference of 173.1% on f_2 is an outlier, but it is believed that this is not the case. It is proposed instead that the structural resemblance between f_1 and f_4 explain their similar sensitivities to initialization (i.e. both are bowl-shaped convex or almost convex functions). This hypothesis is supported by the fact that the same differences for PSO in Table 4 are almost identical (i.e., 28.6 % and 28.7 %). The conclusion is that BA is in fact very sensitive to initialization and that the observed differences are statistically significant to a 0.05 confidence level using the WMW test with a sample of 1,000 observations.

3.6.2 BA with roulette wheel selection instead of pulse rate

The FHT distributions ($n = 1{,}000$) were evaluated for BA with the pulse rate mechanism (Adjusted Median FHT A) and with roulette wheel selection (Adjusted Median FHT B). The adjusted medians are calculated, and statistical significance is assessed using WMW. The results are presented in Table 3.5.

Table 3.5 Effect of replacing pulse rate with roulette wheel selection

Function	Adjusted Median FHT A	Adjusted Median FHT B	Difference	Statistically significant?
f_1	611	254	− 58.4 %	Yes
f_2	4,070	3,683	− 9.5 %	Yes
f_3	478,550	3,470	− 99.2 %	Yes
f_4	2,550	2,000	− 21.6 %	Yes
f_5	184,510	14,894	− 91.9 %	Yes

Using a uniform probability roulette wheel instead of the pulse rate mechanism has increased performance by 56.1% on average. The adjusted median FHT has decreased in all 5 test functions, and the differences are statistically significant based on the WMW test with a sample of 1,000 observations. The differences are particularly important for the functions where BA did poorly, namely f_3 and f_5. With this simple change, BA suddenly becomes competitive with PSO (see Table 3.4 for comparison). It is conjectured that this is due to the lessened impact of component 2 during the early phases of the optimization process. If this is true, then the initialization scheme is expected to have less of a negative impact. This was confirmed during a later experiment. Table 6 shows the adjusted median FHT with random uniform initialization (Adjusted Median FHT A) and initialization on a restricted domain (Adjusted Median FHT B). It can be seen that the differences are smaller with an average increase in FHT of 8.7% against 42.7% (Table 3.3).

Table 3.6 Effect of initialization on BA with roulette wheel selection

Function	Adjusted Median FHT A	Adjusted Median FHT B	Difference	Statistically significant?
f_1	254	281	+ 10.8 %	Yes
f_2	3,855	4,642	+ 20.4 %	No
f_3	3,581	3,551	− 0.8 %	No
f_4	1,980	1,950	− 1.5 %	No
f_5	12,672	14,548	+ 14.8 %	No

3.6.3 BA without the loudness mechanism

The FHT distributions ($n = 1{,}000$) were evaluated for BA with the loudness mechanism (Adjusted Median FHT A) and without it (Adjusted Median FHT B). The adjusted medians are calculated, and statistical significance is assessed using WMW. The results are presented in Table 3.7.

Table 3.7 Effect of removing the loudness mechanism

Function	Adjusted Median FHT A	Adjusted Median FHT B	Difference	Statistically significant?
f_1	622	426	− 31.6 %	Yes
f_2	4,185	1,058	− 74.7 %	Yes
f_3	377,760	493,110	+ 30.5 %	No
f_4	2,680	1,820	− 32.1 %	Yes
f_5	147,440	236,630	+ 60.5 %	Yes

Adjusted median FHT decreased by an average of 46.1% for f_1, f_2 and f_4 and increased by an average of 45.5% for f_3 and f_5. The difference is not statistically significant for f_3 despite the difference being large because the sizes of the samples that did find the target is small ($n_A = 134$, $n_B = 119$). It is believed that this difference is caused by fewer improving positions being rejected therefore making BA greedier (i.e., less explorative). This is supported by the fact that performance decreased for those functions where more exploration is beneficial, namely f_3 and f_5 and improved by a substantial amount otherwise.

3.6.4 Additional observations

Figure 3.11 and Figure 3.12 show that the original BA does behave like a stochastic hill climber. The global best positions are indicated by yellow stars (★) and the movements of the swarm centroids are shown by the red lines with the red cross markers (+). The successive generations of swarms are plotted as black circles in decreasing order of transparency. The first generation is the most transparent and the most recent generations are

the opaquest. These figures show more or less exactly how one would expect a hill climber to navigate the landscape.

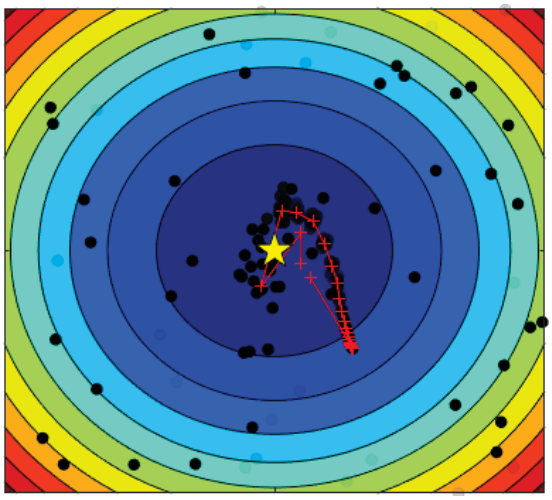

Figure 3.11 BA centroids on f_1

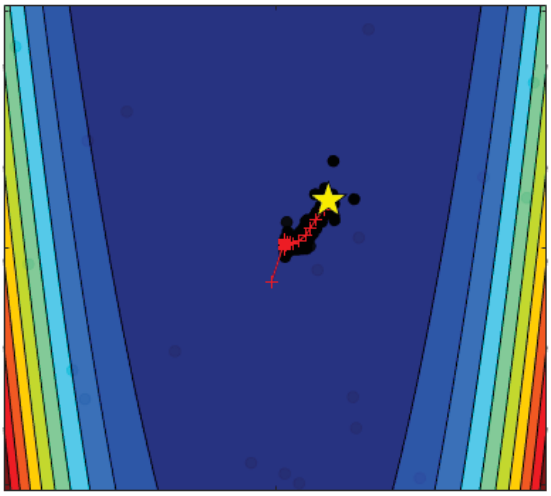

Figure 3.12 BA centroids on f_2

The stochastic hill climbing behaviour of BA is clearly beneficial for convex functions and ridges with non-zero gradients, but it does appear to pose problems for functions with discontinuities and/or noise as one would expect. Figure 3.13 shows BA stagnating on the plateaus of f_3 and Figure 3.14 shows BA centroids stagnating over 1,000 epochs on f_4 in the two regions circled in green.

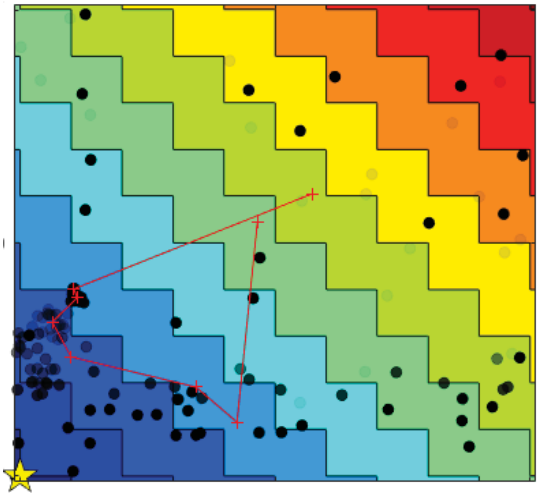

Figure 3.13 BA centroids on f_3

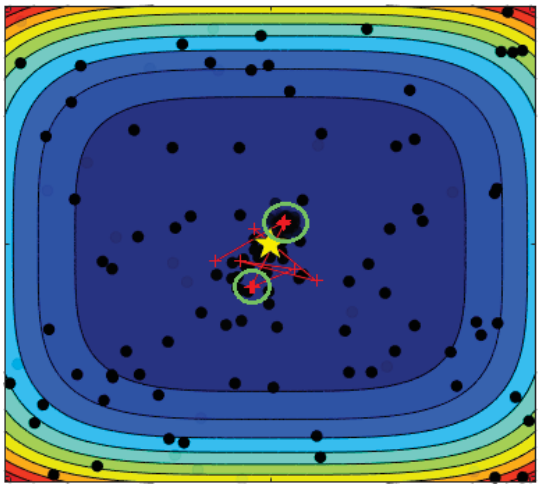

Figure 3.14 BA centroids on f_4

The Foxholes function (f_5) is absent simply because BA consistently fails early on it (i.e., the swarm "falls" into one of the local minima and stays there the entire time) so the centroid movements are uninteresting. This observation suggests that BA could possibly be used to navigate objective functions that possess deep ridges like the HappyCat function described in (Beyer & Finck, 2012) shown in Figure 3.15 along with the level curves that inspired its name. This is left as a future research project.

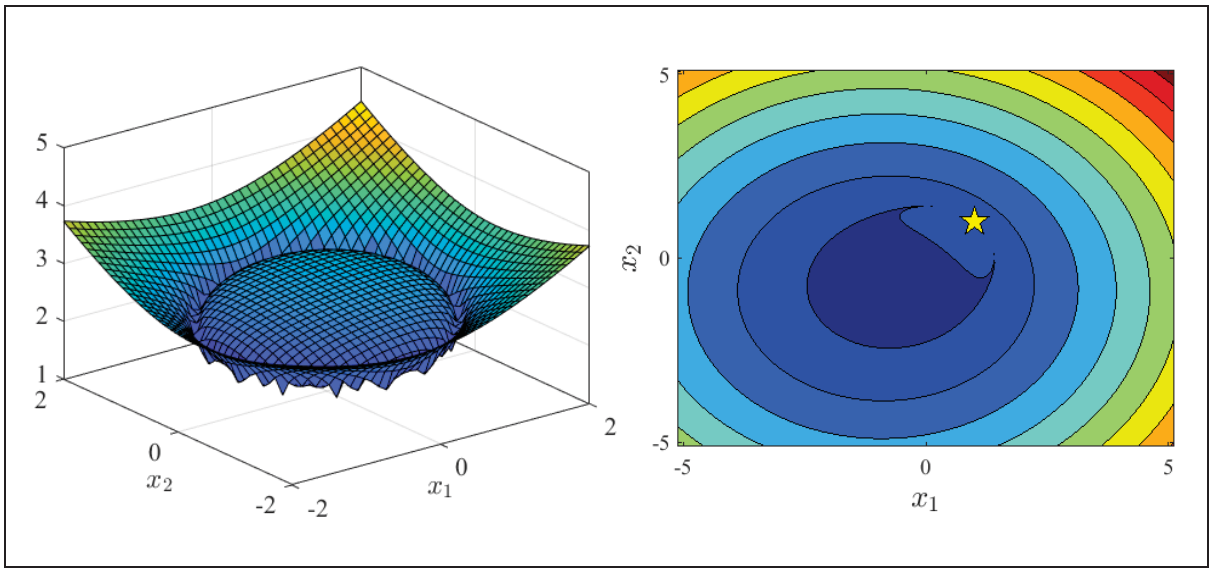

Figure 3.15 HappyCat function with $\alpha = 1/8$ on $x_i \in [-2, 2]$

3.7 Conclusion

The results presented in the previous sections disprove the claim made in (Yang, 2010) that BA is generally superior to PSO. In fact, the results support the opposite. This research also exposed two weaknesses of BA: plateaus and noisy objective functions. This might be attributable to parameter selection, but the results of the parameter selection process described in section 3.5.2 does not support this hypothesis. Instead, it is assumed that BA is simply not well-suited to these objective function characteristics. This being said, the tight swarm configuration of BA observed in section 3.6.4 does appear to be good at finding promising directions in regions where they are rare (e.g., ridges) and the gradients are low (e.g., the banana-shaped valley of f_2) and exploiting them. This finding suggests that BA might be beneficial to the study of neural network landscapes since it was reported in (Garipov, Izmailov, Podoprikhin, Vetrov & Wilson, 2018) that low-energy regions are connected to one another by relatively simple curves. In other words, the optimal regions share similarities with the banana-shaped valley of f_2. The results also support the research hypotheses. The first hypothesis is that BA is highly sensitive to initial conditions. This was confirmed to be true by comparing the effects of two initialization schemes between BA and PSO. The second hypothesis posited that the pulse rate mechanism is no better than roulette wheel selection. Not only was this confirmed, but the experiments also showed that BA's

sensitivity to initial conditions almost entirely disappeared when uniform probability roulette wheel selection was used instead of the pulse rate mechanism. The third and last hypothesis was that the loudness mechanism was inferior or equal to SA's probabilistic acceptance mechanism. Also, in our study, the loudness mechanism could be abandoned entirely for those functions where BA already performed well relative to PSO. All of this suggests that the best-performing configuration of BA is a hybrid between PSO and SA. Further studies could confirm this by opposing this hybrid configuration against BA both with optimal parameters determined with design of experiments and response surface methodology. To conclude, it is important to mention that the findings presented in this article should, in the opinion of the authors, have been made well before presenting BA as an original contribution that outperforms PSO when it is easy to show that both of these statements are incorrect.

CHAPTER 4

AN INVESTIGATION OF THE EFFECTS OF CHAOTIC MAPS ON THE PERFORMANCE OF METAHEURISTICS

Iannick Gagnon[a], Alain April[a] and Alain Abran[a]

[a] Department of Software and IT Engineering, École de Technologie Supérieure, 1100 Rue Notre-Dame West, Montreal, Quebec, Canada, H3C1K3

Paper published in *Engineering Reports*, December 2020

Abstract

This article presents an empirical investigation of the effects of chaotic maps on the performance of metaheuristics. Particle Swarm Optimization and Simulated Annealing are modified to use chaotic maps instead of the traditional pseudorandom number generators and then compared on 5 common benchmark functions using nonparametric null hypothesis statistical testing. Contrary to what has often been assumed, results show that chaotic maps do not generally appear to increase the performance of swarm metaheuristics in a statistically significant way, except possibly for noisy functions. No performance differences were observed with the single state Simulated Annealing algorithm. Finally, it is shown that sequence effects may be responsible for the observed performance increase. These findings reveal new research directions in using chaotic maps for metaheuristics research. The MATLAB code used in this article is available in a GitHub repository for suggestions and/or corrections.

Keywords

Metaheuristics, chaotic maps, particle swarm optimization, simulated annealing.

4.1 Introduction

Metaheuristics use randomness to diversify the search process and escape local minima (Clerc, 2015). A recent trend (Figure 4.1) in metaheuristics research is to replace traditional pseudo-random number generators (PRNGs) with chaotic maps. Figure 1 shows an

increasing trend in yearly publications on metaheuristics with chaotic maps as reported in Web of Science using the query “chaos AND optimization” between 2010 and 2018. It is a re-emerging phenomenon dating back to the mid-1990s. See (Chen & Aihara, 1995) for an example.

This movement leaves important questions unanswered. For example, there does not appear to be any fundamental reason chaotic maps should increase the performance of metaheuristics. Articles on the subject generally conclude that chaotic maps improve the performance of metaheuristics, but do not investigate further. This article gathers evidence for performance differences between metaheuristics with and without chaotic maps. More data is required partly because NHST is rarely used in the literature. When statistical significance measures are present, as in (Mitić, Vuković, Petrović & Miljković, 2015) for example, effect sizes are almost never discussed and there are often easily identifiable methodological flaws that prevent firm conclusions. Such flaws include problematic parameterization and arbitrary measures of performance, among others. Unfortunately, since source code is rarely made available, further investigation is often impossible. This research distinguishes itself by using NHST with large samples, using experimental design for parameterization, reporting effect sizes, introducing the notion of sequence effects and making the data and the computer code available to the readers.

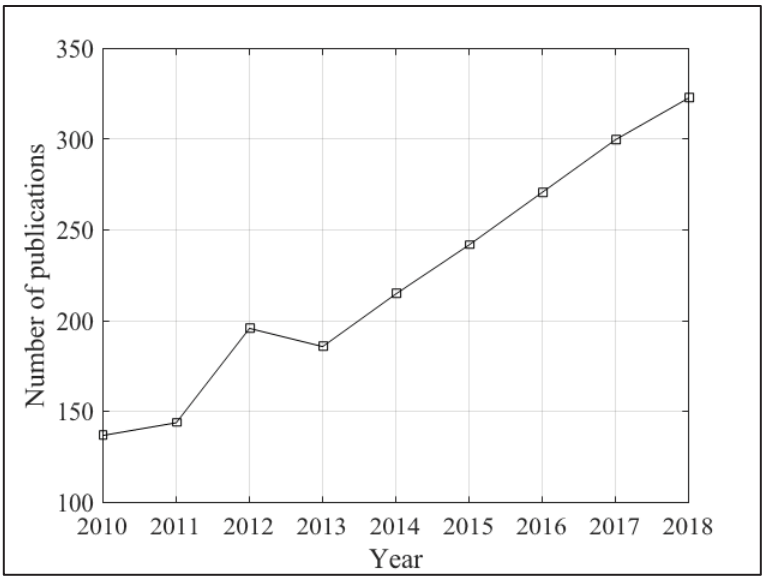

Figure 4.1 Number of yearly publications on optimization with chaos

The rest of this article is organized as follows. Section 4.2 contains a brief literature review of recent applications of chaotic maps with metaheuristic algorithms. Section 4.3 explains the methodology used in this research. Section 4.4 and Section 4.5 present the results for PSO and SA respectively. Section 4.6 discusses the limitations of this study, and Section 4.7 offers a conclusion and proposes further research directions.

4.1.1 Chaotic maps

Chaotic maps differ from PRNGs because they are deterministic. In theory, given an infinitely precise initial condition (x_0) and an iterative function of the form $x_{t+1} = f(x_t)$, the state (x) of a system can be calculated at any future time (t). Chaotic maps are random for two reasons: (1) the fact that infinite precision is impossible and (2) the high sensitivity to initial conditions. It is therefore impossible to make good long-term predictions because the initial conditions cannot be stored or read exactly. The error, however small, will eventually cause divergence between the ideal system and the real one. For example, Figure 4.2 shows the iterative function $x_{t+1} = x_t(1 - x_t)$ for $t \epsilon [0, 40]$ with initial conditions $x_0 = 0.4$ (black) and $x_0 = 0.4 + 10^{-8}$ (red).

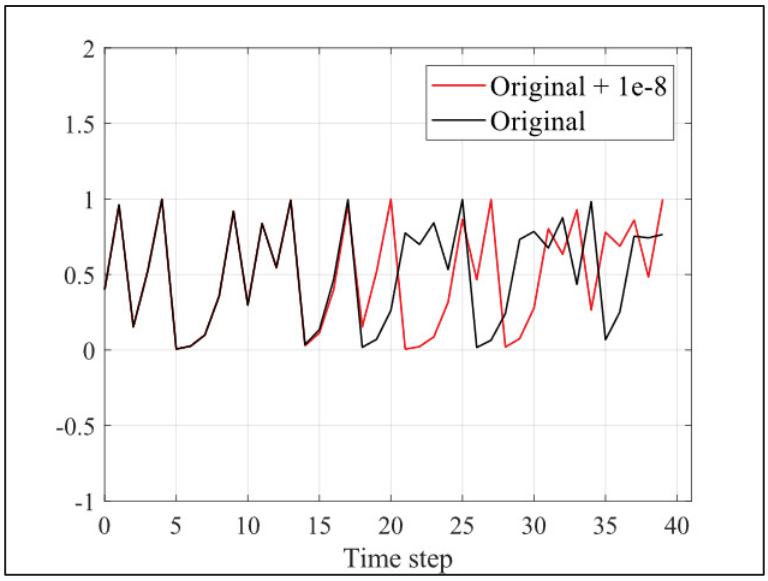

Figure 4.2 Chaotic behaviour of $x_{t+1} = 4x_t(1 - x_t)$

Figure 4.3 shows that up to $t = 15$, the absolute difference between the two curves is nearly zero before shooting up rapidly and oscillating.

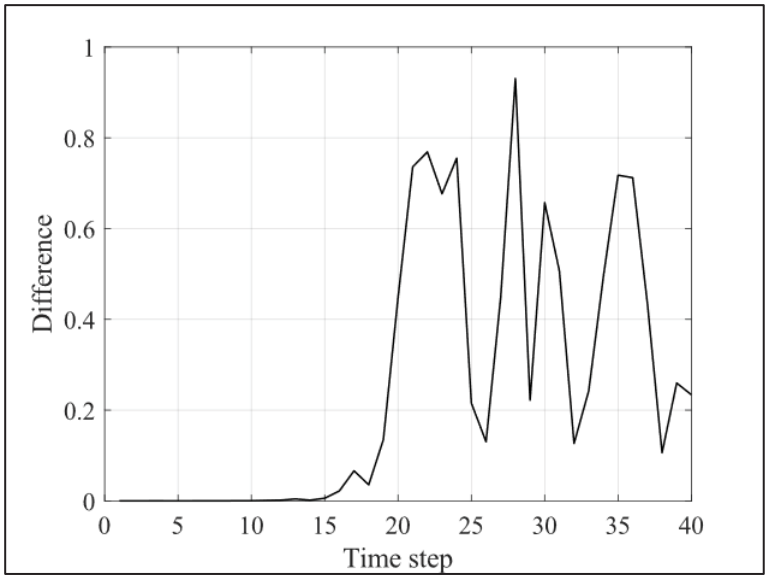

Figure 4.3 Difference between chaotic maps with initial conditions that differ by 10^{-8}

Chaotic maps can therefore be used as PRNGs that are "random enough" for metaheuristics research. Figure 4.4 shows the 6 maps plotted on the interval $T = t \in [0, 40]$.

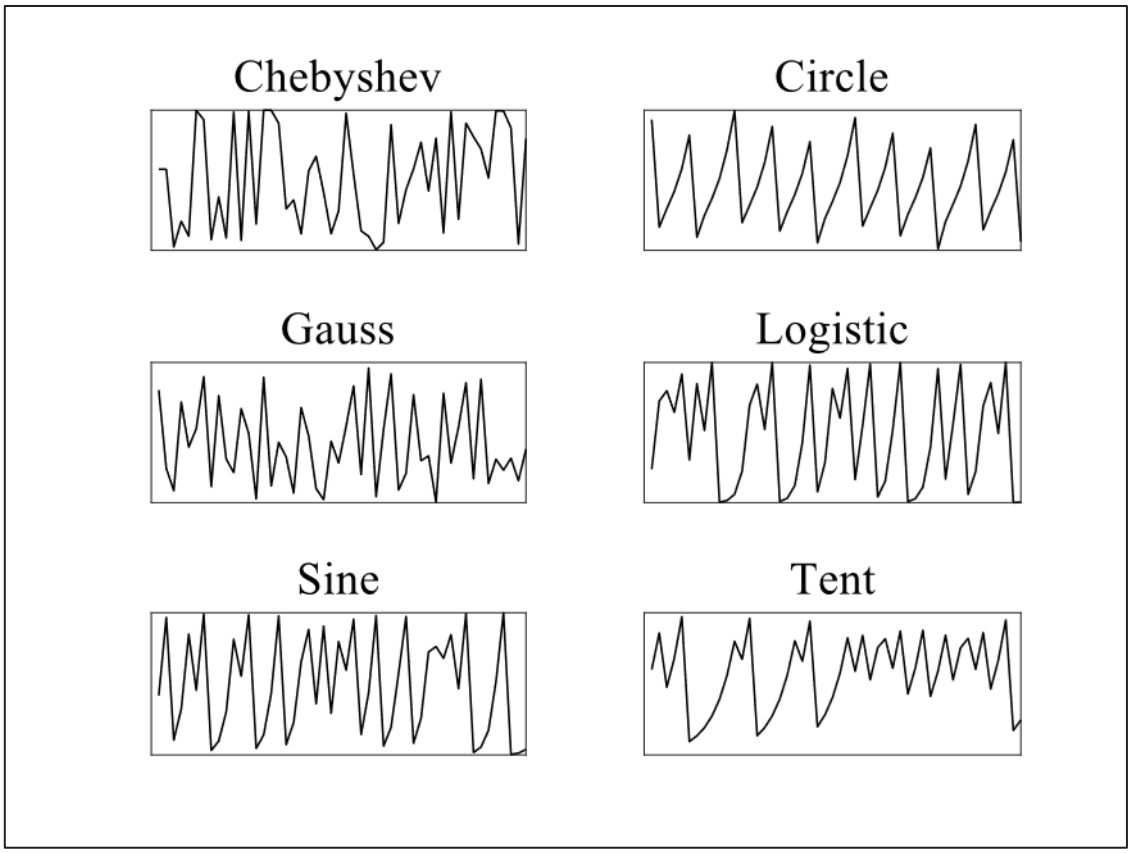

Figure 4.4 Chaotic maps

The Chebyshev map is defined by the following iterative function:

$$x_{t+1} = \cos[t \operatorname{acos}(x_t)] \tag{4.1}$$

The Chebyshev map EPDF is given in Figure 4.5.

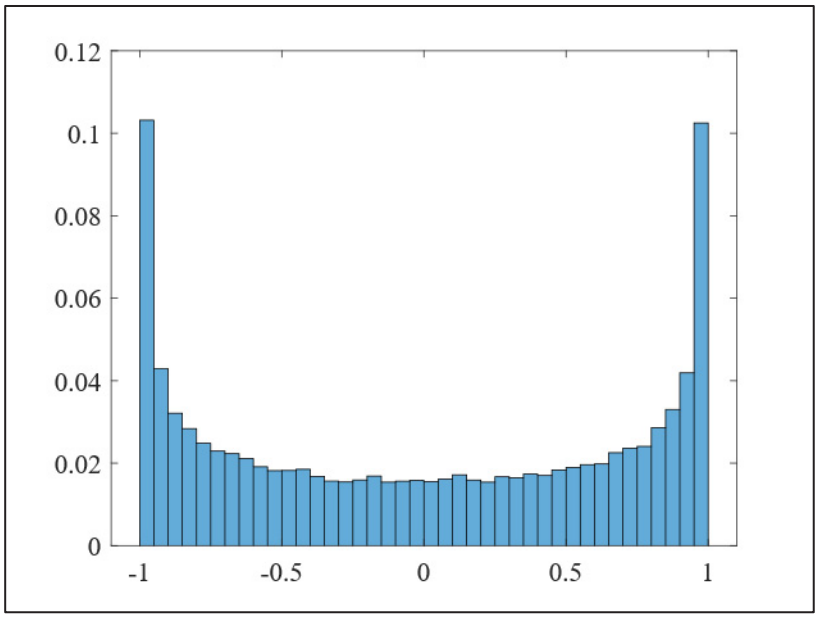

Figure 4.5 Chebyshev map EPDF

The Circle map is defined by the following iterative function with $\Omega = 0.2$ and $K = 0.5$:

$$x_{t+1} = \left(x_t + \Omega - \frac{K}{2\pi}\sin(2\pi x_t)\right) \bmod 1 \tag{4.2}$$

The Circle map EPDF is given in Figure 4.6.

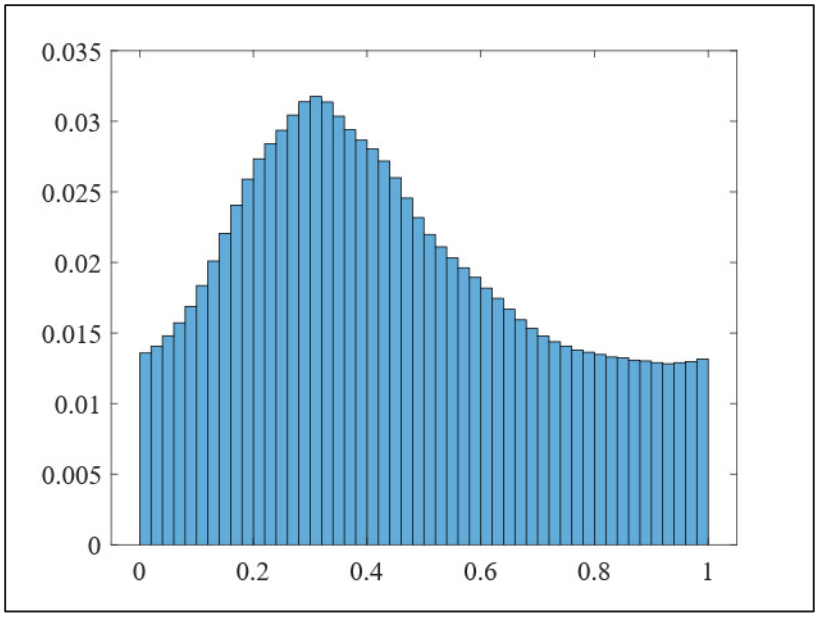

Figure 4.6 Circle map EPDF

The Gauss map is defined by the following iterative function:

$$x_{t+1} = \left(\frac{1}{x_t}\right) \bmod 1 \tag{4.3}$$

The Gauss map EPDF is given in Figure 4.7.

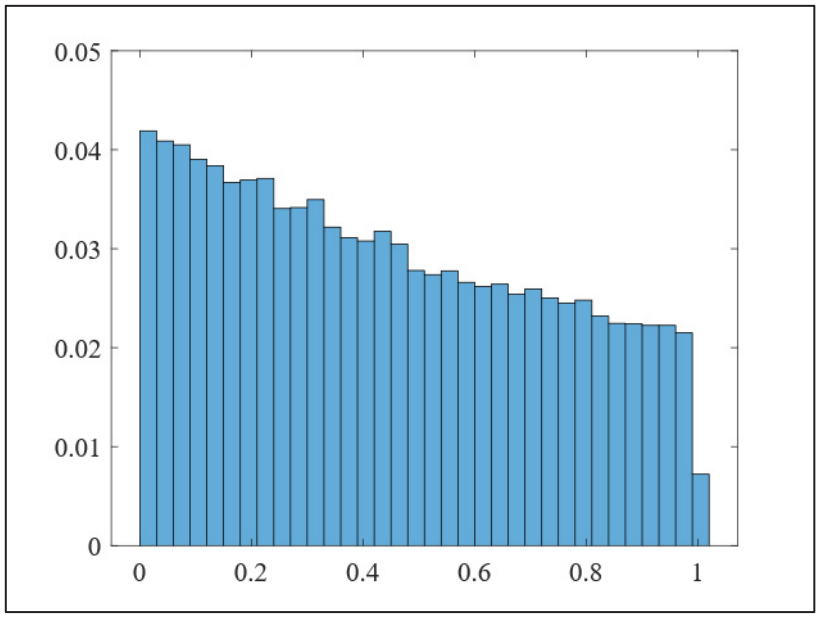

Figure 4.7 Gauss map EPDF

The Logistic map is defined by the following iterative function with $r = 4$:

$$x_{t+1} = rx_t(1 - x_t) \tag{4.4}$$

The Logistic map EPDF is given in Figure 4.8.

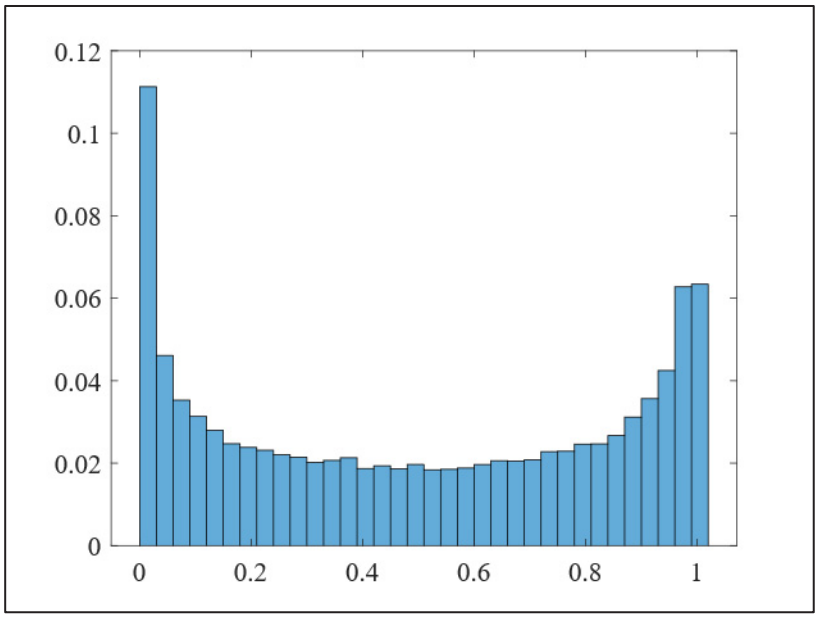

Figure 4.8 Logistic map EPDF

A variant of the Sine map is defined by the iterative formula:

$$x_{t+1} = \sin(\pi x_t) \tag{4.5}$$

The Sine map EPDF is given in Figure 4.9.

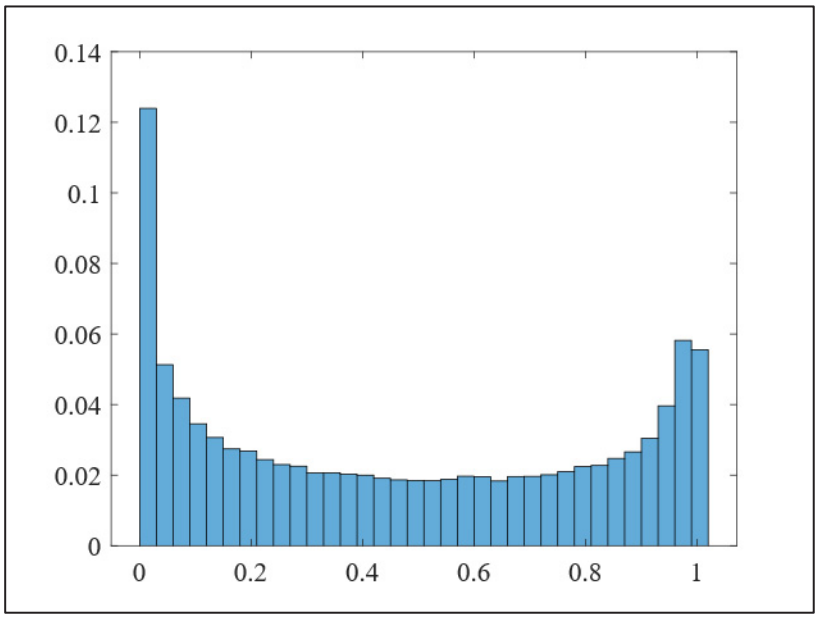

Figure 4.9 Sine map EPDF

The Tent map is defined by the following recurrence formula with $\mu = 10/3$:

$$x_{t+1} = \begin{cases} \mu x_t & \text{for } x_t < \frac{1}{2} \\ \mu(1 - x_t) & \text{for } \frac{1}{2} \leq x_t \end{cases} \tag{4.6}$$

The Tent map EPDF is given in Figure 4.10.

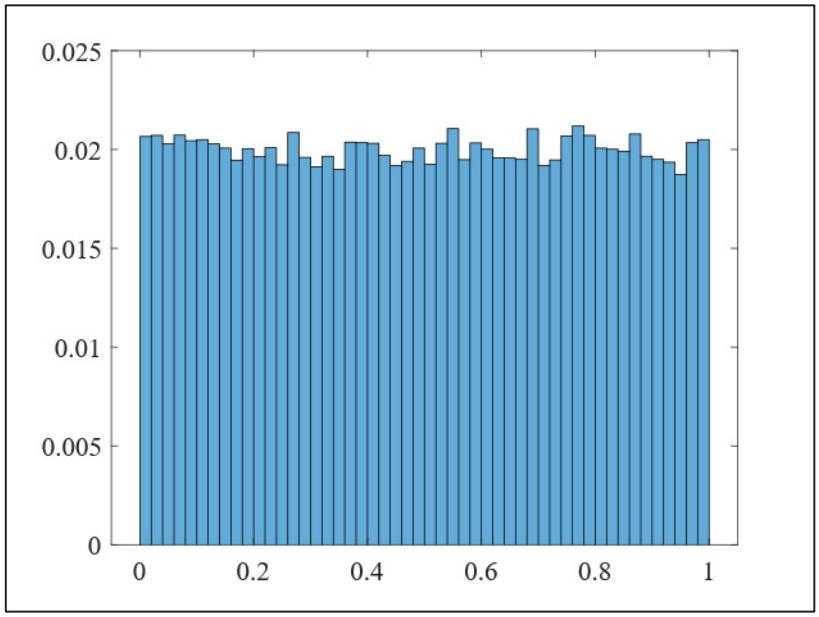

Figure 4.10 Tent map EPDF

Once equations 4.1 to 4.6 are seeded with initial values (x_0), they are used to produce pseudorandom sequences that replace PRNGs.

4.1.2 Particle swarm optimization and simulated annealing

The PSO algorithm (Algorithm 4.1) is used as a baseline for swarm algorithms in this paper. While it is true that the performance and behaviour of PSO highly depend on parameters, many of the more recent nature-inspired metaheuristics, such as the above Bat Algorithm, are considered by some to be marginally different variants of PSO (Gagnon, April & Abran, 2020).

PSO updates the particles' positions in ($\vec{x}_i$) d-dimensional space using what is called a velocity term ($\vec{v}_i$) which is a weighed sum of the previous velocities using an inertia term (ω), the difference vector between the swarm's best known position ($\vec{x}^*$) with current positions multiplied by a constant (c_1) and the random number (r_1) and finally the difference vector between each particles' own historical best position ($\vec{p}_i$) with current positions also multiplied by a constant (c_2) and a random number (r_2). For a more in-depth explanation, see [13]. The random numbers r_1 and r_2 are usually picked from a random uniform distribution with support $[0, 1]$ and are the ones replaced by chaotic maps. The version of PSO used in this research limits each particle's velocity to 15% of the length of the search space. When a particle's velocity exceeds this value, it is truncated to the maximum allowed velocity. This is known in the PSO literature as velocity clamping.

SA (Algorithm 4.1) is one of the simplest metaheuristics to implement, but it has proved to be very capable. It uses a probabilistic acceptance function that allows moves in directions that decrease fitness. This trade-off allows it to escape local minima provided they are not too deep (i.e., ΔQ is small). SA uses a decreasing temperature schedule of the form $T = f(t)$ and a move operator implemented as a function called GenerateNewSolution. The new position is accepted according to a probabilistic function based on fitness difference (ΔQ) and temperature (T). This article uses a linearly decreasing function from $T_{max} = 1000$ to 0 in $t_{max} = 10^4$ time steps. The random uniform distribution of line 5 is replaced by chaotic maps in Chaotic Simulated Annealing (CSA).

Algorithm 4.1 Pseudocode for minimization with SA

Global minimization procedure with SA

Input: The cooling schedule $T(t)$.

Output: An array corresponding to the best-found position $\vec{x}^*$.

```
initialize x⃗ , v⃗, x⃗* and p⃗ with zeros
while t < t_max do
    x⃗_candidate ← GenerateNewSolution(x⃗^t)
    Q_candidate ← f(x⃗^(t+1))
    if U(0,1) < e^(−ΔQ/T(t)) or Q_candidate > Q_best then
        x⃗* ← x⃗_candidate
    end if
  return x⃗*
```

4.2 Literature review

Several metaheuristics have been reworked with chaotic maps including the Firefly Algorithm (Gandomi, Yang & Talatahari, 2013), the Bat Algorithm (Gandomi & Yang, 2014), Cuckoo Search (Wang, Deb, Gandomi, Zhang & Alavi, 2016), Krill Herd (Saremi, Mirjalili & Mirjalili, 2014), Fruit Fly Optimization Algorithm (Mitić, Vuković, Petrović & Miljković, 2015), Particle Swarm Optimization (PSO) (Alatas, Akin & Ozer, 2009), Ant Colony Optimization (Cai et al., 2007) and more. These articles unanimously conclude that chaotic maps improve performance and explain this phenomenon mostly by an increased capacity to avoid/escape local minima.

Articles (Gandomi, Yang, Talatahari & Alavi, 2013) and (Gandomi & Yang, 2014) use descriptive statistics with $n = 100$ observations without NHST, but the article still concludes that some maps are better than others. The same approach is found in (Wang et al., 2015) with $n = 1{,}000$ observations. The chaotic Krill Herd algorithm in (Saremi et al., 2014) is declared superior to its original version without NHST and a sample size of only $n = 10$ observations. The Chebyshev map is declared superior "in terms of reliability of global optimality and algorithm success rate" to other chaotic maps in (Mitić et al., 2015)

based on comparisons made between descriptive statistics with $n = 50$ observations. A similar approach is used in (Alatas et al., 2009) and (Cai et al., 2007) to conclude that chaotic maps improve the search performance of metaheuristics. The most apparent shortcoming observed is that the surveyed literature foregoes NHST and instead uses descriptive statistics alone for performance comparisons. There is also no discussion about the practical significance of the magnitudes of the alleged differences in performance. Some articles such as (Sayed, Tharwat & Hassanien, 2019) use statistical tests to conclude that chaotic maps improve performance, but no hypotheses are given, or analysis done to explain why. The original algorithm's random variable is sampled from a Lévy distribution that, once its EPDF is examined, shows that over 99% of its area is comprised between 0 and 0.1. This means that the original algorithm is likely too greedy for multimodal functions. The article concludes that the Gauss map outperforms others. Interestingly, the Gauss map EPDF is a right-skewed, which tends to make the algorithm greedy also. In numbers, the Gauss map will return values between 0 and 0.1. It is believed that this result is inconclusive because the effect sizes are not reported and because the Lévy distribution results in poor baseline performance. Chaotic maps evidently generate much interest in the research community, but their effectiveness remains an open question.

4.3 Methodology

This section describes the methodological approach used to investigate the research question. Section 4.3.1 describes the performance metric used to compare the algorithms. Section 0 covers NHST and effect size considerations. Section 4.3.3 describes how PSO and CPSO parameters were selected using experimental design. Finally, Section 4.3.4 describes the benchmark functions used for the experiments. The data and code that support the findings of this study are openly available at https://github.com/iangagn/ENG-2019-12-0887.

4.3.1 Definition of performance

Performance is typically defined as either the best or mean value found by an optimizer over several runs. This research deviates from this standard by using FHT measures. It measures the number of objective function evaluations necessary to reach the global optimum within

a Euclidian distance ε set to 10^{-2}. The FHT measure was selected because it combines solution quality with computational effort and therefore represents a more balanced and practical gauge of performance.

4.3.2 Null hypothesis statistical testing and effect size.

Preliminary experiments were run to decide between parametric and nonparametric statistical tests. It was found that median FHT distributions follow exponential-like distributions. The nonparametric 2-sided Wilcoxon Rank-Sum test was selected to test the null hypothesis that two FHT distributions (i.e., two algorithms) have equal medians against the alternative hypothesis that they do not. The homogeneity of variance assumption was validated using the modified Levene's test presented in (Brown & Forsythe, 1974). All tests were performed at the 95% confidence level ($\alpha = 0.05$). When statistical significance was observed, 95% confidence intervals were calculated for median FHT differences ($\tilde{Y}_{CPSO} - \tilde{Y}_{PSO}$) using 10,000 bootstrap samples.

4.3.3 Parameter selection for PSO

To make fair comparisons, PSO and CPSO parameters were defined to be the ones that give the best rank sum of median FHT values in a full factorial experiment using all benchmark functions. The considered values for ω, c_1 and c_2 based on typical values found in the literature are {0.5, 0.7, 0.9, 1.5} and the numbers of particles (n) considered are {20, 30, 40, 50} for a total of $4^4 = 256$ configurations. Each configuration is run 10^3 times until convergence to within $\varepsilon = 10^{-2}$ of the global minimum. The resulting best parameters are given in Table 4.1.

Table 4.1 Parameters for PSO and CPSO

Parameter	Value
n	40
ω	1.5
c_1	0.7
c_2	0.9

4.3.4 Test set

The test bench comprises the 5 De Jong test functions first described in (De Jong, 1975). These functions were selected because of their distinct characteristics: f_1 is convex, f_2 is multimodal with a low-gradient valley, f_3 has multiple discontinuous flat regions, f_4 is a convex quartic function with noise and f_5 is a combination of multiple steep basins with local minima and a large plateau. The small size of the test bench makes the results more amenable to analysis. Detailed descriptions are given in section 3.5.1.

4.4 Results and analysis for particle swarm optimization

Table 4.2 contains the median FHT values for PSO and CPSO. The p-values for the Wilcoxon Rank-Sum test (Table 4.3) are calculated using the normal approximation as detailed in (Walpole & Myers, 2002). Statistically significant p-values ($p < 0.05$) are boldfaced.

Table 4.2 Median FHT for PSO and CPSO

Map	f_1	f_2	f_3	f_4	f_5
Uniform	480	18,160	59,200	4,120	55,580
Chebyshev	1,200	15,440	58,100	3,680	53,400
Circle	380	17,640	62,660	4,080	56,660
Gauss	440	16,680	55,000	4,220	57,880
Logistic	840	16,720	58,700	3,840	53,360
Sine	840	17,760	60,800	3,800	56,840
Tent	880	18,020	64,440	3,880	62,280

Table 4.3 Wilcoxon Rank-Sum test p-values for CPSO

Map	f_1	f_2	f_3	f_4	f_5
Chebyshev	**0.00**	0.40	0.61	**0.01**	0.27
Circle	0.25	0.25	0.61	0.48	0.95
Gauss	0.34	0.13	0.96	0.81	0.75
Logistic	**0.00**	0.36	0.80	0.15	0.30
Sine	**0.00**	0.35	0.56	0.08	0.95
Tent	**0.00**	0.94	0.32	0.36	0.08

Table 4.3 reports 95% confidence intervals for the effect sizes of statistically significant median differences (Table 4.2) with respect to original PSO. Statistically significant results were observed for f_1 and f_3 only. For f_1, it is observed that PSO's performance degrades with the introduction of chaotic maps. It is hypothesized that it may be caused by the order in which the random values are presented. This is called a sequence effect. Since original PSO uses a uniform distribution, whose PDF closely approximates the Tent map's EPDF (Figure 4.9), one should not expect a statistically significant impact on performance unless sequence effects were present. The fact that the Logistic and Sine maps degrade performance to similar extents also points in the direction of sequence effects since their EPDFs are similar. It is also hypothesized that the Chebyshev map's distinctively poor performance is because its support is $[-1,1]$. For a convex function like f_1, this results in fitness-decreasing

moves away from the global optimum half of the time. The only case where a statistically significant positive effect was observed is with f_4 and the Chebyshev map. The effect size confidence interval was narrowed down from the value given in Table 4 to (-80, -440) with 5,000 runs ($p = 0.01$), which represents a 2 – 11% improvement over original PSO. Further experimentation showed that when the Chebyshev map is randomly shuffled, the median FHT difference was not statistically significant on f_4 ($p = 0.08$) for $n = 10^3$ samples. This supports the hypothesis that sequence effects are responsible for the observed statistically significant performance increase.

Table 4.4 PSO and CPSO median FHT differences 95% CI

Map	f_1	f_2	f_3	f_4	f_5
Chebyshev	(680, 800)	-	-	(-80, -980)	-
Circle	-	-	-	-	-
Gauss	-	-	-	-	-
Logistic	(320, 460)	-	-	-	-
Sine	(280, 440)	-	-	-	-
Tent	(360, 520)	-	-	-	-

4.5 Results and analysis for simulated annealing

Table 5 contains the median FHT values for SA and CSA. The p-values for the Wilcoxon Rank-Sum test (Table 4.6) are calculated using the normal approximation. Statistically significant p-values ($p < 0.05$) are boldfaced. The column for f_5 is empty because SA and CSA failed to converge within 10,000 objective function evaluations. SA and CSA are simply not well suited for highly multimodal functions because they are single state methods. The initial temperature is set to $T_0 = 1000$.

Table 4.5 Median FHT for SA and CSA

Map	f_1	f_2	f_3	f_4	f_5
Uniform	3,086	4,620	49	2,866	n/a
Chebyshev	3,279	4,632	53	2,937	n/a
Circle	3,111	4,697	49	2,905	n/a
Gauss	3,109	5,186	51	2,867	n/a
Logistic	3,183	4,602	51	3,120	n/a
Sine	3,048	4,586	51	2,790	n/a
Tent	3,086	5,094	50	3,055	n/a

Table 4.6 Wilcoxon Rank-Sum test p-values for CSA

Map	f_1	f_2	f_3	f_4	f_5
Chebyshev	0.27	0.80	0.09	0.92	n/a
Circle	0.97	0.86	0.76	0.89	n/a
Gauss	0.52	0.17	0.38	0.71	n/a
Logistic	0.20	0.41	0.34	0.13	n/a
Sine	0.92	0.52	0.76	0.83	n/a
Tent	0.13	0.17	0.43	0.39	n/a

Since there are no statistically significant performance differences, the effect size table (e.g., Table 4.4) is omitted. This suggests that the sequence effects identified in section 4.4 may only apply to swarm metaheuristics like PSO. Possibly this is because it has a specific influence on swarm diversity and therefore the exploration and exploitation capabilities of the algorithms.

4.6 Limitations of this study

As stated in Section 1, this study collects evidence for possible differences in performance between metaheuristics that use chaotic maps and those that use traditional PRNGs. One limitation of this study is that no *a priori* power analysis was performed. The reason for this is that since no distributional assumptions are made, there are no computationally affordable

methods to estimate statistical power for a specified effect size. Moreover, the effect size is difficult to fix in advance since the median FHT is not known *a priori*. This could be partly circumvented in further studies by using probabilistic distribution fitting to FHT distributions obtained for small sample sizes and Monte Carlo simulation. It should be noted that despite this, the sample sizes of $\boldsymbol{n = 1000}$ used in this research are considerably larger than what was generally observed in the literature. This results in greater statistical power. Another limitation of this study is that problem dimensionality is fixed (i.e., 2). As such, no conclusions are made about the scalability of the effects of chaotic maps on performance with respect to problem size. The reason for this is that since it was shown that research on chaotic maps and metaheuristics is not yet mature, the purpose is to gather evidence that may or may not warrant further research into subtopics like scalability. Finally, it should be noted that chaotic maps can be applied to other aspects of metaheuristics such as position initialization, but this research focuses on the most used method which consists of replacing the random number generators "inside" the algorithm.

4.7 Conclusion

The results presented in sections 4.4 and 4.5 show that chaotic maps do not improve performance in a statistically significant and general way for PSO and SA. For PSO, performance degraded between 58% and 444% on the convex function f_1 when chaotic maps were used. There were four cases where performance degraded against one where performance improved between 2% and 24%. In the latter case, it was found that when the chaotic map values are uniformly shuffled, no statistically significant performance increase was observed. These tests were performed on the only noisy function of the test bench. Further investigation could focus on the link between chaotic maps and performance on noisy test functions, which is an active area of research. It is hypothesized the observed performance increase is due to sequence effects, because the map that improved performance (i.e., Chebyshev) has an EPDF similar to the Logistic and Sine maps whose impacts were not statistically significant. Interestingly, no statistically significant performance differences were observed between SA and CSA, which suggests that chaotic maps have a different impact on swarm metaheuristics than on single state ones. This may be due to the effects on

swarm diversity, which is known to influence the exploration and exploration properties of swarm metaheuristics. This could be easily verified empirically in further experiments. Finally, sequence effects could be examined further by using time series analysis to expose the properties of chaotic maps under a different framework

CHAPTER 5

A QUANTITATIVE EVALUATION OF STATISTICAL PRACTICES IN METAHEURISTICS RESEARCH

Iannick Gagnon[a], Alain Abran[a] and Alain April[a]

[a] Department of Software and IT Engineering, École de Technologie Supérieure, 1100 Rue Notre-Dame West, Montreal, Quebec, Canada, H3C1K3

Paper submitted to *Engineering Reports*, August 2025

Abstract

This study identifies and measures several gaps in the use of statistical methods in metaheuristics research. This is accomplished by using a quantitative evaluation framework based on standard methodologies employed in the mature scientific fields of medicine, pharmacology, and psychology. A random sample of 70 peer-reviewed metaheuristics articles published between 2016 and 2021 was analyzed. It was found that authors favor descriptive statistics (79%, 95% CI [69.8, 88.2]) over null-hypothesis statistical testing and almost never (< 5%) report confidence intervals or effect sizes. Of the 43%, 95% CI [38.0, 48.0], that used statistical testing, a strong majority (> 80%) do not discuss underlying assumptions, control for family-wise Type 1 error rate or provide exact test statistics. These findings reveal a fundamental disparity between current practices and established standards as well as the pressing need for improving statistical rigor in metaheuristics research.

Keywords

Metaheuristics, methodology, statistics, replication crisis.

5.1 Introduction

Metaheuristics research (MR) studies stochastic optimization algorithms known as metaheuristics, which have become a popular object of study in recent years. These algorithms are efficient and can be faster than traditional methods in applied mathematics

and operations research. However, the trade-off is that they do not always produce the same result for a given problem and are difficult to analyze.

A common method for assessing the performance of metaheuristics algorithms is to repeatedly run them on a set of benchmark problems, known as a test set. The results, such as the best solutions found, are then collected, and analyzed in the form of statistical distributions. The performances of algorithms A and B are then compared through statistical analysis to determine which one is superior if any. Another common approach is to compare an algorithm to a modified version of itself to determine if the modifications meaningfully improve performance. The definition of "performance" varies between studies but usually refers to the number of evaluations of objective functions, hit rates, computational effort, or a combination.

Research studies in medicine, psychology, and pharmacology must adhere to established standards, like those set by the APA (American Psychological Association, 2020) and AMA (American Medical Association, 2020). However, in the field of metaheuristics, there is a lack of consistency in research methods, which can lead to a lack of rigor and uncertainty in results (Glover, 1977).

Critics have called for reform in the field of metaheuristics research (Sörensen, 2015; Hooker, 1995; Boussaïd et al., 2013; Barr et al., 1995), as many studies rely heavily on metaphors and ambiguous definitions instead of thorough analysis and fundamental principles. Some journals have started to revise their publication guidelines in response to this criticism, but it is still hypothesized that researchers in the field are not consistently using best practices when it comes to statistical methods.

This study investigates methodological gaps regarding the use of statistical methods in MR by analyzing a random sample of recently published peer-reviewed articles. The random selection process consists of a simple Python script which scrapes Google Scholar (GS) metadata for article titles, author names and URL in the following format:

TITLE : Metaheuristics for the team orienteering problem

AUTHORS : C Archetti, A Hertz, MG Speranza - Journal of Heuristics, 2007 - Springer
LINK : https://link.springer.com/article/10.1007/s10732-006-9004-0

The returned articles are randomly selected from the large volume of results returned by GS. The original code can be found in an executable online notebook at the following address:

https://github.com/iannickgagnon/random_article_selector

The articles are assessed using a quantitative evaluation framework based on standard statistical practices. The findings from this study will help to narrow the focus on the methodological aspects in MR that require attention.

This article is structured as follows. Section 5.2 presents the selection process for the identification of the MR articles to be assessed, and the criteria used to quantify the current practices. Section 0 synthesizes the coverage rates for the proposed best practices and Section 5.4 presents a summary of the findings, recommendations, and future research avenues.

5.2 Methodology

Analysis of all MR literature would take an impractical amount of time, with thousands to hundreds of thousands of articles to analyze. Therefore, for practical reasons, we randomly selected 70 peer-reviewed articles published between 2016 and 2021, with a confidence level (CL) of 95% ($z = 1.96$). Using Cochran's formula with $\hat{p} = 0.5$ (i.e. the most conservative value), we obtain an 11.7% margin of error (ε):

$$\varepsilon = \sqrt{\frac{z^2 \hat{p}(1-\hat{p})}{n}} = \sqrt{\frac{1.96^2 \cdot 0.5 \cdot (1-0.5)}{70}} = 11.7\% \tag{5.1}$$

The results (x_i) presented in Section 3 can therefore be read as $x_i \pm \varepsilon_i$ clamped between 0% and 100%. This study employed the evaluation process shown in Figure 5.1 and the data were generated using the iterative process illustrated in Figure 5.2.

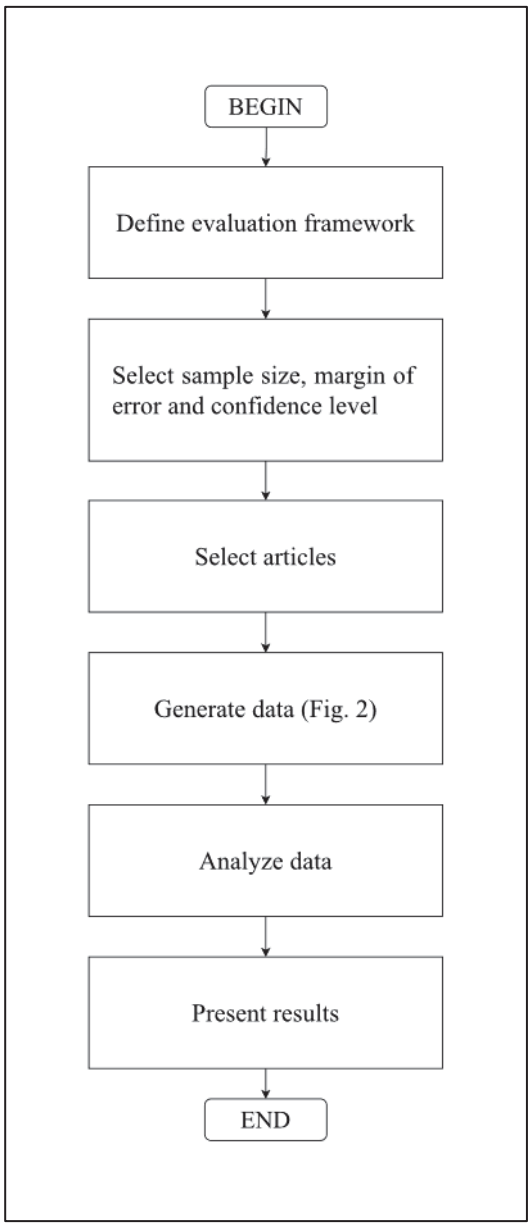

Figure 5.1 Evaluation process

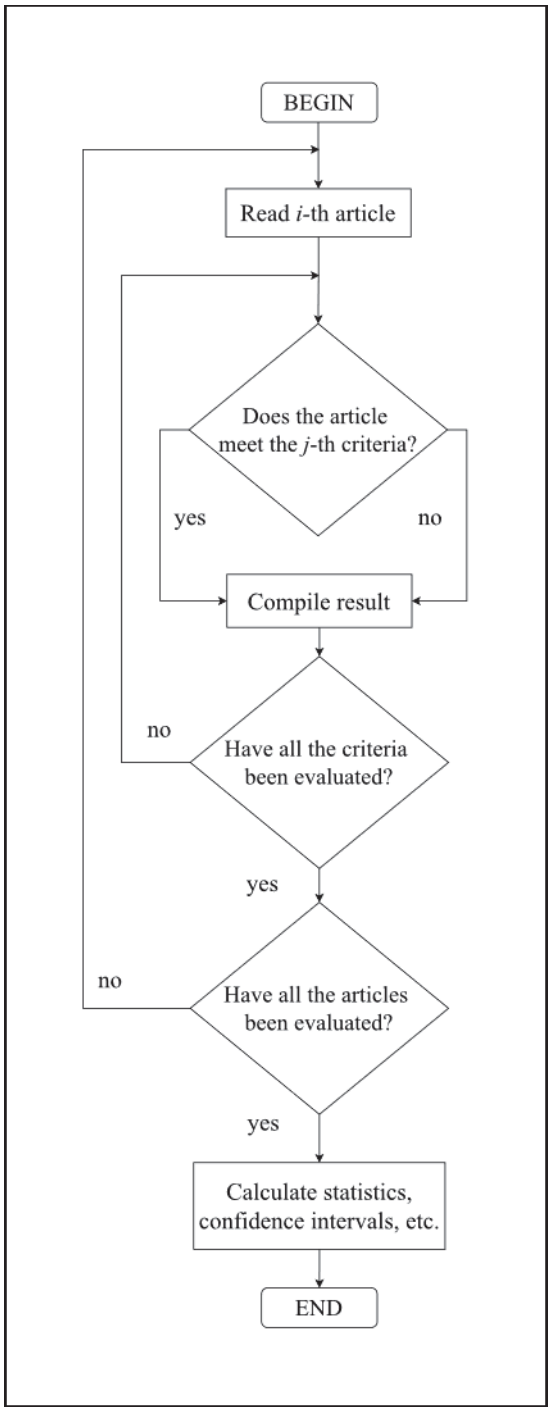

Figure 5.2 Data generation process

The following subsections present the criteria used to assess the sampled articles and explain their relevance. The criteria are divided into two categories (A and B) with the latter

containing those related to null-hypothesis statistical testing and the former containing "everything else" from descriptive statistics to measures of computational effort.

Table 5.1 Criteria for descriptive statistics

Identifier	Criterion
A.1	Is the parameter setting methodology documented?
A.2	Is a measure of central tendency provided?
A.3	Is a measure of variability provided?
A.4	Is a measure of symmetry provided?
A.5	Is a measure of tailedness provided?
A.6	Are confidence intervals provided?
A.7	Are the performance distributions plotted?
A.8	Are sample sizes adequate?
A.9	Are sample sizes equal?
A.10	Is computation effort discussed?
A.11	Is the source code freely available?

The first category of criteria concerns descriptive statistics, confidence intervals, use of graphs, computational effort, and the availability of source code. These criteria were grouped for convenience, not because they were related, although some were related.

(A.1) Fairness of comparisons – Is the parameter setting methodology documented

Parameters are known to have a significant influence on the behavior and, thus, the performance of a metaheuristic. Therefore, it is essential that the parameter selection process be reported in every research article and that parameter-tuning strategies be clearly explained. The ideal setting for fair comparisons is to compare algorithms at their best, but this is an unrealistic goal given that we do not and cannot know what the so-called "best" parameter set is. The next best approach is to use an automated parameter-tuning application such as ParamILS (Hutter, Hoos, Leyton-Brown & Stützle, 2009), which removes most of the guesswork and minimizes experimenter bias. This approach is more complex because it

involves bridging two computer programs (i.e., the metaheuristic and the parameter tuning program) by means of custom software adapters. The next best approach is to make an honest documented effort through experimental design or, at the very least, through some form of guided exploration of the parameter space, such as grid search.

(A.2) Central tendency – Is a measure of central tendency provided

Measures of central tendency included the mean ($\bar{x}$) and median ($\tilde{x}$). The mean is the most popular of the two and is the basis of most statistical procedures. It tells the reader what value to expect from a generating process when sampling from it. The median has the advantage of being insensitive to extreme values, which may or may not be useful, depending on the context. When outliers are present, it is recommended to include the median to account for their effect.

(A.3) Variability – Is a measure of variability provided

Measures of variability include the variance (s^2) and standard deviation (s), with the latter equal to the square root of the former. It informs the reader about the variability of the data *relative to the mean* or how far each observation is from the mean on average. Data with a low standard deviation tend to cluster around the mean, whereas data with a high standard deviation are more scattered. One practical implication of the standard deviation is that it tells the reader how reliable the generating process (i.e., the algorithm), where lower values represent greater reliability.

(A.4) Symmetry – Is a measure of symmetry provided

Symmetry is commonly defined as the degree of skewness. It tells the reader if the distribution looks the same on the left and right sides of the center point and indicates directionality (i.e., left, or right skew). Reporting skewness can be useful because performance distributions are often right skewed given that performance indicators are often strictly positive (e.g., CPU time cannot be negative). In short, the distributions encountered in MR are rarely symmetric, and skewness provides valuable information.

(A.5) Tailedness – Is a measure of tailedness provided

The tailedness of a distribution is commonly measured using kurtosis (κ). It informs the reader about the importance of the tails of the distribution relative to its center, which in turn points to the presence of outliers or the propensity of the generating process to produce them. There is a direct relationship between kurtosis and convergence because algorithms with high kurtosis have weak convergence properties.

(A.6) Uncertainty – Are confidence intervals provided

A confidence interval (CI) informs the reader about the stability of the sample statistics. It also provides a range of values that are likely to appear when we replicate the experiments.

(A.7) Graphs – Are the performance distributions plotted

To quote the classical statistical text *Facts From Figures*, Moroney (1956) wrote, "Better still, go further and present a histogram of the distribution [...]". Even if presenting a graph is not a substitute for reporting the mean directly, it is a worthy supplement. A simple histogram allows readers to gauge most of the descriptive statistics mentioned above (i.e., the central tendency and shape). This study did not differentiate between histograms, boxplots, or other types of representations. An article meets this criterion if any one of these is present.

(A.8) Generalizability – Are sample sizes adequate

Small sample sizes result in larger margins of error for parameter estimates, decreased statistical power, and increased Type I error rates, all of which are undesirable. The problem lies in the definition of "small" and we did not attempt to answer this question here. Many textbooks on statistics use $n = 30 \times m$ with m representing the number of independent variables. If the sample size in a selected study is greater than or equal to 30, the article meets the criterion. While the gold standard is to calculate the sample size based on fixed statistical power and effect size, there are often no neat formulas available, and simulation-based approaches may be outside the reach of many researchers for various reasons.

(A.9) Heterogeneity – Are sample sizes equal

The quality of the sample statistics that NHST relies on is a function of the sample size. It is generally desirable to use equal sample sizes because they influence the statistical power and type I error rate. When two samples had different sizes, the quality of their parameter estimates also differed, resulting in less reliable NHST conclusions. This is reflected in the test assumptions, such as those of equal variances for ANOVA. Although some statistical tests are designed specifically for situations where sample sizes are unequal, the use of equal sample sizes is still good practice.

(A.10) Effort – Is computation effort discussed

Assuming adequate parameterization, algorithm A should find a better solution than algorithm B if they are given enough time to do so. The more time it takes, the more likely it is to be true. Not only is this unfair and unscientific, but it also undermines the practical importance of the findings since "doing better" and requiring more time may be antonymous depending on the magnitude of the time difference. To address this issue, it is vital to contrast computational efforts in one way or another (e.g., CPU time and number of objective function evaluations).

(A.11) Replicability – Is the source code freely available

Reproducibility of experiments is one of the foundational principles of the scientific method. This can only be achieved by making the source code publicly available, and by packaging and documenting it in a manner that makes it sufficiently easy to install and run.

Table 5.2 Criteria for null-hypothesis statistical testing

Identifier	Criterion
B.1	Is null hypothesis statistical testing performed?
B.2	Is power analysis done a priori?
B.3	Are the test assumptions discussed and/or verified?
B.4	Is the significance level (α) provided?
B.5	Is the familywise Type I error rate controlled?
B.6	Are exact test statistics provided?
B.7	Are exact p-values provided?
B.8	Are statistically non-significant results discussed?
B.9	Are effect sizes provided?

Category B concerns the use of NHST and the various procedures that accompany it before, during, and after experiments are conducted.

(B.1) Confidence – Is null hypothesis statistical testing performed

Despite being widely criticized, the NHST remains the gold standard for determining whether an outcome is likely to be caused by natural variability or chance. Again, the aim here is not to settle any debate surrounding the practice but rather to quantify the coverage of tools and techniques used by the MR community.

(B.2) Power – Is power analysis done a priori

When non-significant results are observed, who is to say that the adverse results were not lurking somewhere in the mist of probabilistic uncertainty? It is never known how *likely* it is unless *a priori* power analysis is performed. Power analysis refers to the process of calculating the minimum sample size required to have a fixed percentage chance of correctly detecting the effect of a specific size when there is one. Not detecting an effect when there is one is called a type II error. As mentioned above, this should be done prior to running the experiments and should not be used to justify not detecting an effect after the fact. As

previously mentioned, this can be achieved using formulas or by simulation if closed-form expressions do not exist.

(B.3) Assumptions – Are the test assumptions discussed and/or verified

Parametric and non-parametric statistical tests are not free of assumptions. This is often confused with *distribution free* which means that a procedure does not assume a specific probability distribution function, but it is not free of assumptions altogether. Meeting these assumptions is a prerequisite for the validity of conclusions reached using NHST. Even though some tests, such as the ubiquitous Student t's, are robust to departures from some of its assumptions, it is still a good practice to not take anything for granted and perform basic checks. This is especially relevant for articles that use non-parametric tests without validating normality formally (e.g., distribution test) or informally (e.g., visual check) if the assumptions for a more powerful (in the statistical sense) parametric test are met.

(B.4) Significance – Is the significance level (α) provided

The significance level represents the probability that a test will detect an effect when there is none. This is called a type-I error. The importance of it being reported in research articles is considered self-evident.

(B.5) Tolerance to error – Is the familywise type I error rate controlled

Often, omnibus tests are used to compare $k > 2$ samples. Such tests indicate if *at least one* sample is different from the others, but do not say how many or which one(s). Pairwise comparisons are conducted when significant differences are detected at the group level, but omnibus tests are designed to avoid this pitfall in the first place, so it would be counterproductive. For example, if there are $k = 5$ samples, $C_2^5 = 5!/2!\,(5-2)! = 10$ two-sample tests should be performed.

For $\alpha = 0.05$, this increases the chance of a Type I error to:

$$\begin{aligned} \mathbb{P}(\text{at least one sig. result}) &= 1 - \mathbb{P}(\text{no sig. results}) \\ &= 1 - (1-\alpha)^{C_2^k} \\ &= 1 - (1-0.05)^{10} \\ &\approx 0.40 \end{aligned} \tag{5.2}$$

In this case, the value is eight times the per-comparison rate of 0.05, which is clearly not the intent of any researcher. The usual approach consists in adjusting the per comparison significance level (α_{PC}) so that the overall (sometimes called *familywise*) Type I error rate (α_{FW}) is as close to α_{PC} as possible. Such procedures include the Bonferroni correction, which straightforwardly divides the per-comparison rate by the number of groups (i.e., $\alpha = \alpha_{PC}/k$). Some consider this approach to be too conservative, but there are many alternatives, such as Sheffé's and Tukey's tests, that mitigate this drawback.

(B.6) Interpretation – Are exact test statistics provided

Publication manuals (e.g., APA and AMA's publication manuals) typically advise reporting exact test statistics (e.g., t, χ^2, etc.) to facilitate understanding and enable independent verification.

(B.7) Interpretation – Are exact p-values provided

Publication manuals (e.g., APA and AMA's publication manuals) typically advise reporting exact p-values instead of using the form $p < 0.10$, $p < 0.05$, $p < 0.01$. This indicates the likelihood of obtaining a result that is as extreme as or more extreme than the observed value. This information helps readers fully understand the results and forms the basis of some meta-analysis techniques. As with B.6, the exact value enables independent verification.

(B.8) Thoroughness – Are statistically non-significant results discussed

The APA considers it insufficient to declare that some differences are statistically significant without also analyzing statistically non-significant results and labels it as "hiding by

omission". For example, if Algorithm A outperforms Algorithm B on n out of m benchmarks, it is difficult to conclude anything meaningful because the test set is a subset of an infinite set.

(B.9) Practical importance – Are effect sizes provided

Given sufficiently large sample sizes, the NHST will *always* detect *some* effect; however, it may be small and practically irrelevant. Therefore, it is imperative to quantify the strength and magnitude of this effect by reporting effect sizes.

5.3 Results and discussion

This section presents the obtained coverage results for MR statistical criteria A and B along with immediate summary discussions for each.

A criterion can either be met, not met, or not applicable to a given paper. If a criterion is met, it is considered as covered, and coverage refers to the percentage of articles that adhere to that criterion. The coverage rates for each criterion are presented in a graphical format (Figure 5.3), with higher percentages being viewed as better.

Figure 5.3 Compilation of coverage criteria in Category A

(A.1) Fairness – Is the parameter setting methodology provided

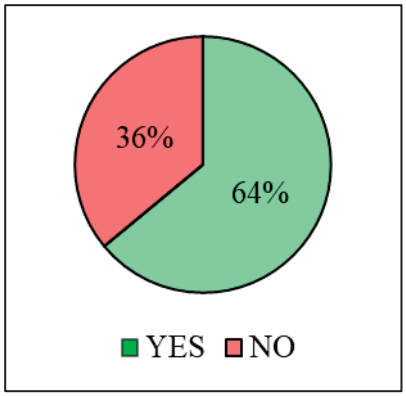

Figure 5.4 Coverage for criterion A.1

About two thirds (64%, 95% CI [56.5, 71.5]) of surveyed articles documented the parameter setting and tuning processes. Authors primarily relied on values found in the literature or some undocumented process referred to as "sensitivity analysis". A minority (6%, 95% CI [5.3, 6.7]) used automated parameter-tuning frameworks, such as OptQuest, ParamILS, and irace. Other methods include design of experiments (DOE), one-at-a-time parameter tuning, and grid search.

Relying on values found in the literature is a good starting point, but it is far from optimal, given that the test functions generally differ from source references. The use of ill-defined "sensitivity analyses" is less useful since it is undocumented. Moreover, these studies did not use *quantitative* sensitivity analysis, which undermines their credibility. It is also worth pointing out the irony of using one-at-a-time parameter tuning, given that it almost surely leads to local optima and that avoiding this is the *raison d'être* of metaheuristics.

On a positive note, these figures indicate that a significant number of researchers are concerned with selecting the appropriate parameters. This suggests that with proper guidance and resources, researchers would likely use more advanced techniques, such as automated parameter tuning, design of experiments, or even a basic grid search.

(A.2) Central tendency – Is a measure of central tendency provided?

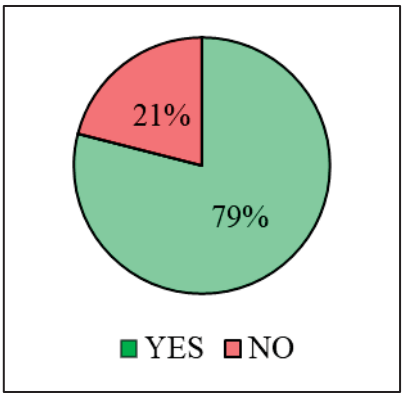

Figure 5.5 Coverage for criterion A.2

Four out of five (79%, 95% CI [69.8, 88.2]) articles report at least one measure of central tendency. All of them (100%, 95% CI [88.3, 100.0]) used the mean, and a majority (66%, 95% CI [58.3, 73.7]) also reported the median. This, again, is a good start, but since performance distributions are often right skewed for the reasons mentioned in the previous section, the median may be a better indicator of central tendency (Hatcher, 2013; Mackridge & Rowe, 2018). Of the 55 participants who reported the mean, only one (< 2%, 95% CI [1.8, 2.2]) also provided a confidence interval.

In addition to the lack of confidence intervals (A.6), the findings also reveal that a significant proportion of articles (21%, 95% CI [18.5, 23.5]) reached their conclusions based on the highest performance achieved. There is no justification for this practice as pointed out in (Birattari & Dorigo, 2007).

(A.3) Spread – Is a measure of variability provided

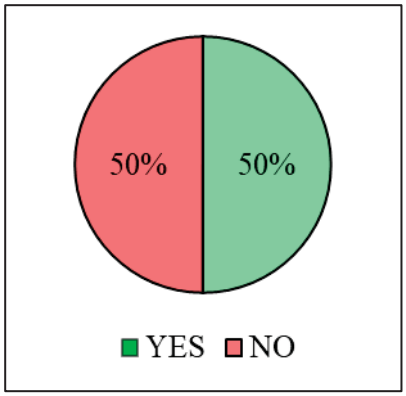

Figure 5.6 Coverage for criterion A.3

Half (50%, 95% CI [44.1, 55.9]) of the articles contained at least one measure of variability. All but one used the standard deviation, with the latter using interquartile range (IQR). None of the articles included confidence interval. The decrease from 64% CI [56.5, 71.5] to 50% CI [44.1, 55.9] between the reporting of measures of central tendency and measures of variability indicates that the authors may be less comfortable with the latter. Other studies like (Matthews & Clark, 2007) have documented this in groups of undergraduate students, of which a subset goes on to graduate and write research articles. It is sometimes argued that reporting the standard deviation for a skewed distribution is misleading, but this is inaccurate. Pick almost *any* distribution, and approximately 99% of all observations fall within $\mu \pm 3\sigma$.

(A.4) Symmetry – Is a measure of symmetry provided

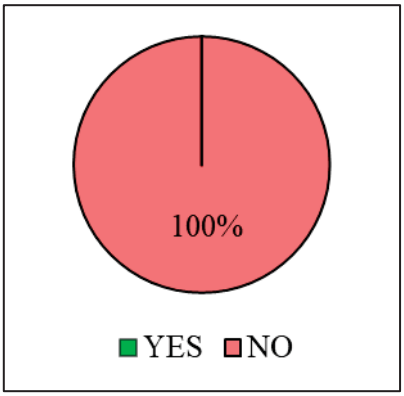

Figure 5.7 Coverage for criterion A.4

None of the surveyed articles report measures of symmetry. Possible remedies that would also serve other purposes are to provide histograms (criterion A.7) and/or report both the mean and the median (Hatcher, 2013) (criterion A.2).

(A.5) Tailedness – Is a measure of tailedness provided

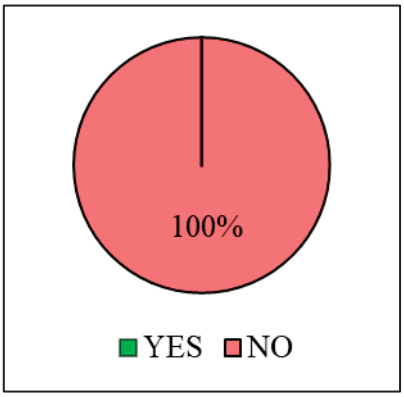

Figure 5.8 Coverage for criterion A.5

None of the surveyed articles report measures of tailedness. Thus, the reader cannot determine the probability of an algorithm failing to converge in the allocated time. Since run times are often fixed at launch on cluster machines, this finding is of notable practical importance. The same conclusions and recommendations as in Criterion A.4 apply to this criterion.

(A.6) Uncertainty – Are confidence intervals provided

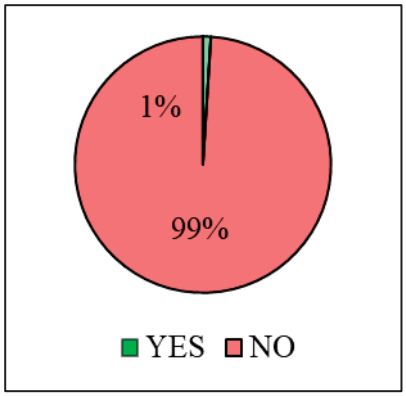

Figure 5.9 Coverage for criterion A.6

None (1%, 95% CI [0.9, 1.1]) of the surveyed articles reports confidence intervals. This low percentage is both unexpected and worrying. It suggests that results are presented as if they were exact, which is misleading and false. This problem can be resolved simply by having journals implement policies that make confidence intervals mandatory such as Nature's journals for example.

(A.7) Visuals – Are the performance distributions plotted

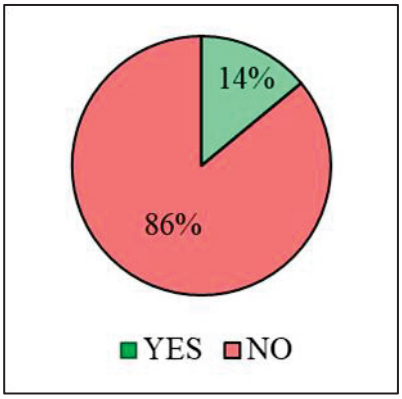

Figure 5.10 Coverage for criterion A.7

About one of every six (14%, 95% CI [12.4, 15.6]) papers provided a visual representation of the results in the form of histograms, box plots, or other graphs. The fact that graphs are

scarcely presented to the reader is aggravated by the absence of shape parameters (see criteria A.4 and A.5).

Those that reported graphs used box plots and/or histograms. The box plots did not contain error bars, adding to the conclusions reached for Criterion A.6 regarding precision.

(A.8) Generalizability – Are sample sizes adequate

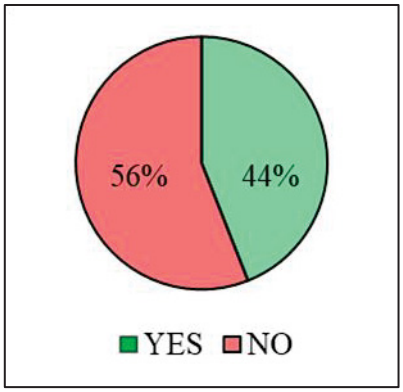

Figure 5.11 Coverage for criterion A.8

More than half (56%, 95% CI [49.4, 62.6]) of the articles used inadequate sample sizes and 7%, 95% CI [6.2, 7.8], did not report them. For those that failed this criterion, the most common sample sizes were 1, 5 and 10, whereas for those who passed, 30 and 50 were the most common.

For the surveyed articles, it is usually possible to increase sample sizes by one order of magnitude or more with little consequences on resource consumption.

(A.9) Heterogeneity – Are sample sizes equal

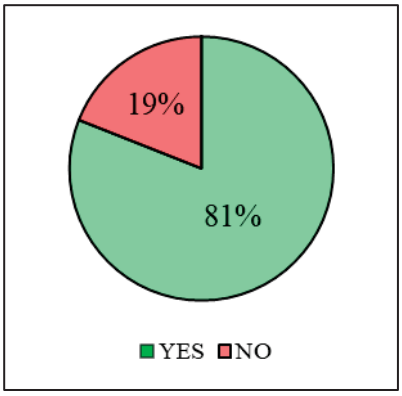

Figure 5.12 Coverage for criterion A.9

The sample sizes across the groups were mostly equal (81%, 95% CI [71.5, 90.5]). For the 19%, 95% CI [16.8, 21.2], that were not, an average difference of 33%, 95% CI [24.8, 41.2], in the relative group sizes was observed. Using unequal sample sizes decreases the power of statistical tests, making it less likely to detect a true difference between the samples. This reduced the credibility of the results. However, using non-parametric tests (criterion B.1) helps mitigate this issue.

(A.10) Fairness – Is computation effort discussed

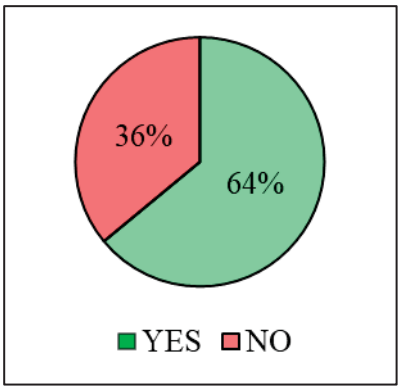

Figure 5.13 Coverage for criterion A.10

About two thirds (64%, 95% CI [56.5, 71.5]) of articles discuss computational effort, mostly in terms of CPU time. This is a positive development, but CPU time is a hardware- and

instance-dependent measure that should be supplemented with the number of objective function evaluations. Reporting both allows for a comprehensive understanding of the algorithm's performance profile. For example, an algorithm with fewer objective function evaluations may be less efficient than another if it introduces computationally expensive overhead measured in CPU time. Conversely, for CPU intensive problems, an algorithm with low CPU time on a set of benchmarks may be less efficient than another if it does significantly more objective function evaluations.

(A.11) Replicability – Is the source code freely available

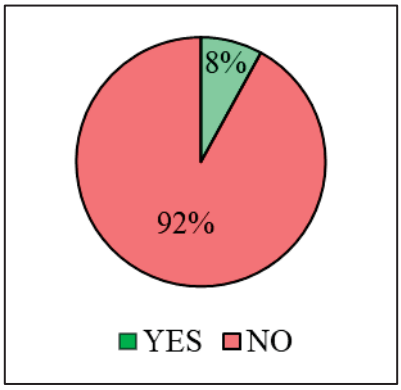

Figure 5.14 Coverage for criterion A.11

Less than one out of ten (8%, 95% CI [7.1, 8.9]) research articles included source code. This aligns with the trend observed in the field of computational science as a whole (Yale Law School Roundtable on Data and Code Sharing, 2010). This problem can also be fixed at the journal level by developing policies and infrastructure (e.g. online repositories) that promote the sharing of source code. Financial considerations may make this proposal more difficult to implement, but a business case certainly exists. For example, eminent funding agencies may decide that sharing source code is a prerequisite for financial backing and having the required infrastructure would put early investors at an immediate competitive advantage.

The rest of this section presents the obtained coverage results for criteria in category B.

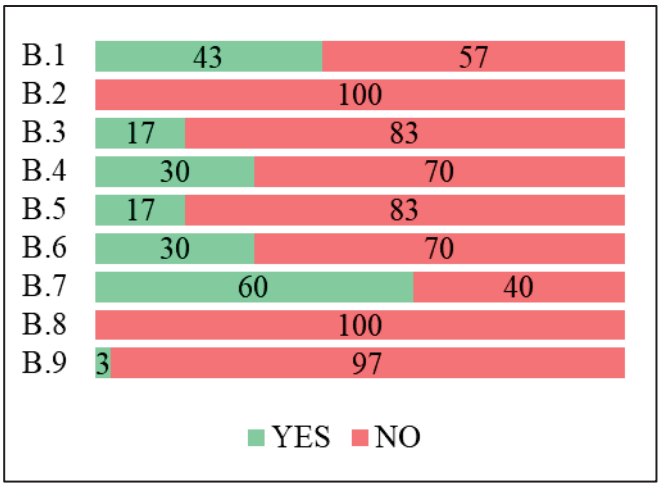

Figure 5.15 Compilation of coverage of criteria in category B

(B.1) Confidence – Is null hypothesis statistical testing performed

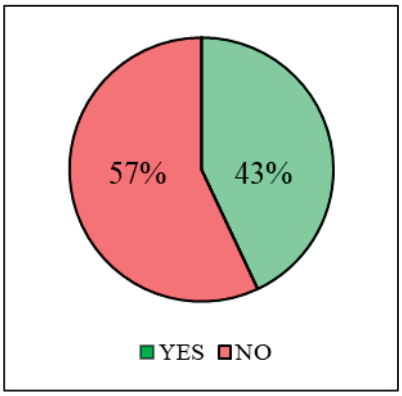

Figure 5.16 Coverage for criterion B.1

About two out of five articles used NHST to validate their results. Therefore, 43%, 95% CI [38.0, 48.0], of the researchers are aware that their results may be due to chance. The remaining 57%, 95% CI [50.3, 63.7], may be unaware of this reality because they rely on descriptive statistics, such as the differences in means, without formally checking that it is statistically different from zero.

Of those who employed NHST, the most common procedures were the non-parametric Wilcoxon signed-rank test, Friedman test, and parametric t-test. Of the three, the Wilcoxon signed-rank test appears to be the most appropriate given that (1) it is non-parametric and

(2) it accounts for the magnitude of the difference between and not just the ranks, as Friedman's test does. The non-parametric part is emphasized because of the frequent absence of normality, which the Wilcoxon signed-rank test does not assume. This property results in greater statistical power than the t-test in such cases. Conversely, it underperforms the t-test when normality is not significantly violated.

(B.2) Power – Is power analysis done a priori

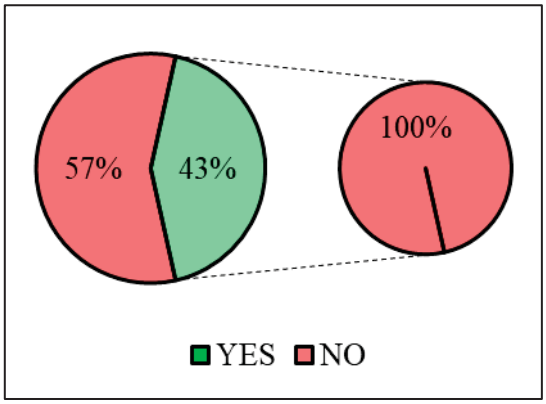

Figure 5.17 Coverage for criterion B.2

None of the surveyed articles performed a priori power analysis. This, along with the fact that most results are not statistically significant, suggests that there may be many important facts yet to be discovered that were simply unlikely to be observed given sample and effect sizes.

(B.3) Assumptions – Are the test assumptions discussed and/or verified

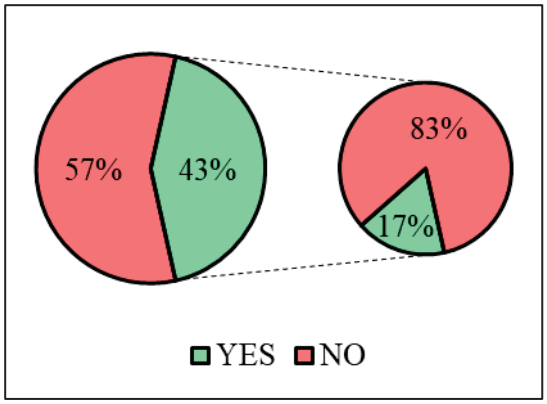

Figure 5.18 Coverage for criterion B.3

While discussion is no remedy for verification, mere acknowledgement of assumptions was sufficient to obtain full marks for this criterion. Slightly less than one out of every five (17%, 95% CI [15.0, 19.0]) articles discussed test assumptions, mostly to justify the use of non-parametric procedures, and two of them (7%, 95% CI [6.2, 7.8]) performed *a priori* normality tests.

This observation may have little effect on outcomes in practice because most researchers use non-parametric tests on distributions that can be safely assumed to be non-normal, but it is better to know for sure than to assume because proof of the contrary could allow researchers to use generally more powerful parametric alternatives.

(B.4) Significance – Is the significance level (α) provided

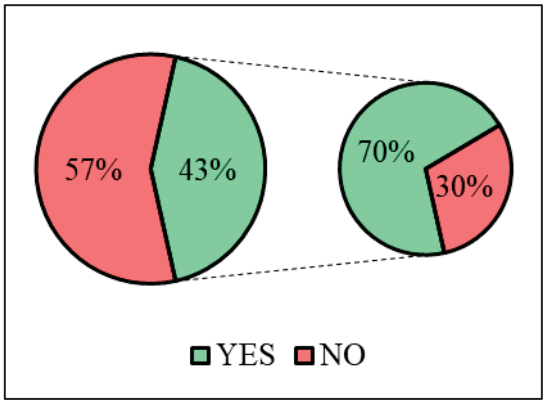

Figure 5.19 Coverage for criterion B.4

When the significance level is not explicitly stated, as in 30%, 95% CI [26.5, 33.5] of the surveyed articles, it becomes an act of faith to trust that the researchers used a sensible value such as 5% or 1%.

(B.5) Tolerance to error – Is the familywise Type I error rate controlled

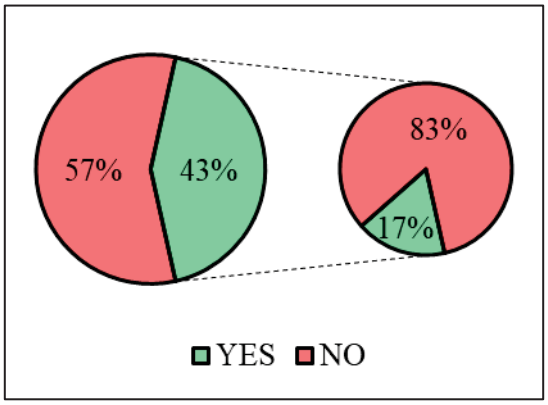

Figure 5.20 Coverage for criterion B.5

About four out of five (17%, 95% CI [15.0, 19.0]) studies that employed NHST did not control the Type I error rate for multiple comparisons, which increases the chances of false positives, in some cases by a lot, therefore invalidating the conclusions. Of those that did control Type I error rate, the Holm and Hochberg tests were the most used followed by the Nemenyi post-hoc test and Sheffé's.

(B.6) Interpretation – Are exact test statistics provided

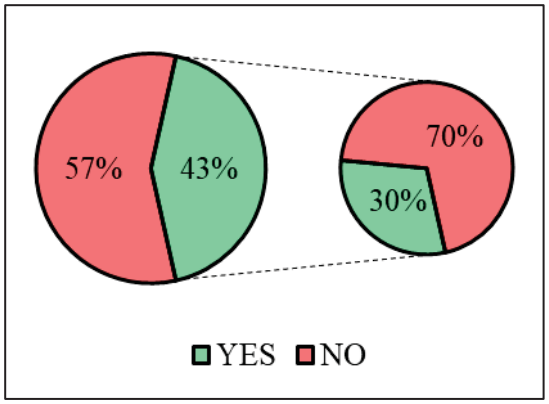

Figure 5.21 Coverage for criterion B.6

In 70%, 95% CI [61.8, 78.2] of cases, researchers do not report exact test statistics, so the rest of the research community can compare findings on their own. As with the reporting of confidence levels, this is also a low effort but necessary task.

(B.7) Interpretation – Are exact p-values provided

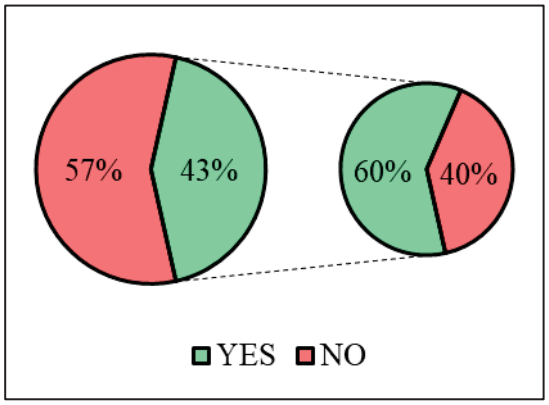

Figure 5.22 Coverage for criterion B.7

The p-values were reported more rigorously than the test statistics at a rate of 60%, 95% CI [53.0, 67.0]. These results are better than those reported in Figure 5.15 concerning the test statistics; however, 40%, 95% CI [35.3, 44.7], of the published works are still incomplete. When the reader is not aware of the exact p-value, it must be assumed that it is lower than an acceptable confidence level.

(B.8) Thoroughness – Are statistically non-significant results discussed

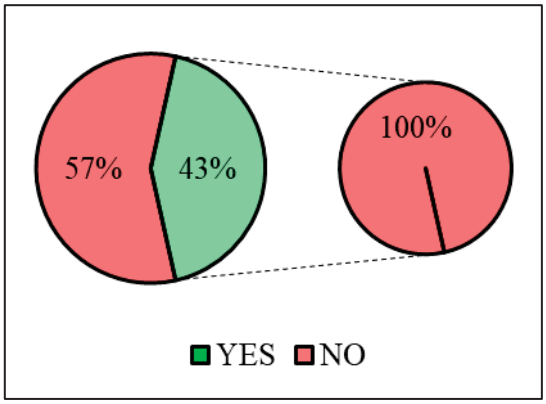

Figure 5.23 Coverage for criterion B.8

Most results were not statistically significant, yet they were never discussed in any of the articles. The usual protocol is to select many test functions and only consider those instances where the proposed algorithm outperforms the others in a statistically meaningful way and ignore the rest. Even if done unconsciously and without malice, this is little better than cherry picking. One counterargument could be the "no free lunch theorem", which states that no single algorithm is optimal for all problem types. However, this argument would not hold as none of the articles referenced in the statement make any effort to identify the specific types of problems for which the proposed algorithms are superior to their competitors.

(B.9) Practical importance – Are effect sizes provided

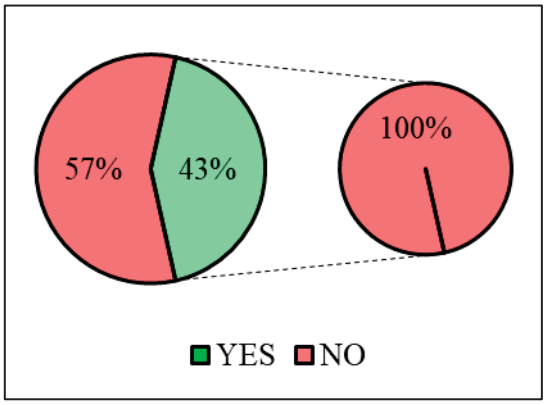

Figure 5.24 Coverage for criterion B.9

Only two articles (3%, 95% CI [2.6, 3.4]) reported effect sizes, without truly calculating it, whereas the rest ignored it entirely. As such, no attention is paid to the practical relevance of statistically significant differences.

5.4 Conclusion

Methodological practices employed by the MR community have been criticized in several ways (Sörensen, 2015; Hooker, 1995; Boussaïd et al., 2013; Barr & al., 1995). This study examined the line of criticism related to the use of statistical methods. A set of 70 randomly selected research articles were evaluated using a proposed quantitative evaluation framework based on established statistical practices in more mature scientific fields such as medicine, pharmacology, and psychology. It is easy to understand, simple to implement, and does not require much effort on the part of the user. Additionally, it can be used to determine to what extent statistical best practices are used in other areas of research, such as computer science, software engineering, and other fields.

The core elements of scientific research, namely significance, confidence, and effect size, are never simultaneously present in any of the 70 surveyed articles. Individually, statistical confidence testing was present 43%, 95% CI [38.0, 48.0] while confidence intervals and effect size, a mere 1%, 95% CI [0.9, 1.1], and 3%, 95% CI [2.6, 3.4], of the time, respectively. It is encouraging to see that this problem has been successfully resolved in psychology as reported recently in (Haynes, Mulrow, Huth, Altman & Gardner, 1990), which states that the field achieves near-perfect scores on all three measures. This is something that can also be accomplished by the MR community.

The lack of availability of source code in MR research (92%, 95% CI [81.2, 100.0]) is a significant issue that could be easily addressed. This lack of access to source code may be a contributing factor to the rarity of replication studies in the field (Swan et al., 2022). This undermines the credibility of the field as a whole and could be tackled head-on by journals, universities and funding agencies following the suggestions made in (Yale Law School Roundtable on Data and Code Sharing, 2010).

It is important to note that the criteria outlined in our framework are not inherently challenging to fulfill, and we believe they should be universally adopted. However, the specific criteria outlined here are believed to yield the greatest benefit.

The evaluation framework we have developed can currently serve as a preliminary assessment tool for both researchers and reviewers in the field of MR. Further research should concentrate on examining the writing techniques utilized in published MR articles to pinpoint areas that require the most attention. A compilation of the most effective methodological practices, like the APA's publication manual but specifically tailored for MR, that is openly available, is needed.

CONCLUSION

This thesis addresses longstanding methodological deficiencies in metaheuristics research by combining critical empirical analyses with a prescriptive methodological framework. Through three complementary peer-reviewed studies, it documents recurring weaknesses in the design, analysis, and reporting of metaheuristics experiments, and uses these findings to motivate and support the development of the MDAF.

It shows that many limitations observed in the literature are not due to the inherent difficulty of optimization problems, but rather to avoidable shortcomings in scientific methodology. These include unclear experimental design choices, insufficient statistical analyses, limited reporting transparency, and practices that hinder reproducibility and interpretation. By examining these issues from different angles, it provides converging evidence that methodological rigor remains a central challenge for the maturation of the field.

Beyond criticism, this thesis makes a constructive contribution by proposing the MDAF as an integrated framework to guide researchers toward more rigorous and transparent empirical studies. The framework is grounded in established principles of experimental design, statistical analysis, and scientific reporting, while remaining adapted to the specific realities of metaheuristics research.

Overall, the results support the view that improving the scientific quality of metaheuristics research does not require additional complexity, but rather more disciplined research methods. By strengthening methodological foundations, the field can improve its credibility and highlight the cumulative value of its results.

APPENDIX I

THE MDAF

In this appendix, the MDAF is introduced as a response to methodological shortcomings. The framework is intended to promote greater rigor, transparency, reproducibility, and overall quality of research in the field of metaheuristics.

I.1 Overview

The MDAF provides an integrated methodological structure that assists researchers with problem definition and conceptual planning through implementation, execution, and the subsequent analysis of findings.

Its guidelines are a set of best practices that guide researchers through the complete lifecycle of metaheuristic research, whether they are for developing a novel algorithm, modifying an existing one, or simply comparing algorithms. The guidelines span four critical dimensions:

1. **Design**	Structuring experiments to ensure reliability, validity, and generalizability.
2. **Implementation**	Promoting modular and maintainable.
3. **Analysis**	Applying appropriate statistical techniques to validate results and draw meaningful insights.
4. **Reporting**	Presenting results with clarity, completeness, and contextual relevance to support evaluation and interpretation.

The following section elaborates on the four dimensions: Design in section I.3, Implementation in section I.4, and Analysis and reporting in section I.5. Before addressing these, it is essential to first consider a foundational prerequisite: the admissibility of the research itself. Engaging in discussions on how to design or analyze an experiment is of limited value if said study fails to meet basic standards.

To this end, MDAF introduces a set of Minimum Research Admissibility Criteria, presented in section I.2. These criteria draw on principles from patent law, treating algorithms as intellectual inventions that must meet defined conditions for publication. In the same way that patents must be novel, useful, and non-obvious, research should be held to similar standards.

I.2 Minimum research admissibility criteria

The motivations behind metaheuristic algorithm development generally fall under the following two broad categories:

1. To introduce a novel idea in the form of a new algorithm;
2. To introduce a novel idea in the form of a variant of a pre-existing algorithm.

Research is defined by the Merriam-Webster dictionary as the systematic investigation of a subject aimed at the discovery of facts, revision of accepted theories or laws in the light of new facts, or the practical application of such new or revised theories or laws. At the heart of this definition is the notion of novelty. Research is driven by the quest to uncover new facts, new interpretations of existing data, or new applications of theories. This pursuit of novelty distinguishes it from information gathering and routine problem-solving. Therefore, the first and most general guideline proposed by MDAF, which applies to both new algorithms and variants, is for researchers, reviewers and editors, to adopt a strict definition of admissibility.

For this, the MDAF finds its inspiration in patent law by considering proposed algorithms as inventions. According to article 54 of the European Patent Convention (EPC), an invention is novel if and only if it does not form part of the art. It defines the art or state of the art as anything that is readily available before the date of the patent filing. The U.S. Patent Act goes further by stating that patentability rests on the notions of novelty, usefulness and non-obviousness. Here they are in context:

A. **Novel:** For contributions that propose new algorithms or variants, the technical features must not be found in any prior public disclosure. For other types of

publications, such as systematic reviews, comparative studies, or replication efforts, novelty may lie in the synthesis of existing knowledge, methodological rigor, or the generation of new insights through reanalysis or reinterpretation.

B. **Useful:** For contributions that propose new algorithms or variants, the work should demonstrate practical utility by solving one or more specific and identifiable problems, either in applied contexts or within the methodological development of metaheuristics. In other types of research, usefulness may be reflected in the clarification of existing knowledge, the identification of gaps, or the provision of guidance for future research and practice.

C. **Nonobvious:** For contributions proposing new algorithms or variants, the approach should not be obvious to someone with ordinary skill in the art, nor should it result from a straightforward combination of known techniques or prior work. The contribution should exhibit a degree of ingenuity, whether through a novel mechanism, an unexpected improvement, or an innovative adaptation. In other types of research, non-obviousness may be reflected in the originality of the analytical perspective, the synthesis of disparate sources, or the formulation of novel hypotheses or frameworks that add conceptual or methodological value.

The MDAF endorses these criteria as guiding principles and suggests that research should ideally be considered admissible for publication when each is clearly addressed. The responsibility for demonstrating these qualities lies with the researcher, who is expected to provide sufficient justification and evidence in support of their claims.

In addition to the criteria inspired by patent law, the MDAF introduces the following research-specific criterion:

D. **Proven:** For contributions proposing new algorithms, the claimed usefulness should be supported by empirical evidence, including appropriate experimentation and statistical analysis. In other forms of research, proof may

> consist of a rigorous argumentation, reproducible analyses, or a well-substantiated synthesis of existing evidence.

The purpose of this criterion is to distinguish between claims and substantiated results. A proposed definition of what constitutes "adequate" in terms of statistical analysis is given in section I.5.

Under strict application of these criteria, publishing the same algorithm under different names or metaphorical descriptions would not be considered acceptable, as names and metaphors do not constitute technical features. As noted earlier, it remains the responsibility of the researcher to demonstrate that the proposed algorithm introduces distinct and substantive technical features, in accordance with Criterion A.

To be considered useful (Criterion B), a proposed algorithm must "work as claimed" which can be demonstrated through open access data and/or open access source code. The referenced data must meet the expectations of Criterion D. Further guidance on data and code sharing is provided in section I.4.1.

The obviousness clause (Criterion C) relates to the degree of difference from prior art. In other words, it should not be possible to publish works based on superficial changes. To use a *reductio ad absurdum*, consider the Simulated Annealing algorithm, which probabilistically selects nonoptimal moves by comparing a randomly generated value with $e^{\Delta Q/t}$, where Q represents fitness and t is a parameter called temperature. Under the current paradigm, it would not be surprising to encounter a "new" algorithm which replaces e with another mathematical constant, say π, published under a different name, for example, "Simulated Pie". In this hypothetical scenario, temperature would be derived not from the cooling schedule of the metallurgical process of annealing but from a baking schedule. While intentionally exaggerated, this example underscores the importance of the criterion. This concern is far from theoretical: empirical analyses have revealed that several published algorithms are, in fact, minor variants of pre-existing methods (Weyland, 2010, 2015), (Gagnon, April & Abran, 2020) and (Camacho-Villalón, Dorigo & Stützle, 2023).

The enabling effect of metaphors is not cited directly in the previous paragraph, but it is implicit. As such, the last criterion of admissibility addresses it directly:

E. **Metaphor-free:** For contributions proposing new algorithms, the method should be described using precise and technically appropriate language, free of metaphors that may obscure its underlying mechanisms. While metaphors can serve a pedagogical role, especially in introductory explanations or outreach, they should not substitute for formal definitions in scholarly work. In other types of research, technical clarity remains essential to ensure reproducibility, transparency, and meaningful evaluation.

It is important to note that Criterion E does not prohibit the use of metaphors but says that an adjacent metaphor-free description must also be present.

If an article meets all the criteria, it may be considered "receivable" in the sense that it can be formally acknowledged for publication. The MDAF defines criteria A to E as "hard" criteria. Ideally, each of these criteria should be satisfied and falling short on any one of them may raise serious concerns regarding the study's admissibility. The criteria are presented together in APPENDIX II.

In summary, the MDAF advocates for a rigorous and principled approach to determining the admissibility of contributions in metaheuristics research. By drawing on established standards from patent law and extending them with a research-specific emphasis on empirical validation and technical clarity, the proposed criteria (novelty, usefulness, non-obviousness, proof, and metaphor-free description) serve as a foundation for maintaining the integrity and relevance of published work. These criteria are not intended to raise the bar arbitrarily, but rather to ensure that research in the field contributes meaningfully to scientific knowledge and practice. When applied systematically by authors, reviewers, and editors alike, they can help prevent redundant publications, promote transparency, and foster genuine innovation.

I.3 Design

Experimental design is defined by the Merriam-Webster dictionary as a research method in which controlled experimental factors are subjected to special treatments for purposes of comparison with a factor kept constant. However, the definition used in this work is more general. It refers to the process of planning and structuring experiments to systematically address specific research questions or hypotheses.

Within metaheuristics research, a robust experimental design ensures the credibility, generalizability, and scientific value of the results. Rather than simply executing algorithm runs and reporting outcomes, it emphasizes methodological discipline: setting clear goals, controlling for variability, identifying meaningful factors, and ensuring repeatability.

A well-defined experimental design within the MDAF framework addresses several key elements, which are detailed in the subsections that follow.

Section I.3.1 presents detailed instructions for writing high-quality research questions, hypotheses and objectives. Section I.3.2 covers sample size selection. Section I.3.3 offers recommendations for ensuring fair comparisons between algorithms, followed by section I.3.4, which discusses the selection of benchmark functions and appropriate performance measures.

I.3.1 Research questions, variables, hypotheses and objectives

Research questions stem from a perceived knowledge deficit within a subject area or field of study (Haynes, 2006). The MDAF proposes a research question development framework in the form of a slightly modified version of the FINER framework (Browner et al., 2022) widely used in clinical research (Table-A I-1).

Table-A I-1 FINER criteria for research questions

F	Feasible	Manageable in scope.
I	Interesting	Obtaining an answer is intriguing to others.
N	Novel	Confirms, refutes or extends previous results.
E	Explicit	Clearly defined and unambiguous.
R	Relevant	Pertinent and significant to the field.

The FINER criteria are high level descriptors of good quality research questions. The MDAF also provides a lower-level structure adapted from the PICO criteria for writing specific and operational research questions, which is presented in Table-A I-2.

Table-A I-2 PICO criteria for specific research questions

P	Population	What specific algorithms or components are you interested in?
I	Intervention	What is your investigational intervention?
C	Comparison	What is the baseline used to compare with the intervention?
O	Outcome	What do you intend to accomplish, measure, improve or affect?

To illustrate how these frameworks can be applied in practice, consider the case of a knowledge gap regarding the influence of chaotic maps on metaheuristic performance. A preliminary question might be: “Do chaotic maps generally improve the performance of metaheuristics?” This question draws attention to an actual gap in the current state of knowledge, which makes it interesting and relevant, but its formulation does not fully meet the FINER criteria: it is vague (i.e., not fully *explicit*), it is overly broad, which raises *feasibility* concerns, and it lacks a clear comparison group or *outcome*.

By applying the PICO framework, the question can be made more precise. In this example:

- **Population (P):** Metaheuristic algorithms.
- **Intervention (I):** Incorporation of chaotic maps.
- **Comparison (C):** Use of traditional pseudorandom number generators.
- **Outcome (O):** Optimization performance.

A refined question might then be: “How does the performance of metaheuristic algorithms equipped with chaotic maps compare with that of the same metaheuristic algorithms equipped with traditional pseudorandom number generators?”

Once a high-quality research question is formulated, it becomes possible to identify key variables: the independent variables (IVs), dependent variables (DVs), and control variables (CVs). In the refined example:

- **Independent variable:** The presence of chaotic maps.
- **Dependent variable:** The performance of metaheuristic algorithms.
- **Control variable:** The algorithm structures and parameters outside of the chaotic maps.

The next step is to establish a testable hypothesis that articulates the relationship between the IV and the DV, informed by prior research. For instance, although chaotic maps are often associated with improved performance in the literature, the lack of rigorous theoretical foundations and appropriate statistical testing may prompt the formulation of a null hypothesis: “Metaheuristic algorithms equipped with chaotic maps do not generally perform better than the same metaheuristic algorithms equipped with traditional pseudorandom number generators.”

Finally, now that the research question and hypothesis are known, a primary research objective (i.e., one that is directly related to the research hypothesis) must be formulated as an active statement. For this, MDAF suggests using a modified version of the SMART criteria (Table-A I-3), which omits the time-bound dimension (T) and replaces Relevance (R) with Relatedness to the research question. The time-bound dimension is excluded

because it pertains more to project management than to research quality, while relevance is already addressed through the prior formulation of FINER-compliant research questions.

Table-A I-3 SMAR criteria for research objectives

S	Specific	Clearly defined and unambiguous.
M	Measurable	Quantifiable and able to be assessed.
A	Achievable	Realistic and attainable within the scope of the project.
R	Related	Directly linked to the original research question or hypothesis.

For example, an initial objective might read: “Investigate the impact of chaotic maps on the performance of metaheuristic algorithms.” However, this version lacks specificity (S) and measurability (M). A SMAR-compliant version might be: “Investigate the impact of chaotic maps on the convergence speed of selected metaheuristic algorithms using a predefined set of benchmark functions.” This revised version narrows the scope and ensures clearer assessment criteria.

When applicable, the MDAF also encourages the inclusion of secondary objectives, especially those intended to explore mechanisms underlying the observed effects. These objectives offer additional insight and enhance the comprehensiveness of the study. For instance, the researcher may hypothesize that statistically significant differences in performance are influenced by the order of random numbers generated. A corresponding secondary objective could be: “Explore the impact of random shuffling of chaotic sequences on the statistical significance of comparative metaheuristic performance results.”

Here is the proposed resulting sequence inspired by (Farrugia et al., 2010):

1. Identify a research gap (e.g., through a systematic literature review).
2. Develop a high-level research question using the FINER criteria (Table-A I-1).
3. Refine the question using the PICO criteria (Table-A I-2).

4. Identify independent, dependent and control variables.
5. Formulate a research hypothesis based on the available knowledge.
6. Develop clear primary and secondary objectives using the SMAR criteria (Table-A I-).

In summary**,** the MDAF provides a structured approach for formulating and refining research questions and hypotheses in metaheuristics. By leveraging established frameworks such as FINER (Table-A I-1) and PICO (Table-A I-2), it ensures that research questions are feasible, novel, explicit, and aligned with both scientific relevance and specificity. The framework emphasizes the importance of identifying meaningful variables and establishing well-reasoned hypotheses grounded in research and analysis.

To guide empirical investigations, the MDAF also introduces the SMAR criteria (Table-A I-), promoting objectives that are Specific, Measurable, Achievable, and Related to the initial question. This clarity enhances the alignment between research goals and experimental protocols.

Finally, the framework advocates for the use of secondary objectives to explore causal mechanisms and reinforce the robustness of results. The systematic application of these principles enables researchers to design experiments that are both scientifically sound and methodologically rigorous, thereby contributing to more impactful and credible research in the field of metaheuristics.

I.3.2 Sample size

In the context of metaheuristics research, performance-related data typically follows one of two distributional patterns: (1) approximately normal distributions, characterized by symmetry or a bell-shaped curve, and (2) right-skewed, strictly positive distributions. The former often emerges because of the Central Limit Theorem (CLT), which states that, under mild conditions, the distribution of sample means tends toward normality as the sample size increases, irrespective of the underlying population distribution. The latter category, right-skewed distributions, frequently results from the intrinsic properties of the data being

analyzed, particularly when values are bound below and exhibit a long tail toward higher values. A notable example is the distribution of first hitting times in stochastic processes, wherein most runs converge relatively quickly, while a non-negligible number requires substantially more time.

The determination of an appropriate sample size depends primarily on three parameters: the significance level (α), the Type II error rate (β), and the expected effect size (δ). The significance level (typically $\alpha = 0.05$) represents the probability of a Type I error, that is, rejecting the null hypothesis when it is in fact true. Its complement, $1 - \alpha$, corresponds to the confidence level, which reflects the degree of certainty that the confidence interval contains the true effect. The Type II error rate (typically $\beta = 0.2$) denotes the probability of failing to detect a true effect. Its complement, $1 - \beta$, is referred to as the statistical power of a test, typically set at 0.8 (80%). Power represents the probability of correctly detecting an effect of the specified size if it exists. The effect size quantifies the magnitude of the difference or relationship that the study is designed to detect and plays a central role in balancing sensitivity with efficiency in experimental design.

For approximately normal distributions, the required sample size to detect a specified effect size with given levels of significance and power can be estimated using the following standard formula:

$$n = \left(\frac{Z_{1-\alpha/2} + Z_{1-\beta}}{d}\right)^2 \qquad \text{(A I-1)}$$

In this expression, Z is the Z-score, and d is Cohen's d, which is the standardized effect size ($d = \delta/\sigma$). This test assumes equal variances across groups, an assumption that may not be held in practice. In cases of heteroscedasticity, researchers are advised to use variance-adaptive methods, such as Welch's t-test, which does not require the assumption of equal variances (section I.5.3).

Table-A I- interprets common effect sizes based on (Cohen, 2013). These are common interpretations in the behavioral sciences and are known to be conservative.

Table-A I-4 Common interpretations of the standardized effect size

d	Interpretation	$n_{\alpha=0.05,\beta=0.2}$
0.2	Small	197
0.5	Medium	32
0.8	Large	10

For right-skewed distributions, the formula below is used to determine the required sample size to detect an effect size (Cundill & Alexander, 2015):

$$\sqrt{N} = \frac{\left(Z_{1-\alpha/2} + Z_{1-\beta}\right)}{\log(\mu_1/\mu_2)} \cdot \sqrt{\frac{1}{Q_1\kappa_1} + \frac{1}{Q_2\kappa_2}} \qquad \text{(A I-2)}$$

In this expression, $Q_i = n_i/N$ represents the proportion of total samples allocated to group i, and κ_i denotes the shape parameter of the Gamma distribution in that group. Assuming equal group sizes (i.e., $n_1 = n_2 = n$ and $N = 2n$) yields $Q_1 = Q_2 = 1/2$, which simplifies the expression and often results in more stable and less biased variance estimates, particularly when the group variances are similar. In such cases, equal allocation also tends to maximize statistical power. However, under heteroscedasticity, this benefit may diminish, and alternative allocation strategies or variance-robust methods should be considered (section I.5.3).

Substituting these into the formula:

$$\sqrt{2n} = \frac{\left(Z_{1-\alpha/2} + Z_{1-\beta}\right)\sqrt{\frac{2}{\kappa_1} + \frac{2}{\kappa_2}}}{\log(\mu_1/\mu_2)} \qquad \text{(A I-3)}$$

Squaring both sides:

$$2n = \left(\frac{Z_{1-\alpha/2} + Z_{1-\beta}}{\log(\mu_1/\mu_2)}\right)^2 \left(\frac{2}{\kappa_1} + \frac{2}{\kappa_2}\right) \qquad \text{(A I-4)}$$

Finally, solving for n:

$$n = \left(\frac{Z_{1-\alpha/2} + Z_{1-\beta}}{\log(\mu_1/\mu_2)}\right)^2 \left(\frac{1}{\kappa_1} + \frac{1}{\kappa_2}\right) \quad \text{(A I-5)}$$

The standardized effect size on the log scale can be expressed as the log-ratio of group means. For the purposes of this analysis, a reasonable threshold for detection is defined as a 20% relative difference in means, corresponding $\mu_1/\mu_2 = 1.2$.

Also, the shape parameter of Gamma distributions can be reasonably expected to range from around $\kappa = 2$ to $\kappa = 5$ (Figure-A I-5), so the largest sample size is for $k_i = 2$.

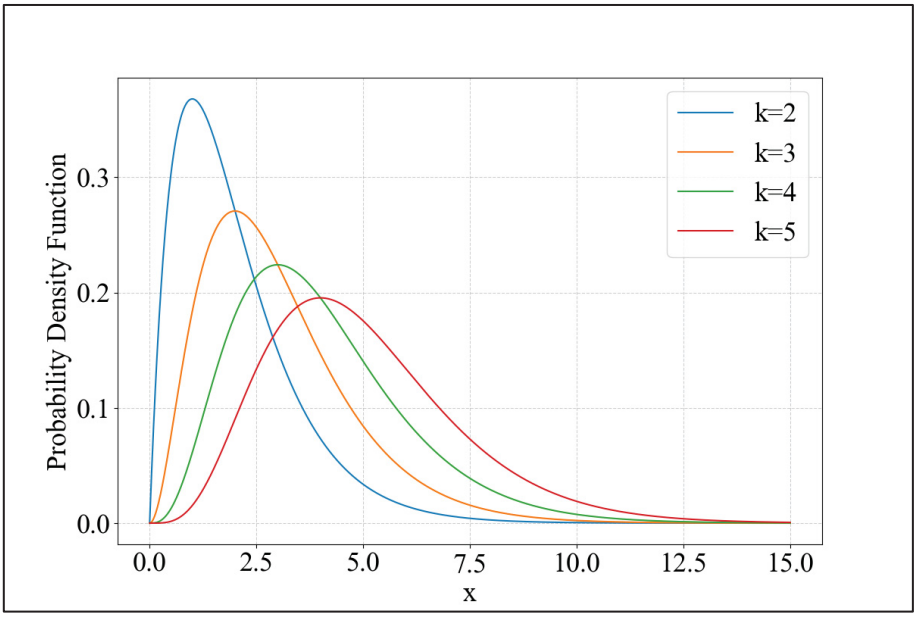

Figure-A I-5 Gamma distribution PDF for $\kappa \in \{2, 3, 4, 5\}$

Substituting these values into Equation-A I-5 with $\alpha = 0.05$ and $\beta = 0.2$ yields a sample size of approximately $n = 237$. This value was validated through simulation (Figure-A I-25).

For skewed distributions, the MDAF recommends using the nonparametric Mann-Whitney U test or the Kruskal-Wallis H test (see section I.5.3 for additional information) to test for differences in medians. These tests do not have a closed-form solution relating sample size to statistical power. Therefore, simulation methods were used to calculate a value that is approximately equal to that suggested by (Cundill & Alexander, 2015) as shown in Figure-

A I-25. In many cases, increasing sample size entails minimal cost, therefore the MDAF suggests using the largest calculated value rounded to $n = 250$. This value corresponds to $\beta \approx 0.84$ based on visual inspection of Figure-A I-25.

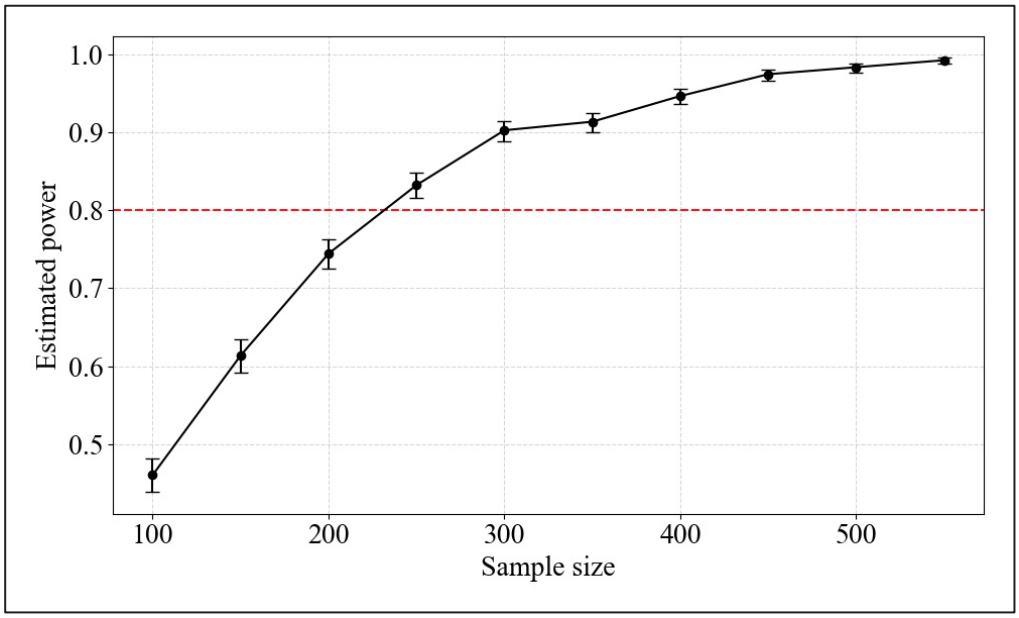

Figure-A I-25 Estimated power curve with 95% bootstrap CIs using the Kruskal-Wallis H test for a 20% difference in medians and equal shape parameters ($\kappa = 2$)

Whether comparing (a) approximately normal distributions or (b) right-skewed, positive-only distributions, the MDAF recommends a default sample size of $n = 250$ per group. This value typically achieves a statistical power of at least 0.8 in both scenarios. For case (a), it is sufficient to detect small, standardized effect sizes, as defined in Table-A I-4. For case (b), it allows detection of a mean ratio difference of approximately 20%, which may exceed the minimum effect of interest in some studies. In such cases, it is advisable to use Equation-A I-5 to calculate the required sample size more precisely. A sample size of $n = 250$ per group also offers several practical advantages. It is easy to remember, reasonably conservative, and remains within one order of magnitude of the sample sizes commonly observed in metaheuristics research. As noted by (Gagnon, Abran & April, 2025) 56% (95% CI [49.4, 62.6]) of reported sample sizes were less than or equal to $n = 30$ and that $n \in \{1, 5, 10\}$ are most common.

However, increasing sample size universally introduces the risk of detecting statistically significant but practically negligible effects. As statistical power increases, even trivial differences may reach significance, potentially diverting attention from more meaningful results. To mitigate this risk, the MDAF recommends systematically reporting both effect sizes and their associated confidence intervals (section I.5.4), to ensure that statistical significance is interpreted along with a measure indicative of practical relevance.

It is also important to acknowledge the limitations of applying a uniform sample size across studies. While using a standard value such as $n = 250$ simplifies planning and tends to raise statistical power, it does not account for study-specific factors. In some cases, it may even *reduce* power or lead to inefficient allocation of resources. For instance:

- When group variances differ substantially, larger sample sizes may be required in the group with higher variance to stabilize standard error estimates.
- When effect sizes are unequal across groups, fewer samples may be sufficient for the group with the larger effect, and more for the group with the smaller one.

To address these cases, researchers could consider asymmetric sample allocation, particularly when prior knowledge suggests unequal variability or effect magnitude between groups. Uniformly large samples can lead to unnecessary computational cost and effort. A more adaptive approach such as the iterative design method proposed by (Campelo & Takahashi, 2018) enables researchers to balance statistical rigor with efficiency by tailoring sample size to expected effect sizes, variance structure, and available resources.

In summary, given the statistical distributions typically observed in metaheuristics research, the MDAF suggests using a default sample size of $n = 250$ per group for practical reasons. Though lacking full analytical rigor, it is theoretically and empirically supported under common conditions. When these conditions are not satisfied, simulation-based approaches can be employed, such as the one used to derive Figure-A I-25. In rare cases, it may be necessary to refer to specialized literature.

I.3.3 Fair comparisons

A strong majority of research articles in metaheuristics aim to demonstrate that a proposed algorithm B outperforms a baseline algorithm A, or sometimes a variant of itself, such as B'. However, such comparisons can be misleading if A is poorly parameterized or if researchers invest significantly more effort in fine-tuning B than its comparators. Under these conditions, demonstrating superiority becomes a trivial exercise rather than a meaningful scientific contribution. To enhance the generalizability of conclusions, algorithms should be compared in configurations that are close to their respective performance optima.

In practice, identifying the globally optimal configuration for an algorithm may be computationally intractable, and aggressively pursuing it risks overfitting to the benchmark suite. For this reason, the MDAF does not expect exhaustive optimization of parameter settings. Instead, it recommends the following mitigating strategies:

- **Systematic parameter search**

  According to (Gagnon, Abran & April, 2025), approximately 64% (95% CI [56.5, 71.5]) of the articles that describe their parameter tuning methodology rely on values previously reported in the literature. This practice is problematic, as it typically lacks transparency regarding how those values were originally obtained. Without documentation of the underlying tuning procedures, it becomes difficult to assess whether the parameter settings are appropriate.

  Instead, the MDAF recommends the use of grid search to explore the parameter space and identify reasonable configurations. Grid search is a systematic tuning method in which a predefined set of candidate values is specified for each parameter. The algorithm is then evaluated across all possible combinations of these values. While exhaustive, this approach ensures transparency and repeatability and provides a structured means of identifying performant configurations without relying on undocumented or ad hoc defaults.

One effective way to determine a "reasonable" configuration is to use the rank-sum method over a suite of benchmark functions. This technique involves executing the algorithm with multiple parameter configurations P_i across a set of test functions f_j, and ranking the resulting performance or quality measure Q for each configuration-function pair. For each function, configurations are ranked from best to worst, and the ranks are then summed across all functions. The configuration with the lowest cumulative rank is selected as the most robust. This approach promotes generalizability by favoring configurations that perform consistently well across diverse problem instances, rather than excelling on a narrow subset.

For example, consider m parameter configurations P_1, P_2, …, P_m and n benchmark functions f_1, f_2, …, f_n. For each combination, the performance indicator $Q(P_i, f_j)$ is evaluated. For each function f_j (i.e., each row in Table-A I-5), the configurations are ranked based on their performance, with rank 1 assigned to the best-performing configuration. After computing ranks across all functions, the ranks are aggregated by configuration (i.e., column-wise). The parameter configuration corresponding to the lowest total rank in the bottom row is selected.

Table-A I-5 Rank-sum table for parameter configuration

	$\boldsymbol{P_1}$	$\boldsymbol{P_2}$	...	$\boldsymbol{P_n}$
$\boldsymbol{f_1}$	$\text{rank}(Q(P_1, f_1))$	$\text{rank}(Q(P_2, f_1))$	...	$\text{rank}(Q(P_n, f_1))$
$\boldsymbol{f_2}$	$\text{rank}(Q(P_1, f_2))$	$\text{rank}(Q(P_2, f_2))$	...	$\text{rank}(Q(P_n, f_2))$
⋮	⋮	⋮	...	⋮
$\boldsymbol{f_m}$	$\text{rank}(Q(P_1, f_m))$	$\text{rank}(Q(P_2, f_m))$	...	$\text{rank}(Q(P_n, f_m))$
$\boldsymbol{\Sigma}$	$\sum_{j=1}^{n} \text{rank}(Q(P_1, f_j))$	$\sum_{j=1}^{n} \text{rank}(Q(P_2, f_j))$	...	$\sum_{j=1}^{n} \text{rank}(Q(P_m, f_j))$

A step-by-step procedure to automate this analysis is given in Algorithm-A I-1 below.

Algorithm-A I-1 – Rank-sum optimal parameter set search procedure

Rank-sum optimal parameter set search procedure

Input: An algorithm $\mathcal{A}$, an array of parameter sets $\boldsymbol{P}_{\mathcal{A}} = [p_1, p_2, \dots, p_n]$, an array of test functions $\boldsymbol{F} = [f_1, f_2, \dots, f_m]$ and the number of evaluation replicates N.

Output: The parameter set p^* whose rank-sum is the highest

```
M_{m×n} ← zeros(m, n)                          ▶ Initialize zero arrays
R_{m×n} ← zeros(m, n)
S_{1×n} ← zeros(n)

for each f_i in F                              ▶ Evaluate (p_j, f_i) pairs
for each p_j in P_A
m_ij ← evaluate(A, p_j, f_i, N)
end for
end for

for each f_i in F                              ▶ Rank (p_j, f_i) pairs
R[i, :] ← rank(M[i, :])
end for

for each p_j in P                              ▶ Calculate rank sums
s_j ← sum(R[:, j])
end for

j* = index_of_min(S)                           ▶ Find best
p* = P_A[j*]
```

The evaluate function runs algorithm $\mathcal{A}$ on test function f_i a total of N times to generate an array of the form $\varphi = [\varphi_1, \varphi_2, \dots, \varphi_N]$ where each φ denotes a performance measure (e.g., best solution quality, first hitting time, etc.). A scalar summary statistic, such as the mean or median, is then computed from φ and returned. This value is stored in m_{ij} and the process is repeated $n \times m$ times in total.

This method is proposed as a practical compromise to the more rigorous methods presented in the next subsection, specifically those based on the design of experiments

(DOE). Its simplicity and intuitive nature increases the likelihood of adoption by researchers. Furthermore, by relying on rank-based comparisons rather than raw performance values, it is less susceptible to outliers and noise, which enhances robustness, particularly when the number of replicates is small. Finally, the method is generally less computationally demanding than DOE, albeit at the potential cost of reduced precision.

- **Design of experiments (DOE)**

The systematic parameter search method described above is a practical approach, but it results in the loss of information (i.e., the magnitude of the differences) by using ranks. To address these limitations, more rigorous methods are required, such as DOE. This section contains a general discussion of DOE. See (Montgomery, 2021) for a detailed treatment of this topic.

DOE is a systematic approach to determine the relationship between factors affecting a process and the output of that process. In the context of metaheuristic algorithms, DOE can be used to rigorously explore the parameter space and understand the effects of different parameter settings on an algorithm's performance.

The main steps involved in DOE for metaheuristic parameter tuning include:

- Identifying the key parameters (*factors*) that influence the algorithm's performance. These may include hyperparameters (e.g., learning rate, population size), structural components (e.g., crossover or mutation operators), or other experimental elements (e.g., initialization methods). For each factor, a set of values (levels) is selected to be tested. These factors and their respective levels define experimental space.

- Selecting an appropriate experimental design to systematically explore combinations of factors and levels. Common choices include factorial designs, fractional factorial designs, and Latin hypercube sampling. The choice depends

on the number of factors, available computational resources, and the desired resolution of interaction effects.

- Conducting the experiments by running the algorithm according to the configurations specified in the experimental design. For each parameter combination, the algorithm's performance is evaluated on a set of benchmarks as in Table-A I-5.

- Analyzing the results using statistical analysis. This typically involves analyzing the main effects and interactions of the parameters on the algorithm's performance. Techniques such as Analysis of Variance (ANOVA) can be used to identify significant factors and their contributions to performance variations. Finally, the best parameter settings can be determined.

One of the less explored areas in the current literature is the application of DOE in scenarios involving multiple test functions. A key challenge lies in addressing the fact that algorithm performance must be assessed across a diverse set of problem instances. The MDAF proposes a solution by modifying the representation in Table-A I-5: each cell contains a raw performance value, and each value within a row is normalized by dividing it by the maximum value in that row (as described in Algorithm-A I-2 below). This transformation preserves the relative magnitude of performance across problem instances while enabling the use of conventional DOE techniques. Additional refinements may include weighted aggregation schemes to account for the relative importance of individual test functions, as well as cross-validation procedures to enhance generalizability.

By using DOE, researchers may draw more reliable and generalizable conclusions about the performance of metaheuristic algorithms. Designs like full factorial and fractional factorial provide a thorough analysis of parameter effects, including main effects (i.e., the average effect on the response averaged over the levels of the other factors) and interactions.

These methods have a well-established theoretical foundation, but they also have steep learning curves. This is because some of the designs and analyses involved require a solid foundation in statistics. It is believed that this is the main reason for it not being used more often.

Although DOE may be more computationally intensive than simpler methods, like grid search or the proposed rank-sum method, the additional insights gained from it may justify the additional effort.

Algorithm-A I-2 Normalized performance aggregation procedure

Normalized performance aggregation procedure

Input: An algorithm $\mathcal{A}$, an array of parameter sets $\boldsymbol{P}_{\mathcal{A}} = [p_1, p_2, \dots, p_n]$, an array of test functions $\boldsymbol{F} = [f_1, f_2, \dots, f_m]$ and the number of evaluation replicates N.

Output: An array of aggregated performance measures $\boldsymbol{S} = [s_1, s_2, \dots, s_n]$.

```
M_{m×n} ← zeros(m, n)                    ▶ Initialize zero arrays
R_{m×n} ← zeros(m, n)
S_{1×n} ← zeros(n)

for each f_i in F                        ▶ Evaluate (p_j, f_i) pairs
for each p_j in P_A
m_ij ← evaluate(A, p_j, f_i, N)
end for
end for

for each i in {0, 1, ... m − 1}          ▶ Normalize rows and
m_max ← max (M[i, : ])                     aggregate
for each j in {0, 1, ..., n − 1}
s_i ← s_i + m_ij/m_max
end for
end for
```

- **Off-the-shelf automated parameter tuning**

The preceding sections discussed two relatively simple methods of finding good parameter settings. While the MDAF Python library offers open-source implementations of these methods, alternatives exist that may better suit the needs of some research projects (e.g., some may use the same programming language as the project like Java, C, C++, Python or R).

A recent survey (Huang, Li & Yao, 2020) identifies three categories of automated parameter tuning methods. The following section outlines the three categories, accompanied by a non-exhaustive list of representative open-source libraries and frameworks associated with each.

1. **Simple generate-evaluate methods**

   Simple generate-evaluate methods are straightforward approaches that consist of generating a set of candidate parameter configurations and evaluating each to find the best one. For example, in DOE, researchers can use a factorial design to systematically test all possible combinations of factors (parameters) at different levels. Similarly, in simple generate-evaluate methods, given a set of constraints, parameter configurations are generated and exhaustively evaluated to identify the best-performing one. The brute-force approach exemplifies this by exhaustively testing each configuration. This method rapidly becomes prohibitively inefficient. An improvement over brute-force is the F-Race algorithm introduced in (Birattari et al., 2002), which incrementally evaluates configurations on a stream of problem instances, discarding inferior candidates early using rank-based statistical testing to focus resources on the most promising configurations, thus enhancing efficiency.

2. **Iterative generate-evaluate methods**

   Iterative generate-evaluate methods improve efficiency by generating and evaluating configurations iteratively, using historical data to guide the search. This approach is

like an adaptive DOE method where experiments are conducted in stages and refined progressively.

Iterated F-Race (I/F-Race) is an example that extends the F-Race algorithm by iteratively updating a probabilistic model based on the performance of surviving configurations, allowing for a more focused search in promising regions (López-Ibáñez et al., 2016).

Another notable approach is ParamILS, which combines stochastic local search with specific mechanisms like adaptive capping to find optimal parameter settings efficiently (Hutter et al., 2009). These methods strike a balance between exploration and intensification, making them well-suited for large-scale parameter tuning problems.

3. **High-level generate-evaluate methods**

These methods aim to quickly generate a set of high-quality parameter configurations (exploration) followed by filtering (intensification), and evaluation of the remaining candidates (intensification) before final selection. The Post-Selection Mechanism is an example of this approach, dividing the tuning process into two phases: elite qualification and elite selection (Yuan et al., 2013).

Hyperband is another high-level technique that combines random search with early stopping to dynamically allocate resources across a broad range of configurations, progressively focusing on the most promising ones (Li et al., 2017). These methods, like the iterative generate-evaluate methods, try to balance exploration and intensification, making them suitable for large-scale tuning problems.

The previous libraries are generally more sophisticated than the methods proposed by the MDAF. For example, the F-Race and Iterated F-Race algorithms are natural extensions of Algorithm-A I-1 since they are based on ranks. F-Race focuses on the early elimination of "worse" solutions through statistical testing, and Iterated F-Race refines the candidate solution generation process iteratively. Both have clear advantages but also possess

disadvantages, such as the early elimination of "good" solutions due to high familywise error rates caused by repeated statistical tests. This issue can be mitigated using various correction procedures like Bonferroni's, Šidák's, or Tukey's, but these increase complexity and may be too conservative, potentially failing to eliminate "bad" solutions early enough to matter. Additionally, the underlying probabilistic model underpinning Iterated F-Race may converge prematurely or be unduly influenced by early results.

These methods generally receive less methodological and software support compared to the proposed DOE-based approaches, and their outcomes are often more challenging to interpret. This observation is not intended as a critique, but rather as a rationale for favoring more established and interpretable techniques. Additionally, a key advantage of traditional DOE-based methods is their ability to reveal interactions between parameters, an aspect that is typically overlooked or inadequately captured by alternative tuning strategies.

Despite differences among these approaches, it is reasonable to assert that any of the proposed methods represents an improvement over current common practice (Gagnon, Abran, & April, 2025). However, for this assertion to be held, it is essential that the same parameter tuning procedure be applied consistently across all algorithms under comparison. Ensuring uniformity in the tuning process helps mitigate parameterization bias, thereby increasing the likelihood that observed performance differences reflect genuine algorithmic capabilities. This principle lies at the core of what is meant by "fair" comparisons.

I.3.4 Other considerations

This section highlights more aspects of experimental design for which the MDAF offers explicit recommendations. By adhering to the preceding guidelines, researchers should be equipped with the following:

1. Well-formulated research questions
2. Clearly defined research variables
3. Complete and testable research hypotheses
4. Explicit research objectives

5. Adequate sample sizes
6. Fairly tuned algorithms

Achieving fair algorithm tuning (6) presupposes that the algorithms have been appropriately selected. As a starting point, the following set of algorithms is recommended to establish meaningful benchmarks:

1. A naïve baseline algorithm, such as Random Search
2. A tuned variant of the PSO algorithm
3. A tuned variant of the SA algorithm

Outperforming pure chance (e.g., Random Search) is a foundational principle of empirical science. In virtually all cases, a method that fails to yield results meaningfully better than random behavior should be rejected, as it provides no compelling evidence of systematic effectiveness. Beyond this minimal baseline, comparison with a tuned PSO variant offers a useful point of reference against a canonical multi-state or "swarm-based" algorithm. Similarly, comparison with a tuned SA variant serves as a representative baseline for single-state search strategies. These references help contextualize the performance of newly proposed algorithms within the broader landscape of metaheuristic design. In addition to this, context-specific comparators may be included to reflect the state of the art in the given problem domain.

These selections provide a balanced foundation for comparative analysis and help contextualize the performance of more complex or novel algorithms.

To compare algorithms, an appropriate suite of benchmark functions must also be selected. The choice of test functions is critical, as it directly influences the generalizability and interpretability of the results. The MDAF offers explicit recommendations to guide this selection process. An effective benchmark suite should satisfy the following criteria:

- **Diversity of problem characteristics**

The suite should include functions with varied features such as modality (unimodal vs. multimodal), separability (separable vs. non-separable), dimensionality, and landscape features (e.g., ruggedness, basins, valleys, etc.). This ensures that algorithms are evaluated across a broad spectrum of problem types.

It is important to recognize that while the breadth and diversity of a test suite are valuable, they should not be pursued at the expense of clarity or practicality. A comprehensive suite need not be excessively complex. For instance, the De Jong test suite, comprising only five benchmark functions, has been criticized for its simplicity and limited diversity. Nevertheless, it encompasses both unimodal and multimodal landscapes, is scalable to higher-dimensional spaces, exhibits diverse landscape features such as ridges, discontinuities, and noise, and includes functions with analytically known global optima. Its simplicity offers the advantages of analytical tractability and computational efficiency. When used in conjunction with more complex or domain-specific benchmarks, it can play a useful role within a broader and well-balanced experimental framework (see (Gagnon, April & Abran, 2020) for a worked example).

- **Relevance to the research question**

  Benchmark functions should reflect the types of optimization problems targeted by the research. For instance, if the goal is to optimize engineering design problems, the suite should include functions that emulate real-world applications.

- **Standardization**

  Whenever possible, use functions that are widely adopted in the metaheuristics literature such as those from the Black-Box Optimization Benchmarking (BBOB) or Congress on Evolutionary Computation (CEC) benchmark sets. This facilitates reproducibility and allows for more direct comparison with prior work.

The last element of experimental design covered by the MDAF is the selection of performance criteria. Performance is generally defined in terms of the best objective function

evaluation on a given problem instance. This leaves out critical information regarding the proposed algorithm because it ignores computational considerations. To remedy this, the MDAF suggests using the following performance criteria:

- **First hitting time**

  This measure captures the number of objective function evaluations required to reach a predefined target value (e.g., a known optimum). FHT is particularly relevant in time-sensitive applications or when early convergence is a desired property. Unlike CPU time, which can be influenced by hardware, software implementation, or computational platform, FHT is platform-independent. This makes it a more reliable and comparable indicator of algorithmic efficiency across studies and experimental setups.

- **Success rate**

  This measure captures the proportion of independent runs in which the algorithm successfully reaches a predefined target value (e.g., a known global optimum) within a fixed evaluation budget. Success rate is a useful complement to FHT, as it provides insight into the algorithm's reliability and consistency across stochastic runs. While FHT emphasizes how quickly a solution is found, success rate emphasizes whether a satisfactory solution is found at all. Like FHT, it is independent of hardware, software, and implementation details, making it a robust and comparable measure across platforms. High success rates are especially important in practical applications where repeatability and dependability are critical.

A hybrid performance criterion, referred to in this thesis as the Success-Normalized First Hitting Time (SN-FHT), was introduced in (Gagnon, April, & Abran, 2020) to reflect both efficiency and reliability. It is computed by dividing FHT by the success rate, thereby penalizing algorithms that converge quickly but do so inconsistently. This adjustment mitigates misleading optimism by estimating the expected number of function evaluations required per successful run, offering a more balanced and informative measure of algorithmic performance.

In summary, this section outlined several key experimental design considerations for conducting rigorous and reproducible research in metaheuristics, as recommended by the MDAF. By carefully selecting representative algorithms, diverse and relevant benchmark functions, and informative performance criteria, researchers can ensure that their evaluations are both fair and meaningful. These elements provide a strong foundation for comparative analysis and contribute to the development of more robust, interpretable, and generalizable optimization methods.

I.4 Implementation

This section outlines the MDAF's recommendations for implementing the experimental design developed in section I.3. It is organized as follows. Section I.4.1 describes the importance of version control for accessibility, transparency and reproducibility. Section I.4.2 addresses the principles and benefits of high-quality documentation followed by section I.4.3 which covers modular development. Finally, section I.4.4 covers the role of testing and automated validation for integrity, maintainability, and reproducibility.

I.4.1 Version control and accessibility

Version control systems (VCS) are software tools designed to track changes made to files over time, enabling the management of multiple project versions and facilitating collaboration within and between groups. In the context of metaheuristics research, VCS help ensure the transparency, traceability, and reproducibility of experimental results.

Transparency is facilitated by making a version-controlled repository (i.e., the codebase) publicly accessible. This level of visibility may encourage greater diligence, as studies have shown that authors who decline to share their data are more likely to have committed errors in their analyses and tend to employ weaker statistical methodology (Wicherts et al., 2011).

Traceability is inherently supported by VCS, as each modification is recorded with a unique hash identifier and accompanied by metadata including the author, a timestamp, and a descriptive commit message. Reproducibility is further ensured by referencing the specific commit hash, or its human-readable alias known as a Git tag (e.g., v1.0), within scholarly

publications. In addition, a specific commit hash can be linked to a Digital Object Identifier (DOI) through platforms such as Zenodo, which serve as archival repositories and assign a permanent, citable web address to the corresponding version of the codebase. Zenodo is maintained by the European Organization for Nuclear Research (CERN), known in French as the "Conseil Européen pour la Recherche Nucléaire", which developed and maintained it in the context of the Open Science initiative to promote transparency, accessibility, and reusability of research outputs across disciplines. To maximize these benefits, all research artifacts, including source code, scripts, configuration files, documentation, and manuscripts, should be maintained under version control.

VCS use local and remote (e.g., hosted on GitHub's, GitLab's, or BitBucket's servers) repositories to store files and maintain a synchronized record of changes across different development environments. Local repositories allow researchers to track and manage changes on their own machines, while remote repositories serve as centralized platforms for backup, collaboration, and integration. Distinct features or issues can be developed concurrently by leveraging branches which allow researchers to isolate specific lines of development, such as new algorithmic components, experimental configurations, or bug fixes, without affecting the main codebase. This practice preserves a complete and auditable history of changes, allowing researchers to trace the origin of specific outcomes and to reproduce results.

In addition to these benefits, VCS also integrates with other important practices such as documentation (section I.4.2), modular development (section I.4.3), testing and automated validation (section I.4.4).

For these reasons, the MDAF strongly recommends adopting a version control system from the outset of a research project and incorporating as many relevant files as possible.

I.4.2 Documentation

Recent estimates suggest that software maintenance accounts for 60% or more of the total cost of a software project (Singh et al., 2019), indicating that most of the project effort is

dedicated to ensuring the system remains functional over time. This underscores the critical role of comprehensive, high-quality documentation in supporting maintenance activities. As noted in the Software Engineering Body of Knowledge (SWEBOK) v4.0, “Meaningful and comprehensive documentation is crucial at all stages of software lifecycle,” particularly because it ensures that knowledge about the system’s architecture, behavior, and rationale is retained and accessible over time (SWEBOK v4.0, Chapter 16, Computing Foundations).

At a minimum, scripts, functions, and modules should include inline comments and docstrings that describe their functionality, expected inputs and outputs, and any key assumptions or side effects. Descriptions should prioritize clarity and avoid jargon when possible.

In addition to internal documentation, each repository should include a top-level README file that provides a high-level overview of the project. This document typically outlines the project’s goals, the structure of the repository, installation instructions, software dependencies, and step-by-step guidance for getting started or reproducing key results. Supplementary files, such as CONTRIBUTING.md, LICENSE, and CITATION.cff, can further clarify usage rights, contribution guidelines, and citation formats.

For more complex projects, particularly those intended for reuse, it is highly recommended to provide extended documentation. Platforms such as GitHub Wikis, Read the Docs, or static HTML documents generated with Doxygen or Sphinx, usually stored in a docs/ folder at the root of the project, can be used to create detailed guides containing usage examples, and design rationales. This type of documentation improves the project’s accessibility, reduces onboarding time for new contributors, and serves as a long-term knowledge base that can outlast the original development “team”, which can consist of a single researcher. By clearly explaining the architecture, intended use cases, and integration points, in-depth documentation significantly enhances the sustainability and impact of research software.

To promote reproducibility, experimental protocols and configuration files should be documented in a manner that allows others to replicate the procedures precisely. For instance, parameter settings used for algorithm evaluation should be clearly reported

alongside the conditions under which experiments were conducted. The inclusion of structured metadata, such as the problem instances used, the number of repetitions, random seed values, and computational environment, further reinforce replicability.

Literate programming tools such as Jupyter Notebooks, RMarkdown, Sweave, or Org Mode can be effective for combining code, narrative explanation, and results into a single, executable document. These tools enable researchers to construct computational narratives that seamlessly integrate descriptive text with the scripts that generate figures, tables, or performance measures, thereby minimizing divergence between reported results and the procedures that produced them (Sandve et al., 2013). This addresses a common failure point where written documented methods diverge from what was done. If the document is the code, then they cannot diverge. Some examples are provided in the referenced GitHub repositories of this thesis.

Finally, documentation should be viewed as a living component of the research process. As the project evolves, the associated documentation must be updated accordingly to remain accurate and useful. Projects that treat documentation as an afterthought risk accumulating technical debt and undermining the reproducibility and longevity of their results.

For these reasons, the MDAF advocates for including disciplined documentation practices into all phases of the research lifecycle. When combined with version control (section I.4.1) and other software engineering practices, documentation becomes a foundational element of high-quality, reproducible computational research.

I.4.3 Modular development

“Modularity measures the degree to which a system or software is composed of components that are independent, such that a change to one component has minimal impact on other components” (SWEBOK v4.0, Chapter 7, Software Maintenance).

A recent survey (Gagnon, Abran & April, 2025) estimates that 8%, (95% CI [7.1, 8.9]) of research articles provide access to the source code used to generate the reported results. None of the surveyed codebases employed modular design. Instead, they relied on monolithic

structures, typically implemented as scripts with tightly coupled components. This is common for small projects with a single release, but it negatively impacts development time and quality of subsequent releases and compromises reusability (SWEBOK v4.0, Chapter 2, Software Architecture).

The MDAF advocates human-centered coding standards to enhance software quality and promote code reuse. This involves structuring code into logically organized, clearly named functions and modules to reduce the cognitive load on readers. Each function, class, or method should serve a single, well-defined purpose to avoid the emergence of so-called "spaghetti code," a disorganized and difficult-to-maintain structure that is often comprehensible only to its original author(s). A guiding principle is DRY, or "Don't Repeat Yourself," which discourages duplicating code across different parts of a program. Copying and pasting code can lead to inconsistencies, as any modification in one instance must be manually replicated in all others to maintain correctness. In practice, this can be achieved by writing a library of common routines or classes for metaheuristic algorithms (e.g., a function for solution initialization, a method for evaluating the fitness of a solution, etc.). Another way is to leverage existing libraries whenever possible.

Modularity also facilitates testing and automated validation (section I.4.4), as smaller, self-contained functions are easier to isolate. This improves the reliability of the software and accelerates debugging, especially in experimental contexts where rapid iterations are common. Moreover, modular code is inherently more collaborative: contributors can work on distinct components with minimal risk of conflicts, and the interface between modules becomes a natural boundary for responsibility and review. From a maintainability perspective, modular systems are better equipped to evolve over time, as updates to one module can be made with minimal risk of unintended side effects elsewhere in the codebase. For research software intended to support reproducibility and extension, these qualities are particularly desirable.

In summary, modular development is a foundational practice that improves the quality, readability, maintainability, and testability of research software. For these reasons, the

MDAF advocates for adopting modular development practices from the early stages of a research project. When combined with version control (section I.4.1), disciplined documentation (section I.4.2), and other software engineering principles, modular design becomes a basis of high-quality, maintainable, and reproducible computational research.

I.4.4 Testing and automated validation

“Software testing consists of the dynamic validation that a system under test (SUT) provides expected behaviors on a finite set of test cases suitably selected from the usually infinite execution domain” (SWEBOK v4.0, Chapter 5, Software Testing).

Correctness and robustness are essential components of quality in scientific research. Testing and validation practices help to detect defects in the implementation but also to build confidence in the reliability of the results.

Unit tests and assertions play a fundamental role in verifying that individual components behave as intended. In computational research, however, traditional software testing must often be complemented with domain-specific validation strategies. For instance, when developing a new metaheuristic algorithm, one should first evaluate its performance on simplified instances where the optimal solution is analytically known or computationally tractable. This approach allows researchers to confirm that the algorithm can safely be used for further benchmarking.

Additionally, special or degenerate cases (e.g., toggling a feature via a parameter) should be explicitly tested to ensure correct behavior across the full operational range of the software. It is also advisable to incorporate invariance checks. For example, one may want to ensure that distances between agents or particles are strictly non-negative. These practices can reveal both logical errors and subtle numerical instabilities.

Beyond correctness, performance profiling may be employed to identify computational bottlenecks. However, optimization efforts should only be undertaken after functional correctness is established.

Automating test execution through structured test suites and integrating them into continuous integration (CI) pipelines significantly improves development efficiency and reliability. Platforms such as GitHub Actions allow developers to automatically run tests on every commit or pull request, helping ensure that modifications do not inadvertently break existing functionality and that results remain consistent over time. This is particularly critical in collaborative or long-term research projects where code evolves across multiple contributors and iterations.

Together, these practices support not only the immediate integrity of the code but also its long-term quality, maintainability, and reproducibility. For these reasons, the MDAF advocates for incorporating systematic testing and automated validation from the early stages of a research project. When combined with version control (section I.4.1), disciplined documentation (section I.4.2), and modular development practices (section I.4.3), testing becomes a basis of high-quality, maintainable, and reproducible computational research.

I.5 Analysis and reporting

This section presents the MDAF's recommendations for analyzing the results produced by the experimental design and implementation practices outlined in sections I.3 and I.4.

The goal of this section is to promote the use of rigorous, transparent, and interpretable statistical methods for validating experimental results and drawing meaningful insights. Emphasis is placed on reproducibility, appropriate use of statistical tests, effect size reporting, and the avoidance of common analytical pitfalls. The practices outlined here are designed to complement the MDAF's broader objectives of increasing the scientific credibility and practical utility of metaheuristics research.

The remainder of this section is organized as follows. Section I.5.1 describes the use of descriptive statistics to summarize algorithm performance across runs and problem instances. Section I.5.2 discusses recommended visualization techniques for exploring and communicating performance patterns. Sections I.5.3 through I.5.5 then address the statistical

interpretation of results, covering null hypothesis significance testing, the use of confidence intervals, and the reporting of effect sizes, respectively.

I.5.1 Descriptive statistics

Descriptive statistics offer a compact representation of experimental data by quantifying properties such as central tendency, dispersion or variability, shape, and other attributes of interest. Providing these enables readers to efficiently evaluate intergroup differences (e.g., between algorithm performance distributions) and to assess the magnitude of observed effects (Utochkin, 2015). The APA's Task Force on Statistical Inference (TFSI) gives the following recommendations (Leland, 1999):

1. Always report sample sizes.
2. In cases where the distribution is normal or approximately normal, it is recommended to report at minimum the mean and the standard deviation (SD).
3. In the case of non-normal distributions, it is recommended to report at minimum the median and the IQR.

The SAMPL Guidelines (Statistical Analyses and Methods in the Published Literature) also adds the following (Lang & Altman, 2015):

4. Report data with a level of precision suitable to the field, ensuring that rounding enhances readability without undermining the analysis.
5. Always report percentages alongside their raw counts: "3 out of 10 (30%)" rather than stating the percentage alone.

An example of (4) specific to metaheuristics research is to report a runtime of 10.5 seconds instead of 10.45125 or 100 mean function evaluations instead of 99.625.

As noted by (Gagnon, Abran & April, 2025), 79% of recent metaheuristics studies include measures of central tendency (95% CI [69.8, 88.2]), while only 50% report measures of variability (95% CI [44.1, 55.9]). This constitutes a readily addressable gap, as the calculation of these statistics is computationally trivial. Likewise, sample sizes are omitted

in 7% (95% CI [6.2, 7.8]) of the articles surveyed, an issue that could be rectified with minimal effort.

In the case of non-normal distributions, it is also recommended to provide five-point summaries consisting of the following elements:

1. **Minimum:** The smallest observed value in the dataset.
2. **First quartile (Q1):** The median of the lower half (25^{th} percentile) of the data.
3. **Median (Q2):** The middle value (50^{th} percentile) of the data.
4. **Third quartile (Q3):** The median of the upper half (75^{th} percentile) of the data.
5. **Maximum:** The largest observed value in the dataset.

This summary is particularly useful for understanding the shape of the data's distribution, especially when supported by high-quality visual representations such as histograms or boxplots (section I.5.2). Due to its multiple components, presenting the information in a table (e.g., Table-A I-6) is advised for clarity.

Table-A I-6 Example of a five-point summary with mean and standard deviation

Algorithm	$\boldsymbol{n}$	**Min**	$\boldsymbol{Q1}$	$\boldsymbol{Mdn}$	$\boldsymbol{Q3}$	**Max**	$\boldsymbol{IQR}$	$\boldsymbol{M \pm SD}$
A	250	305	390	420	445	490	55	412.3 ± 28.7
B	250	360	470	510	560	620	90	508.1 ± 42.6

Although not explicitly mandated by the TFSI or the APA's Publication Manual, both implicitly encourage clarity and transparency. Consequently, in cases where normality is required or assumed, it may be useful to report common measures of distributional shape such as skewness and kurtosis. Histograms and boxplots are frequently employed as graphical alternatives for assessing distributional shape (section I.5.2). None of the articles surveyed by (Gagnon, Abran & April, 2025) included such numerical measures, and only 14% (95% CI [12.4, 15.6]) provided graphical alternatives.

For bivariate or multivariate analyses, descriptive statistics can be extended to capture relationships between variables through measures such as cross-tabulations, contingency

tables, and measures of association for groups of variables. In addition, graphical methods such as scatterplots can complement these by revealing patterns not easily detected through numerical measures alone. Descriptions of conditional distributions may also be useful when investigating how one variable behaves given the value of another, especially in the context of algorithm performance across varying problem features or benchmark functions.

The preceding recommendations are summarized in Figure-A I-3 and Figure-A I-4 below as flowcharts. Their purpose is to guide researchers in selecting appropriate descriptive measures and visual aids for systematic informative reporting. Further details regarding the recommended graphical representations can be found in section I.5.2.

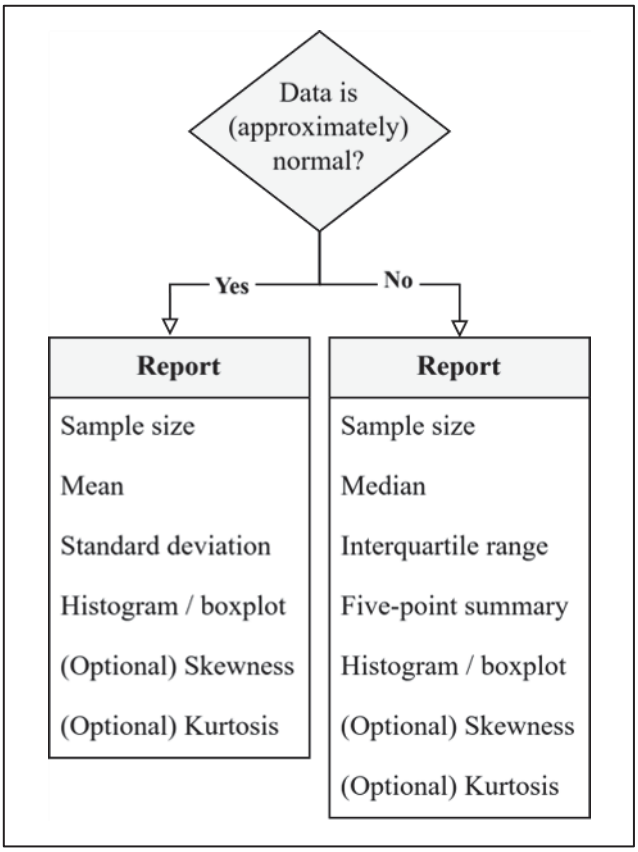

Figure-A I-3 Reporting guidelines for univariate statistics based on type and relationship

The above-mentioned values should be reported using common abbreviations for clarity (Table-A I-7) when referred to in the narrative form (Field & Hole, 2011).

Table-A I-7 Standard abbreviations for statistical values

$\boldsymbol{M}$	Mean
$\boldsymbol{Mdn}$	Median
$\boldsymbol{SD}$	Standard deviation
$\boldsymbol{Sk}$	Skewness
$\boldsymbol{K}$	Kurtosis
$\boldsymbol{SE}$	Standard error
IQR	Interquartile range

The following are illustrative examples:

- Particle Swarm Optimization ($M = 745.50$, $SD = 135.45$, $n = 250$) achieved lower first hitting times on the Rastrigin function than Simulated Annealing ($M = 1225.75$, $SD = 335.10$, $n = 250$), suggesting superior convergence performance under the evaluated settings.

- Algorithm A required an average of 10,500 function evaluations ($SE = 420$), while Algorithm B required 12,100 ($SE = 380$), suggesting greater efficiency and consistency in Algorithm A's performance.

- Particle Swarm Optimization converged more rapidly than Algorithm B, as evidenced by a lower median number of function evaluations ($Mdn = 420$, $IQR = 390 - 445$, $n = 250$) compared to Simulated Annealing ($Mdn = 510$, $IQR = 470 - 560$, $n = 250$), suggesting both faster and more consistent performance.

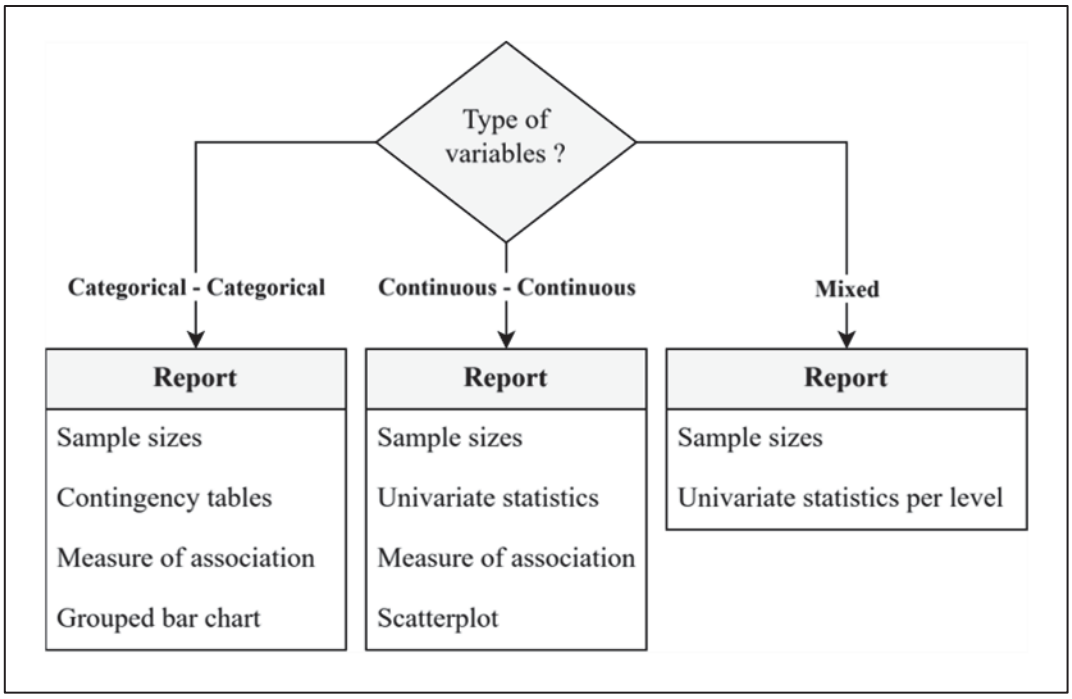

Figure-A I-4 Reporting guidelines for bivariate / multivariate descriptive statistics based on type and relationship

Contingency tables (Table-A I-8) describe the relationship between two categorical variables, such as when comparing success/failure outcomes across algorithms, or other categorical groupings (e.g., acceptable/not acceptable, converged/not converged, low/medium/high, etc.).

Table-A I-8 Illustrative example of a contingency table

Algorithm	**Success**	**Failure**	**Total**	**Success rate [%]**
A	24	6	30	80.0
B	17	13	30	56.7
Total	41	19	60	-

Note: A chi-squared test of independence indicated a statistically significant association between algorithm and success rate, $\chi^2(1, n = 60) = 4.57, p = .033$.

See section I.5.3 for more information regarding the reference to the chi-squared test below Table-A I-9. Common measures of association for categorical data include the phi coefficient (φ) for 2×2 tables and Cramér's V for larger tables. Here is a contextualized example in narrative form:

> As shown in Table-A I-8, Algorithm A achieved a higher success rate (80.0%) than Algorithm B (56.7%) across 30 independent runs each. A chi-squared test of independence confirmed that the difference in success rates was statistically significant, $\chi^2(1,\ n = 60) = 4.57,\ p = .033$. To quantify the strength of this association, the phi coefficient was computed, yielding $\varphi = 0.276$. This indicates a small-to-moderate association between algorithm choice and success outcome according to conventional thresholds (Cohen, 2013). In other words, algorithm selection appears to have a nontrivial influence on the likelihood of success under the tested conditions.

The referred thresholds for the phi coefficient and Cramér's V are given in (Cohen, 2013) and are reproduced in Table-A I-9.

Table-A I-9 Interpretation of common measures of association

Effect size	**Value**
Small	.10
Medium	.30
Large	.50

Common measures of association for continuous data include Pearson's correlation coefficient (r), Spearman's rank correlation coefficient (ρ), and Kendall's tau (τ), each suited to different assumptions about linearity, normality, and monotonicity.

Here is a contextualized example in narrative form:

> To investigate the relationship between algorithm convergence time and final solution quality, Pearson's correlation coefficient was computed across 50 independent runs per algorithm. For Algorithm A, convergence time and solution quality were moderately negatively correlated, $r = -0.42$, $p = .003$, indicating that shorter convergence times tended to yield higher-quality solutions. In contrast, the relationship was weaker for Algorithm B, with $r = -0.19$, $p = .18$, suggesting a non-significant association. These results suggest that Algorithm A's convergence behavior is more systematically related to its optimization performance.

When assumptions of linearity or normality are not met, nonparametric alternatives such as Spearman's ρ and Kendall's τ are used (Sheshkin, 2024). These measures assess the monotonic relationship (i.e., always either increasing or decreasing) between variables and are more robust to outliers and skewed distributions. The conventional thresholds for interpretation are the same as the ones given in Table-A I-9.

Continuous measures of association provide insight into the strength and direction of relationships between numeric variables. The MDAF recommends reporting both the correlation coefficient and its statistical significance while also specifying the method used (e.g., Pearson vs. Spearman) and verifying assumptions when applicable. See section I.5.3 for additional information.

In summary, descriptive statistics are essential for summarizing experimental results and communicating them in a manner that is both clear and informative. Despite their simplicity and computational efficiency, their consistent and complete reporting remains uneven across the metaheuristics literature. As noted by (Gagnon, Abran & April, 2025), about 80% of surveyed studies report measures of central tendency, but only about half include measures of variability, and fewer than 15% supplement their analyses with graphical summaries. These omissions hinder interpretation and reproducibility. The MDAF recommends systematic inclusion of sample sizes, measures of central tendency and variability, and five-point summaries when appropriate. These should be reported per algorithm and per problem

instance and ideally supported by graphical representations to convey distributional characteristics. While this section offers concrete examples and guidelines, the number of possible descriptive scenarios is too large to address exhaustively. Researchers are therefore encouraged to apply the overarching principles of transparency, clarity, and reproducibility to guide their reporting decisions. These recommendations are summarized in checklist format in APPENDIX V, and additional details are available in (Field & Hole, 2011) and (Hatcher, 2013).

I.5.2 Visualization

Graphical representations serve as essential complements to numerical summaries by conveying patterns, anomalies, and distributional features that may be difficult to detect through descriptive statistics alone. Effective visualizations can highlight intergroup differences, detect outliers, reveal multimodality, and communicate uncertainty. These features contribute to a more comprehensive interpretation of experimental results.

In the context of metaheuristics research, the most common and informative visualizations include histograms, boxplots, and bar charts. Histograms (Figure-A I-26) provide insight into the shape, skewness, and modality of continuous distributions, helping researchers assess whether assumptions of normality are reasonable. Beyond this, histograms can illustrate the presence or absence of extreme values by revealing whether the distribution exhibits heavy tails. Boxplots (Figure-A I-27), in turn, offer a compact view of the five-point summary (i.e., minimum, first quartile, median, third quartile, and maximum), while also highlighting potential outliers. This format also facilitates group comparisons by enabling side-by-side placement of boxplots. Grouped bar charts (Figure-A I-28) are particularly useful for visualizing comparisons across categorical groupings, such as algorithm performance across multiple problem instances or parameter settings.

The dotted lines of Figure-A I-26 represent kernel density estimates, which provide a smooth approximation of the underlying distribution and help reveal structure such as skewness, modality, or the presence of heavy tails that may not be immediately visible from the

histogram bars alone. This is in line with the guiding principle of clarity espoused by the APA, the SAMPL guidelines, and therefore the MDAF as well.

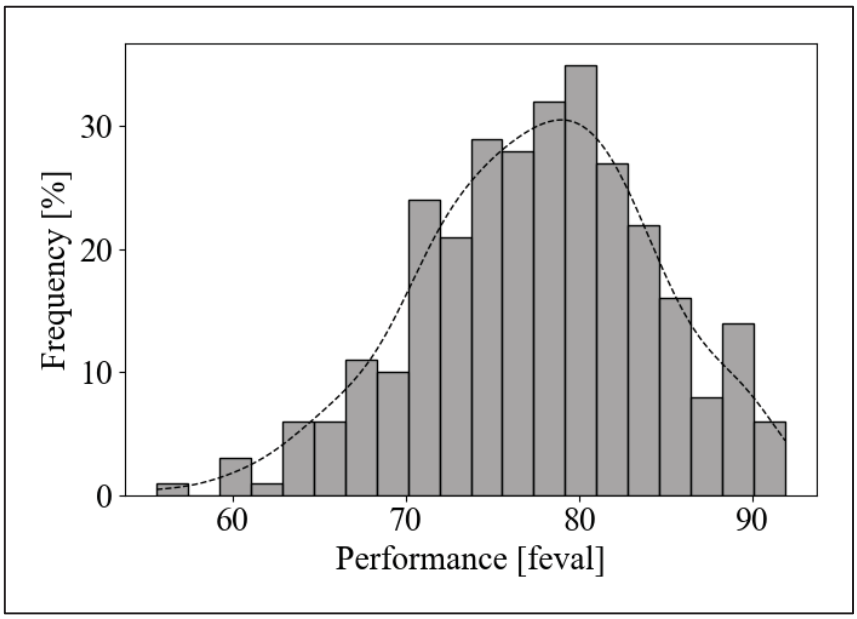

Figure-A I-26 Illustrative example of a histogram with a smooth density approximation

In Figure-A I-27, the data points identified by round markers with black edges are meant to be interpreted as outliers. It may be useful to explicitly mention how these are calculated. For example, in the standard Tukey boxplot, an outlier is any point that falls outside of $[Q_1 - 1.5 \times IQR, Q_3 + 1.5 \times IQR]$, where Q_1 and Q_3 denote the first and third quartile respectively, and $IQR = Q_3 - Q_1$. This rule is implemented by default in the `Seaborn` and `Matplotlib` libraries in the Python programming language as well as the boxplot() function in the R programming language. It is also important to clearly describe the outlier detection method if an alternative is used such as a variant of the Tukey method, an approach based on the Z-score, or any other technique. This becomes particularly critical if outliers are removed from the analysis, as this may impact results and interpretations. If outliers are retained, their influence on the results should be explicitly addressed and discussed.

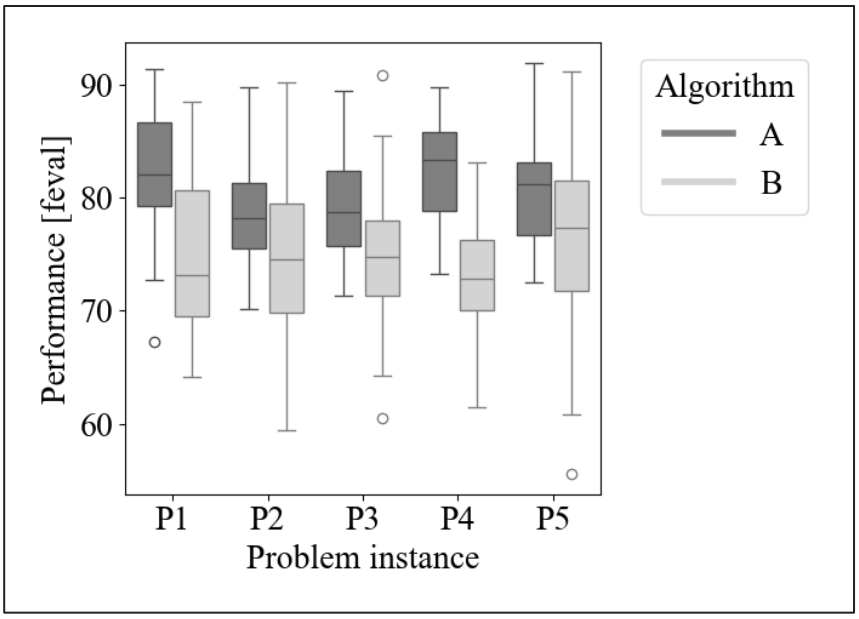

Figure-A I-27 Illustrative example of grouped boxplots

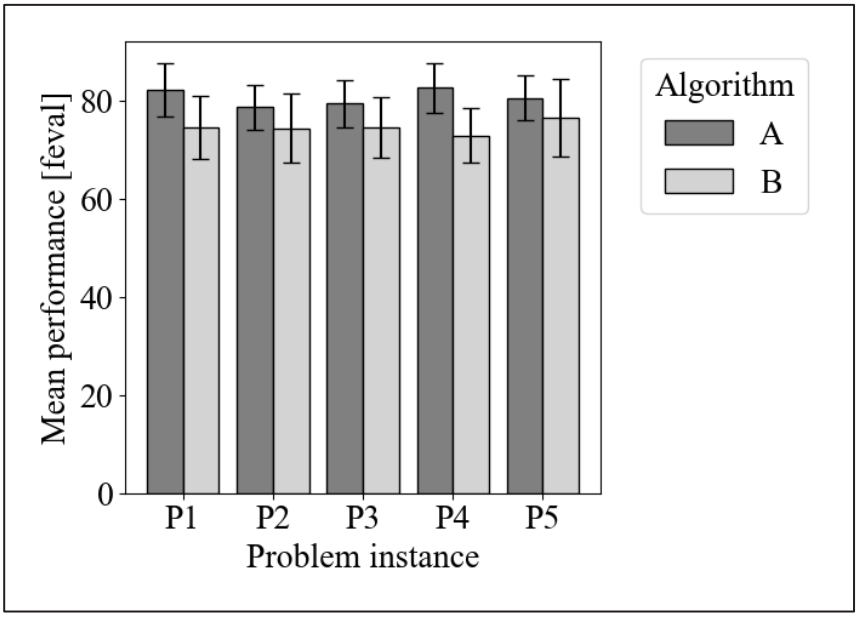

Figure-A I-28 Illustrative example of a grouped bar chart

The MDAF recommends including error bars in figures to convey the precision of an estimate, such as the variability around a mean as in Figure-A I-28. However, it strongly advises against using error bars to visually assess statistical significance between groups,

echoing concerns raised by (Lanzante, 2005). Even in cases where the risk of a false negative is not severely impacted, such as when group variances differ markedly, drawing inferences from overlapping or non-overlapping error bars can be misleading. This is partly because the meaning of the error bars is often left unspecified: they may represent standard deviations, standard errors, or confidence intervals, or may have been derived from bootstrap resampling. Without clear annotation, visual interpretation is imprecise and may lead to erroneous conclusions. For this reason, the MDAF emphasizes that error bars should serve primarily as a descriptive aid, not as a substitute for formal hypothesis testing (section I.5.3).

Despite their utility, graphical summaries remain underutilized. As reported by (Gagnon, Abran & April, 2025), 14% (95% CI [12.4, 15.6]) of surveyed studies include such visualizations. This represents a significant missed opportunity, given the minimal effort required to generate these plots with modern tools. Their inclusion not only improves interpretability but also facilitates peer review and replication by making distributional properties more readily apparent.

For bivariate and multivariate data, visualizations play a similarly important role. Scatterplots, while not descriptive statistics in the strictest sense, serve as exploratory tools that can reveal correlations, clusters, or trends between variables, such as the relationship between algorithm performance and the value of a hyperparameter (Figure-A I-29). When paired with descriptive statistics (e.g., means, standard deviations), these plots can support more nuanced analyses. Similarly, pairwise comparison grids, sometimes called pairplots, can provide a compact overview of multivariate interactions.

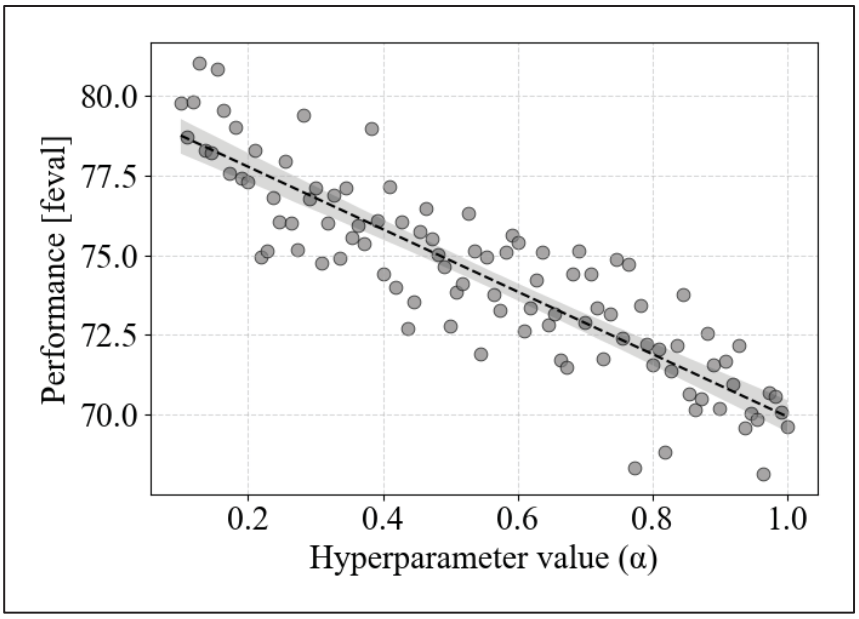

Figure-A I-29 Illustrative scatter plot showing a linear trend with 95% confidence interval

To promote reproducibility and clarity, the MDAF recommends that visualizations be used systematically and not merely as decoration. Each figure should be accompanied by a descriptive caption and, where applicable, annotated with relevant statistics or confidence intervals. Axes must be clearly labelled, and consistent colour schemes should be used to avoid misinterpretation. For comparative studies, plots should include all relevant baselines to ensure fairness and transparency.

The MDAF cautions against the following common pitfalls:

- **Overburdened figures**: For example, where too many variables, annotations, or styling elements are combined into a single plot. Complex relationships should be broken into multiple, well-scoped figures to maintain clarity. Additionally, figures can be supplemented with accompanying tables or narrative explanations to distribute cognitive load.
- **Inconsistent labeling**: Omitting or inconsistently applying axis titles, units of measurement, or legends, which can confuse interpretation and hinder reproducibility.

- **Inconsistent axis scaling**: Using different axis ranges or intervals for similar figures without justification, which may mislead comparisons or exaggerate differences.
- **Truncated axes**: These can exaggerate or obscure differences between groups. Always start the y-axis at zero unless a justified rationale is clearly indicated.
- **Overuse of color or 3D effects**: This may distract from the data or reduce accessibility for colorblind readers. Grayscale or colorblind-friendly palettes should be preferred.
- **Low-resolution images**: This may degrade readability. Use vector images (e.g., the SVG format) whenever possible.

In summary, graphical representations enhance the communication of experimental results by complementing numerical summaries with intuitive visual insights. When designed carefully and interpreted alongside descriptive statistics, they provide a powerful means to assess algorithm behaviour, support informed conclusions, and uphold the principles of rigorous empirical research.

I.5.3 Statistical significance testing

Statistical hypothesis testing serves as a critical complement to descriptive analysis (section I.5.3) by enabling researchers to assess whether observed differences (e.g., a difference in mean convergence time) or relationships (e.g., Pearson's r) in experimental data are statistically distinguishable from chance. In metaheuristics research, this often involves comparing solution quality, convergence time, or success rates across multiple algorithms or problem instances. Despite its potential to strengthen empirical claims, hypothesis testing remains inconsistently applied and frequently misused in the literature (Gagnon, Abran & April, 2025). For example, 43% (95% CI [38.0, 48.0]) use statistical testing, and only 17% (95% CI [15.0, 19.0]) discuss or validate underlying assumptions. The MDAF addresses this gap by proposing a simple and structured approach to hypothesis testing that emphasizes practical validity.

The goal of hypothesis testing is not to prove one algorithm categorically "better" than another, but to evaluate whether observed performance differences are statistically

distinguishable from chance. The MDAF's position is that statistical significance alone is insufficient without confidence intervals (section I.5.4) and effect sizes (section I.5.5). Together, they provide a decision-making framework that, when used carefully, reinforces the reliability of experimental results. The MDAF aligns with the position of (Cohen, 1993), who argued against dichotomous accept/reject decisions and emphasized reporting exact p-values, confidence intervals, and effect sizes. This stance is also endorsed by the TFSI (Leland, 1999), the APA's Publication Manual, and others.

As mentioned in section I.3, performance-related data typically follow one of two distributional patterns: (1) approximately normal distributions, characterized by symmetry or a bell-shaped curve, and (2) right-skewed, strictly positive distributions. The MDAF separates these two categories based on skewness (Sk) in Equation A I-6.

$$\begin{aligned} &\textbf{Category 1: } |Sk| < 0.5 \\ &\textbf{Category 2: } |Sk| \geq 0.5 \end{aligned} \tag{A I-6}$$

Empirical support for this rule is provided by analyzing how Shapiro–Wilk p-values vary with skewness. The test evaluates whether the data is likely to follow a normal distribution (i.e., symmetrical). Under the null hypothesis H_0, normality is rejected when $p < \alpha$ (e.g., 0.05 or 0.01). Figure-A I-30 shows that near-zero p-values emerge in appreciable numbers only when skewness reaches or exceeds 0.5. Note that the points represent the results of tests run on samples of size $n = 30$. To ensure accurate categorization, this should be considered alongside histogram evidence, which is recommended in section I.5.2.

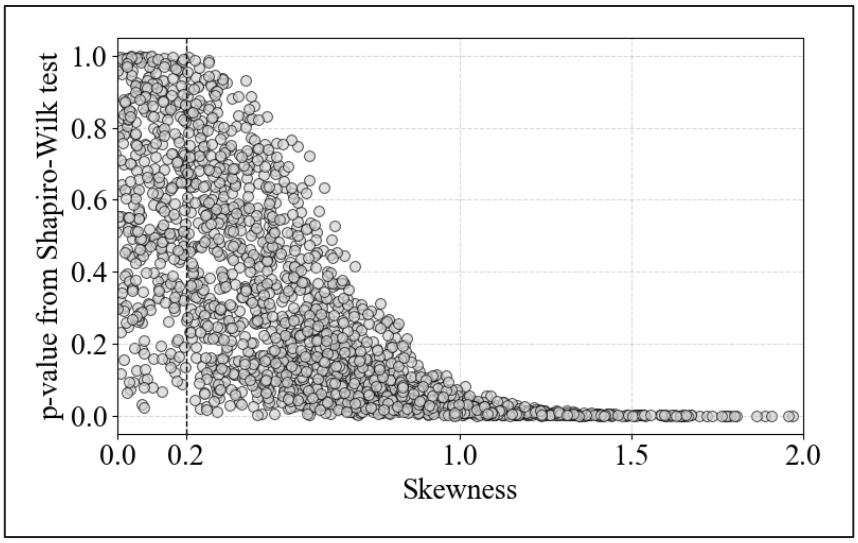

Figure-A I-30 Impact of skewness on normality via the Shapiro-Wilk test ($n = 30$)

For symmetrical data (i.e., Category), the MDAF recommends using Welch's t-test to compare two groups or Welch's ANOVA for more than two groups. The simulation studies reported in (Wilcox, 2021) show that they are robust generalizations of the classical tests that do not require the assumption of equal variance (i.e., homoscedasticity) and are less affected by unequal sample sizes. Other examples that support this recommendation are found in the field of psychology with (Delacre et al., 2017) and (Ruxton, 2006) which recommend using Welch's ANOVA by default because of its demonstrated ability to maintain appropriate Type I error rates and offer comparable or superior statistical power to classical tests across a wide range of conditions.

When significant differences are detected, the Games-Howell post-hoc procedure provides pairwise comparisons that adjust both for heteroscedasticity and unequal sample sizes, producing confidence intervals and adjusted p-values that control for Type I error without requiring preliminary variance testing (Wilcox, 2017). This eliminates the reliance on homoscedasticity tests like Levene's or Bartlett's, which have limited power and may lead to inconsistent or inappropriate test selection (Delacre et al., 2017). A methodological analysis by (Ruxton & Beauchamp, 2008) further supports the use of the Games-Howell procedure when variances are unequal, recommending it over traditional methods such as

Tukey's Honestly Significant Difference (HSD) test. Here is an illustrative example of how to report the results of the two-step procedure in narrative form:

> To assess the performance of the presented algorithms, the number of function evaluations required to reach an ε-optimal solution was compared using Welch's ANOVA (Algorithm A: $n = 30$; B: $n = 27$; C: $n = 25$). The analysis revealed a significant overall difference ($F(2, 66.9) = 14.3;\ p < .001$). Group means were $M_A = 985.3;\ 95\%\ \text{CI}\ [944.6, 1026.0])$, $M_B = 1123.8;\ 95\%\ CI\ [954.4, 1188.2])$, and $M_C = 1165.9;\ 95\%\ \text{CI}\ [1100.5, 1231.3])$. Games–Howell post-hoc comparisons show that Algorithm A converged significantly faster than both B ($P = .004;\ ES = -138.5;\ 95\%\ \text{CI}\ [-235.4, -45.1]$) and C ($p < .001;\ -ES = -80.6;\ 95\%\ \text{CI}\ [-90.2, -70.5]$), while the difference between B and C was not statistically significant ($p = .42$). These results support the superiority of Algorithm A under the tested conditions.

The preceding example includes references to effect size (ES) which is covered in section I.5.5. In accordance with best practices outlined by (Lang and Secic, 2006), p-values are reported without leading zeroes and are rounded to two significant digits when $p > .01$, and to three significant digits when $p \leq .01$, unless $p < .001$, in which case it is reported as "$p < .001$" to avoid overstating precision.

For non-normal or skewed data (i.e., Category 2), the MDAF suggests using the Mann-Whitney U test for the two-sample case, or the Kruskal-Wallis H test for multiple samples. The latter is a nonparametric method also known as the one-way ANOVA on ranks or the Kruskal-Wallis test by ranks. Unlike parametric approaches such as Welch's ANOVA, the Kruskal-Wallis test makes no assumptions about the underlying distribution of the data. Instead, it assesses whether the distributions of the groups are identical in shape and location under the null hypothesis. If the distributions differ only in central tendency, then the test can be interpreted as a test of median differences.

A statistically significant test result suggests that at least one algorithm differs in performance, but it does not identify which pairs are different. This can be addressed using

the Dunn post-hoc test for pairwise comparisons, which evaluates all pairwise group differences using ranked data. Because multiple comparisons inflate the risk of Type I errors, the Dunn test is typically combined with adjustments for multiple testing, such as the Bonferroni, Holm-Bonferroni, or Benjamini-Hochberg procedures. These adjustments control the familywise error rate (FWER) or false discovery rate (FDR), depending on the context, and ensure that the conclusions drawn from multiple pairwise differences remain statistically valid. The MDAF recommends using the Holm-Bonferroni procedure since it maintains greater statistical power than the classic Bonferroni correction and explicitly controls the FWER rather than the FDR, unlike the Benjamini-Hochberg procedure (Bender & Lange, 2001). When reported, the adjusted p-values must be clearly identified. Here is an illustrative example in narrative form:

> As the performance distributions were non-symmetric ($|Sk| = 0.65 > 0.5$), the median number of function evaluations (feval) to reach an ε-optimal ($\varepsilon = 10^{-3}$) solution was used to summarize the performance of each algorithm with $n = 250$ runs each:

Table-A I-10 Performance evaluation results in number of function evaluations

Algorithm	Median [95% CI]	IQR
A	985 [945, 1020]	940 – 1010
B	1125 [1080, 1180]	1080 – 1170
C	1170 [1110, 1250]	1110 – 1230

> The Kruskal-Wallis H test revealed a statistically significant difference in performance, $H(2) = 14.1,\ p < .001$, indicating that at least one algorithm's performance distribution differed from the others. To identify which specific pairs of algorithms were responsible for the observed difference, a Dunn post-hoc test was conducted with Holm–Bonferroni adjustments for multiple comparisons. The results showed that Algorithm A converged significantly faster than both Algorithm B ($p =$

.004, adjusted; $ES = -138$; 95% CI [-145, -112]) and Algorithm C ($p < .001$, adjusted; $ES = -150$; 95% CI [-190, -70]), while the difference between B and C was not statistically significant ($p = .23$, adjusted). The results support the conclusion that Algorithm A consistently required fewer evaluations to reach the target solution under the tested conditions.

Although the Mann-Whitney U test and the Kruskal-Wallis H test typically have less statistical power than parametric alternatives such as Welch's ANOVA, their robustness to violations of normality makes them more suitable choices when the data are not normally distributed.

In summary, the MDAF recommends selecting statistical tests based on the distributional shape using skewness: approximately normal data ($|Sk| < 0.5$) vs. skewed data ($|Sk| \geq 0.5$). For symmetrical data, use Welch's t-test in the two-sample case, or Welch's ANOVA followed by Games–Howell post-hoc tests for pairwise comparisons. For non-normal data, use the Mann-Whitney U test in the two-sample case, or the Kruskal-Wallis H test followed by Dunn post-hoc tests and Holm-Bonferroni correction for multiple comparisons. These recommendations are shown schematically in Figure-A I-31 below.

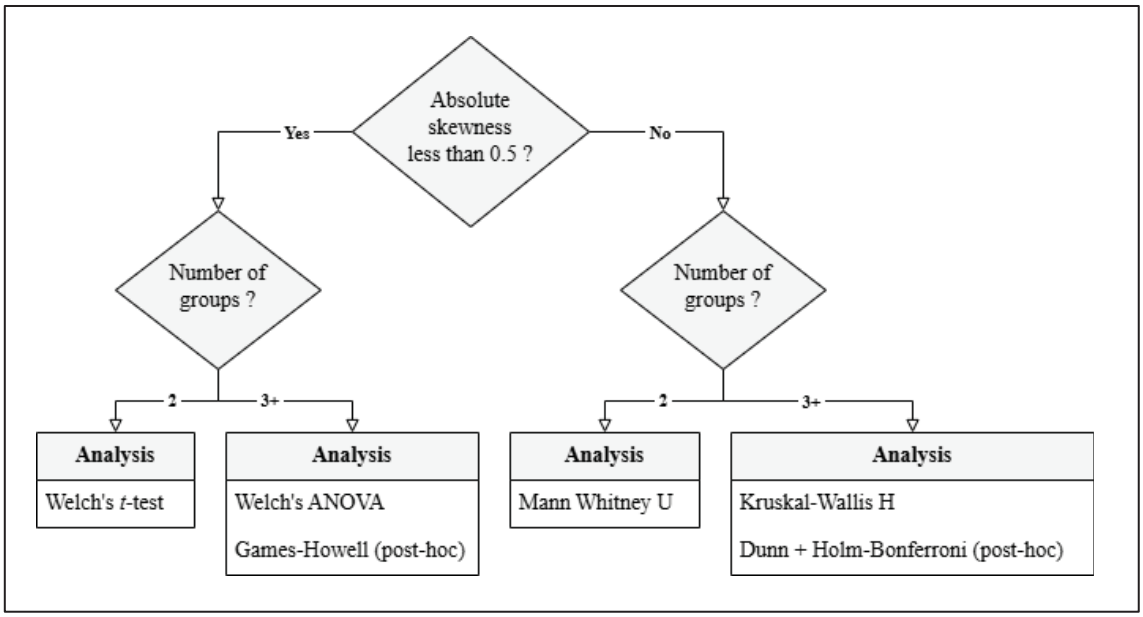

Figure-A I-31 Decision tree for selecting statistical tests based of skewness and number of groups

These procedures are available in R and SPSS.

I.5.4 Confidence intervals

Statistical testing concerns the probability of obtaining the observed results assuming the null hypothesis is true. Unlike p-values, confidence intervals (CIs) provide information about precision. More specifically, CIs give a range of plausible values in the form of confidence limits defined by lower and upper limits that correspond to a specified level of confidence $(1 - \alpha)$. Authoritative sources cited in this thesis (e.g., APA, SAMPL, etc.) consistently recommend their inclusion to indicate measurement error or uncertainty. For this reason, the MDAF recommends that CIs be reported when a point estimate is presented (section I.5.1) in the context of hypothesis testing (section I.5.3) and effect size estimation (section I.5.5). This includes estimates of means, medians, proportions, differences between groups, correlation coefficients, and model parameters (e.g., regression coefficients).

To be clear, the MDAF does not recommend reporting confidence intervals for all estimates unless variability is a specific focus of the analysis. Instead, confidence intervals should be reserved for point estimates that inform comparisons or inference, such as means, medians,

differences, proportions, or effect sizes. For example, it is rarely useful to report CIs about standard deviations since researchers are usually more interested in whether group means or medians differ, not whether variances differ.

The MDAF encourages researchers to compute confidence intervals using bootstrap resampling, especially when the underlying distribution of the estimator is unknown, non-normal, or difficult to model analytically. Unlike classical methods based on parametric assumptions (e.g., t-distribution or normal approximations), bootstrapping makes minimal distributional assumptions, thereby increasing robustness in skewed or heteroscedastic conditions which are common in metaheuristics research. Furthermore, bootstrap methods are applicable to a wide range of estimators, including medians, effect sizes, and model performance measures, where closed-form confidence intervals may not be available or valid.

Research shows that elementary bootstrap techniques perform reliably in diverse contexts (Davison & Hinkley, 2014). One such technique is the percentile bootstrap which constructs confidence intervals by taking the appropriate percentiles (e.g., 2.5th and 97.5th for a 95% CI) directly from the empirical distribution of bootstrap estimates. A bootstrap estimate is a value of a statistic (e.g., mean, median, effect size) computed from a resampled version of the original dataset, where the resampling is done with replacement. For example, given an original dataset F of size n, draw a bootstrap sample F' by sampling n observations with replacement from F. Compute the statistics of interest (e.g., the mean) from F' and add this value to a new distribution of estimates, denoted G. Repeat this process many times (typically 10,000) to approximate the sampling distribution of the statistic. Finally, extract the $\alpha/2$ and $1-\alpha/2$ percentiles from G to define the lower and upper bounds of the $(1-\alpha)\times 100\%$ confidence interval.

While the simple percentile approach works, it can underperform when the statistic is biased or when the sampling distribution is skewed. For this reason, the MDAF recommends using the bias-corrected and accelerated (BCa) bootstrap (Efron & Tibshirani, 1993). This method improves upon the simple percentile approach by adjusting the confidence interval limits

based on two quantities: (1) a bias correction factor which accounts for systematic differences between the observed estimate and the center of the bootstrap distribution, and (2) an acceleration factor, which adjusts for changes in variability across the range of estimates. These adjustments result in intervals that better reflect the true uncertainty of the estimate, particularly in small samples or when the distribution is skewed.

As illustrated in previous sections, a typical format for reporting confidence intervals is $M = 110$ (95% CI $[105, 115]$), but $M = 110$, 95% CI $[105, 115]$ is also common. In table format, it is recommended to include the CIs along with the point estimates as in Table-A I-11 below.

Table-A I-11 Illustrative example of reporting a confidence interval in a table

Algorithm	**Median [95% CI]**	**IQR**
A	985 [945, 1020]	940 – 1010
B	1125 [1080, 1180]	1080 – 1170
C	1170 [1110, 1250]	1110 – 1230

In this example, the units of measurement would be placed in the title of the table. In addition, researchers should clearly specify the method used to compute confidence intervals (e.g., percentile bootstrap, BCa bootstrap, or analytical approximation), along with the number of bootstrap replications. For example, the following note could be placed under Table-A I-11 or in a footnote:

> Values in brackets represent 95% confidence intervals calculated using the bias-corrected and accelerated (BCa) bootstrap method with 10,000 replications.

In summary, confidence intervals enhance p-values by capturing the uncertainty around point estimates. The MDAF underscores the importance of reporting CIs to promote transparency and practical insight and recommends using the BCa bootstrap technique to

account for skewness. The BCa bootstrap procedure is available as a standalone procedure such as in R and in SPSS, though it requires a paid add-on in the latter.

I.5.5 Effect size

While statistical significance reflects the likelihood of a result under the null hypothesis and confidence intervals indicate precision, neither conveys the magnitude or practical relevance of the effect. For example, an experiment may reveal a statistically significant difference between two algorithms even when the actual size of the difference is trivially small. For this reason, the MDAF recommends reporting effect sizes alongside confidence intervals and p-values, as promoted by the APA, the SAMPL guidelines, and the TFSI.

The MDAF emphasizes the use of absolute effect sizes, such as differences in means, medians, or percentages, expressed in the original measurement units (e.g., number of function evaluations or success rate). These are more intuitive and meaningful than standardized measures like Cohen's d, η^2, or Cliff's delta, which are harder to interpret and depend on assumptions such as homoscedasticity or normality. For example, reporting that "Algorithm A required 138 fewer evaluations on average than Algorithm B" provides clearer insight than stating that "Cohen's d = 0.82*"*, especially for application-minded audiences.

To estimate the effect size between two algorithms without relying on parametric assumptions, the MDAF adopts a bootstrap-based approach with BCa correction. Specifically, pairs of values are repeatedly sampled with replacement from each distribution, and the difference between each pair is recorded to form a bootstrap distribution of effect sizes. The mean of this distribution represents the estimated average effect size, while the BCa-corrected percentiles (e.g., 2.5^{th} and 97.5^{th}) define a confidence interval. This method offers a robust, nonparametric estimation of effect size and uncertainty, consistent with the MDAF's emphasis on transparency and practical interpretability in empirical comparisons.

The suggested reporting format is the same as for other parameters and should include CIs : $ES = -110$ (95% CI $[-115, -105]$) or $ES = -110$, 95% CI $[-115, -105]$. In table format, the effect sizes and their confidence intervals may be reported as in Table-A I13 below.

Table-A I-12 Illustrative example of reporting effect sizes

Comparison	Effect size [95% CI]	*p*
A vs. B	-140 [-235, -45]	< .001
A vs. C	-180 [-290, -70]	< .001
B vs. C	-40 [-115, 30]	.35

In this example, the units of measurement would be placed in the title of the table. In addition to raw effect sizes, narrative clarifications should be provided in the text to help the reader understand the practical significance of such results. Although researchers are generally encouraged to interpret their effect sizes considering prior results, (Gagnon, Abran & April, 2025) found that only 3.0% (95% CI [2.6, 3.4]) of studies in the metaheuristics literature report them. References like (Hatcher, 2013) recommend using the general interpretations given in (Cohen, 2013), but these only apply to the standardized effect sizes referred to earlier. To reduce confusion linked to the variety of effect size measures, the MDAF continues to promote reporting effect sizes in their original units. To remedy the persistent problem of interpretability, effect sizes may also be expressed as percentage differences, offering a scale-independent and intuitive representation of magnitude that is easier to understand and communicate in applied contexts.

This procedure is available in external software such as R via the boot package, and SPSS, although SPSS requires both a paid module and use of its relatively uncommon scripting language.

I.6 Conclusion

The MDAF offers a comprehensive and structured approach to conducting rigorous, transparent, and reproducible research in metaheuristics. From the formulation of research questions to the reporting of results, the MDAF provides concrete recommendations that integrate best practices from both experimental methodology and software engineering.

At the design stage (section I.3.1), the MDAF encourages researchers to frame their research questions using established models such as FINER and PICO, ensuring that they are feasible, interesting, novel, explicit, and relevant. It further supports the development of high-quality objectives through the SMAR criteria (specific, measurable, achievable, and related) along with secondary objectives aimed at exploring mechanisms behind observed effects. This structured approach promotes clear variable definitions, logical hypothesis formulation, and robust experimental protocols. To address the common challenge of sample size determination in metaheuristics, where analytical solutions are rarely available, the MDAF proposes a default of $n = 250$ per group, supported by both theoretical justification and simulation-based evidence. It also highlights the importance of tailoring sample sizes to account for unequal variances or expected effect sizes across groups and recommends flexible alternatives such as iterative or simulation-based designs to improve statistical efficiency. To ensure fair algorithm comparisons (section I.3.3), the MDAF promotes systematic parameter tuning (e.g., grid search, rank-sum methods) and encourages the integration of DOE principles to explore interactions and assess robustness.

In terms of implementation, the framework advocates for version control (section I.4.1), comprehensive documentation (section I.4.2), modular code design (section I.4.3), and automated testing (section I.4.4) to maintain quality and reproducibility throughout the research lifecycle.

The guidelines also promote the use of study-specific statistical methods for validating experimental results and drawing meaningful insights. Descriptive statistics (section I.5.1) are presented as essential for summarizing experimental results and communicating them clearly, with specific recommendations for reporting central tendency, variability, and five-point summaries, as well as addressing the current underreporting in the literature based on (Gagnon, Abran & April, 2025). Graphical representations (section I.5.2) are equally emphasized as crucial complements to numerical summaries, providing intuitive visual insights into patterns, distributional characteristics, and improving interpretability.

The MDAF's approach to statistical significance testing is detailed in section I.5.3, outlining a structured method for evaluating observed differences by selecting appropriate tests based on data distribution (e.g., Welch's t-test or Welch's ANOVA for symmetrical data, Mann-Whitney U or Kruskal-Wallis H for skewed data). Crucially, the MDAF aligns with best practices by stressing that statistical significance alone is insufficient without accompanying CIs (section I.5.4) and effect sizes (section I.5.5). CIs are presented as vital for capturing the uncertainty around point estimates, with the MDAF recommending bootstrap methods like the BCa bootstrap. Finally, effect size reporting is strongly advocated, with a preference for absolute measures in original measurement units to enhance practical interpretability, also using bootstrap-based approaches for robust estimation. These practices are designed to complement one another, forming a coherent strategy for deriving and communicating meaningful insights.

Together, these components establish a unified framework that integrates methodological reliability with software best practices. When adopted collectively and early in the research process, the MDAF enables the development of computational studies that are scientifically robust. By supporting principled experimental design and rigorous analysis, the MDAF contributes to elevating the empirical standards in metaheuristics research and increasing the field's overall scientific impact.

APPENDIX II

Minimum Research Receivability Criteria

A. **Novel:** The technical features[1] of the proposed invention must not be found in any previous public disclosure (e.g., research article). In the context of research, it is also allowed to confirm, refute or extend previous results.

B. **Useful:** The proposed invention must have a practical utility by solving one or more specific[2] and identifiable[3] problems.

C. **Nonobvious:** The invention must not be considered obvious to someone with ordinary skill in art. It should also not be attainable with the straightforward combination of previous disclosures.

D. **Proven:** The invention's usefulness must be demonstrated through empirical evidence, including adequate statistical analysis.

E. **Metaphor-free:** The invention must be presented in a technically appropriate language that is devoid of metaphors.

[1] Technical features are those that pertain to the technical character of the invention, including its structure, function, operation, or effect.

[2, 3] Specific in the sense of being fully detailed and identifiable as in "concrete" as opposed to abstract or vague.

APPENDIX III

Publication Figures and Tables Checklist

Necessity and relevance

- ☐ Does the figure or table significantly contribute to illustrating key concepts, results, conclusions, or methodologies?

Readability

- ☐ Is the figure or table simple, clear, and free of extraneous detail (e.g., decorative elements, overlapping text)?

Integration and reference

- ☐ Is the figure or table mentioned in the text, with references contextualized within an analytical narrative?

Results and analysis

- ☐ Does the presented statistical data, such as performance comparisons or convergence rates, include appropriate uncertainty measures (e.g., error bars, confidence intervals)?

Reproducibility

- ☐ Are the data sources and any software or tools used for generating the figure or table mentioned, and are they made available to the reader when possible (e.g. publisher's platform, GitHub)?

APPENDIX IV

Analysis Checklist

Necessity and relevance

- ☐ Does the proposed comparison or benchmarking address a significant research question, hypothesis or practical problem?

Experimental design

- ☐ Are the parameters and configurations for each algorithm reported in detail?
- ☐ Has either automated parameter tuning, or parameter optimization been used (e.g., grid search)?
- ☐ Is a comparison made with either a baseline and/or other relevant results (e.g., random search, a modern variant of another algorithm)?
- ☐ Are sample sizes equal among groups and larger than 30?

Performance measures

- ☐ Are performance measures robust and machine-independent (e.g., median hitting time measured in number of objective function evaluations)?
- ☐ Are statistical comparisons made (e.g., null hypothesis testing, Bayesian testing)?
- ☐ Are the practical implications of the results discussed?
- ☐ Are technical reasons for the observed differences (or lack thereof) given?

APPENDIX V

Descriptive Statistics Checklist

	Item	Description
☐	**Sample size**	Clearly reported for each algorithm or configuration and condition.
☐	**Central tendency**	Mean (M) for normal data or median (Md) for non-normal data.
☐	**Dispersion / variability**	Standard deviation (SD) for normal data or interquartile range (IQR) for non-normal data.
☐	**Five-point summary (non-normal data)**	Includes Min, $Q1$, Median ($Q2$), $Q3$, and Max. Reported in tabular form for clarity.
☐	**Justification for method choice**	Choice of summary statistics (mean vs. median) is justified based on distributional characteristics.
☐	**Units and precision**	Units are specified, and values are rounded appropriately (e.g., "10.5 s" not "10.45125 s").
☐	**Raw counts with percentages**	When percentages are reported, raw counts are also given (e.g., "3 out of 10 (30%)").
☐	**Abbreviation consistency**	Standard abbreviations (M, Md, SD, etc.) are used.
☐	**Bivariate summaries (if applicable)**	Contingency tables are used to describe multivariate relationships.
☐	**Association measures**	Phi coefficient (φ) for 2×2 contingency tables or Cramér's V for larger ones. Pearson's r for continuous variables. Interpretation is supported with effect size thresholds.
☐	**Reporting per instance**	Results are reported per benchmark function or problem instance, not only in aggregate.

BIBLIOGRAPHY

Alatas, B., Akin, E., & Ozer, A. B. (2009). Chaos embedded particle swarm optimization algorithms. *Chaos, Solitons & Fractals, 40*(4), 1715–1734. https://doi.org/10.1016/j.chaos.2007.09.063

AMA Manual of Style Committee. (2020). *AMA manual of style: A guide for authors and editors* (11th ed.). Oxford University Press. https://doi.org/10.1093/jama/9780190246556.001.0001

American Psychological Association. (2020). *Publication manual of the American Psychological Association* (7th ed.). American Psychological Association. https://doi.org/10.1037/0000165-000

Aranha, C., Camacho-Villalón, C. L., Campelo, F., Dorigo, M., Ruiz, R., Sevaux, M., Sörensen, K., & Stützle, T. (2021). Metaphor-based metaheuristics: A call for action—The elephant in the room. *Swarm Intelligence, 16*(1), 1–6. https://doi.org/10.1007/s11721-021-00202-9

Barr, R. S., Golden, B. L., Kelly, J. P., Resende, M. G. C., & Stewart, W. R. (1995). Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics, 1*(1), 9–32. https://doi.org/10.1007/BF02430363

Bartz-Beielstein, T., Doerr, C., van den Berg, D., Bossek, J., Chandrasekaran, S., Eftimov, T., & Weise, T. (2020). Benchmarking in optimization: Best practice and open issues. *arXiv*. https://arxiv.org/abs/2007.03488

Bastos Filho, C. J., de Lima Neto, F. B., Lins, A. J., Nascimento, A. I., & Lima, M. P. (2008). A novel search algorithm based on fish school behavior. In *Proceedings of the 2008 IEEE International Conference on Systems, Man and Cybernetics* (pp. 2646–2651). IEEE.

Bender, R., & Lange, S. (2001). Adjusting for multiple testing—When and how? *Journal of Clinical Epidemiology, 54*(4), 343–349. https://doi.org/10.1016/S0895-4356(00)00314-0

Benítez-Hidalgo, A., Nebro, A. J., García-Nieto, J., Oregi, I., & Del Ser, J. (2019). jMetalPy: A Python framework for multi-objective optimization with metaheuristics. Swarm and Evolutionary Computation, 51, 100598. https://doi.org/10.1016/j.swevo.2019.100598

Beyer, H. G., & Finck, S. (2012). HappyCat—A simple function class where well-known direct search algorithms do fail. In *Proceedings of the International Conference on Parallel Problem Solving from Nature* (pp. 367–376). Springer.

Birattari, M., & Dorigo, M. (2006). How to assess and report the performance of a stochastic algorithm on a benchmark problem: Mean or best result on a number of runs? *Optimization Letters, 1*(3), 309–311. https://doi.org/10.1007/s11590-006-0011-8

Birattari, M., Stützle, T., Paquete, L., & Varrentrapp, K. (2002). A racing algorithm for configuring metaheuristics. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002)*.

Birattari, M., Yuan, Z., Balaprakash, P., & Stützle, T. (2010). F-race and iterated F-race: An overview. In *Experimental methods for the analysis of optimization algorithms* (pp. 311–336). Springer. https://doi.org/10.1007/978-3-642-02538-9_13

Bishop, J. M. (1989). Stochastic searching networks. In *Proceedings of the 1st IEE International Conference on Artificial Neural Networks* (Conference Publication No. 313, pp. 329–331). IET.

Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences, 237*, 82–117.

Brian Haynes, R. (2006). Forming research questions. *Journal of Clinical Epidemiology, 59*(9), 881–886. https://doi.org/10.1016/j.jclinepi.2006.06.006

Brown, M. B., & Forsythe, A. B. (1974). Robust tests for the equality of variances. *Journal of the American Statistical Association, 69*(346), 364–367. https://doi.org/10.1080/01621459.1974.10482955

Browner, W. S., Newman, T. B., Cummings, S. R., & Grady, D. G. (2022). *Designing clinical research*. Lippincott Williams & Wilkins.

Brownlee, J. (2007). A note on research methodology and benchmarking optimization algorithms. *Complex Intelligent Systems Laboratory, Centre for Information Technology Research, Faculty of Information and Communication Technologies, Swinburne University of Technology*. Technical report 70125.

Cahon, S., Melab, N., & Talbi, E. (2004). PARADISEO: A framework for the reusable design of parallel and distributed metaheuristics. *Journal of Heuristics, 10*(3), 357–380. https://doi.org/10.1023/B:HEUR.0000026900.92269.ec

Cai, J., Ma, X., Li, L., Yang, Y., Peng, H., & Wang, X. (2007). Chaotic ant swarm optimization to economic dispatch. *Electric Power Systems Research, 77*(10), 1373–1380. https://doi.org/10.1016/j.epsr.2006.10.006

Camacho-Villalón, C. L., Dorigo, M., & Stützle, T. (2022). Exposing the grey wolf, moth-flame, whale, firefly, bat, and antlion algorithms: Six misleading optimization techniques inspired by bestial metaphors. *International Transactions in Operational Research, 30*(6), 2945–2971. https://doi.org/10.1111/itor.13176

Camacho-Villalón, C. L., Stützle, T., & Dorigo, M. (2023). Designing new metaheuristics: Manual versus automatic approaches. *Intelligent Computing, 2*. https://doi.org/10.34133/icomputing.0048

Campelo, F., & Aranha, C. (2021). Sharks, zombies and volleyball: Lessons from the evolutionary computation bestiary. In *Proceedings of the LIFELIKE Computing Systems Workshop 2021*. CEUR-WS.

Chen, L., & Aihara, K. (1995). Chaotic simulated annealing by a neural network model with transient chaos. *Neural Networks, 8*(6), 915–930. https://doi.org/10.1016/0893-6080(95)00033-V

Chiarandini, M., Basso, D., & Stützle, T. (2005). Statistical methods for the comparison of stochastic optimizers. *International Transactions in Operational Research*, 189–196. https://www.imada.sdu.dk/~marco/Publications/Files/032_chiarandini.pdf

Clerc, M. (2012). Standard particle swarm optimisation. *HAL*. https://hal.science/hal-00764996

Clerc, M. (2015). *Guided randomness in optimization* (Vol. 1). John Wiley & Sons.

Cohen, J. (2013). *Statistical power analysis for the behavioral sciences*. Routledge. https://doi.org/10.4324/9780203771587

Colorni, A., Dorigo, M., & Maniezzo, V. (1992). Distributed optimization by ant colonies. In *Proceedings of the First European Conference on Artificial Life* (Vol. 142, pp. 134–142).

Cundill, B., & Alexander, N. D. (2015). Sample size calculations for skewed distributions. *BMC Medical Research Methodology, 15*(1), Article 23. https://doi.org/10.1186/s12874-015-0023-0

Mathews, D., & Clark, J. M. (2007, July 25). *Successful students' conceptions of mean, standard deviation, and the Central Limit Theorem* [Unpublished manuscript]. Central Michigan University & Hollins University.

Davison, A. C., & Hinkley, D. V. (2014). *Bootstrap methods and their application*. Cambridge University Press.

De Jong, K. A. (1975). *Analysis of the behavior of a class of genetic adaptive systems* (Doctoral dissertation). University of Michigan.

Delacre, M., Lakens, D., & Leys, C. (2017). Why psychologists should by default use Welch's t-test instead of Student's t-test. *International Review of Social Psychology, 30*(1), 92–101. https://doi.org/10.5334/irsp.82

Dorigo, M. (1992). *Optimization, learning and natural algorithms* (Doctoral dissertation). Politecnico di Milano.

Dorigo, M., & Stützle, T. (2004). *Ant colony optimization*. MIT Press.

Duman, E., Uysal, M., & Alkaya, A. F. (2012). Migrating birds optimization: A new metaheuristic approach and its performance on quadratic assignment problem. *Information Sciences, 217*, 65–77.

Durillo, J. J., Nebro, A. J., & Alba, E. (2010). The jMetal framework for multi-objective optimization: Design and architecture. In *Proceedings of the IEEE Congress on Evolutionary Computation* (pp. 1–8). IEEE. https://doi.org/10.1109/CEC.2010.5586354

Eberhart, R., & Kennedy, J. (1995). Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks* (Vol. 4, pp. 1942–1948).

Efron, B., & Tibshirani, R. J. (2006). *An introduction to the bootstrap*. Chapman & Hall/CRC.

Fausto, F., Reyna-Orta, A., Cuevas, E., Andrade, Á. G., & Perez-Cisneros, M. (2019). From ants to whales: Metaheuristics for all tastes. *Artificial Intelligence Review, 53*(1), 753–810. https://doi.org/10.1007/s10462-018-09676-2

Field, A., & Hole, G. (2011). *How to design and report experiments*. Sage.

Gagnon, I., April, A., & Abran, A. (2020). A critical analysis of the bat algorithm. *Engineering Reports, 2*(8), e12212. https://doi.org/10.1002/eng2.12212

Gandomi, A. H., & Yang, X.-S. (2014). Chaotic bat algorithm. *Journal of Computational Science, 5*(2), 224–232. https://doi.org/10.1016/j.jocs.2013.10.002

Gandomi, A. H., Yang, X.-S., Talatahari, S., & Alavi, A. H. (2013). Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation, 18*(1), 89–98. https://doi.org/10.1016/j.cnsns.2012.06.009

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., & Wilson, A. G. (2018). Loss surfaces, mode connectivity, and fast ensembling of DNNs. In *Advances in Neural Information Processing Systems* (pp. 8789–8798).

Geem, N. Z. W., Kim, N. J. H., & Loganathan, G. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation, 76*(2), 60–68. https://doi.org/10.1177/003754970107600201

Gent, I. P., et al. (1994). *How not to do it*. University of Edinburgh, Department of Artificial Intelligence.

Geoffrion, A. M. (1976). The purpose of mathematical programming is insight, not numbers. *Interfaces, 7*(1), 81–92. https://doi.org/10.1287/inte.7.1.81

Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences, 8*(1), 156–166. https://doi.org/10.1111/j.1540-5915.1977.tb01074.x

Glover, F., & McMillan, C. (1986). The general employee scheduling problem: An integration of MS and AI. *Computers & Operations Research, 13*(5), 563–573. https://doi.org/10.1016/0305-0548(86)90050-X

Hatcher, L. (2013). Central tendency, variability, and descriptive statistics. In *Advanced statistics in research: Reading, understanding, and writing up data analysis results* (pp. 64–66). ShadowFinch Media LLC.

Halim, A. H., Ismail, I., & Das, S. (2020). Performance assessment of the metaheuristic optimization algorithms: An exhaustive review. *Artificial Intelligence Review, 54*(3), 2323–2409. https://doi.org/10.1007/s10462-020-09906-6

Hamming, R. W. (1962). *Numerical methods for scientists and engineers*. McGraw-Hill.

Haynes, R. B., Mulrow, C. D., Huth, E. J., Altman, D. G., & Gardner, M. J. (1990). More informative abstracts revisited. *Annals of Internal Medicine, 113*(1), 69–76. https://doi.org/10.7326/0003-4819-113-1-69

Huang, C., Li, Y., & Yao, X. (2020). A survey of automatic parameter tuning methods for metaheuristics. *IEEE Transactions on Evolutionary Computation, 24*(2), 201–216. https://doi.org/10.1109/TEVC.2019.2921598

Hutter, F., Hoos, H. H., Leyton-Brown, K., & Stützle, T. (2009). ParamILS: An automatic algorithm configuration framework. *Journal of Artificial Intelligence Research, 36*, 267–306. https://doi.org/10.1613/jair.2861

Holland, J. H. (2019). *Adaptation in natural and artificial systems*. MIT Press.

Hooker, J. N. (1995). Testing heuristics: We have it all wrong. *Journal of Heuristics, 1*(1), 33–42.

IEEE Computer Society. (2024). *Guide to the software engineering body of knowledge (SWEBOK® Guide)* (Version 4.0; H. Washizaki, Ed.).

Jakobsen, L., Brinkløv, S., & Surlykke, A. (2013). Intensity and directionality of bat echolocation signals. *Frontiers in Physiology, 4*, Article 89. https://doi.org/10.3389/fphys.2013.00089

Johnson, D. S. (2002). A theoretician's guide to the experimental analysis of algorithms. In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* (Vol. 59, pp. 215–250). American Mathematical Society. https://doi.org/10.1090/dimacs/059/11

Kalra, M., & Singh, S. (2015). A review of metaheuristic scheduling techniques in cloud computing. *Egyptian Informatics Journal, 16*(3), 275–295.

Kendall, G., et al. (2016). Good laboratory practice for optimization research. *Journal of the Operational Research Society, 67*(4), 676–689. https://doi.org/10.1057/jors.2015.77

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks* (Vol. 4, pp. 1942–1948). IEEE.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680. https://doi.org/10.1126/science.220.4598.671

LaTorre, A., et al. (2021). A prescription of methodological guidelines for comparing bio-inspired optimization algorithms. *Swarm and Evolutionary Computation, 67*, 100973. https://doi.org/10.1016/j.swevo.2021.100973

Mackridge, A., & Rowe, P. (2018). Presenting and summarizing data. In *A practical approach to using statistics in health research: From planning to reporting* (pp. 18–20). Wiley. https://doi.org/10.1002/9781119383628.ch3

Maulana, A. (2018). *Many objective optimization and complex network analysis* (Doctoral dissertation).

Millonas, M. M. (1993). Swarms, phase transitions, and collective intelligence. *arXiv*. https://arxiv.org/abs/adap-org/9306002

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51–67.

Mitić, M., Vuković, N., Petrović, M., & Miljković, Z. (2015). Chaotic fruit fly optimization algorithm. *Knowledge-Based Systems, 89*, 446–458. https://doi.org/10.1016/j.knosys.2015.08.010

Lang, T. A., & Altman, D. G. (2015). Basic statistical reporting for articles published in biomedical journals: The SAMPL guidelines. *International Journal of Nursing Studies, 52*(1), 5–9. https://doi.org/10.1016/j.ijnurstu.2014.09.006

Lang, T. A., & Secic, M. (2006). *How to report statistics in medicine: Annotated guidelines for authors, editors, and reviewers*. American College of Physicians.

Lanzante, J. R. (2005). A cautionary note on the use of error bars. *Journal of Climate, 18*(17), 3699–3703. https://doi.org/10.1175/JCLI3499.1

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2018). Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research, 18*(185), 1–52.

López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M., & Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives, 3*, 43–58. https://doi.org/10.1016/j.orp.2016.09.002

McGeoch, C. C. (1996). Toward an experimental method for algorithm simulation. *INFORMS Journal on Computing, 8*(1), 1–15.

McGeoch, C. C. (2012). *A guide to experimental algorithmics*. Cambridge University Press.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Mirjalili, Seyedali. "The Ant Lion optimizer." *Advances in Engineering Software*, vol. 83, May 2015, pp. 80–98, https://doi.org/10.1016/j.advengsoft.2015.01.010.

Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems, 89*, 228–249. https://doi.org/10.1016/j.knosys.2015.07.006

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

Montgomery, D. C. (2021). *Design and analysis of experiments*. John Wiley & Sons.

Nourani, Y., & Andresen, B. (1998). A comparison of simulated annealing cooling strategies. *Journal of Physics A: Mathematical and General, 31*(41), 8373–8385.

Open Science Collaboration. (2015). Estimating the reproducibility of psychological science. *Science, 349*(6251). https://doi.org/10.1126/science.aac4716

Pagnozzi, F., & Stützle, T. (2019). Automatic design of hybrid stochastic local search algorithms for permutation flowshop problems. *European Journal of Operational Research, 276*(2), 409–421. https://doi.org/10.1016/j.ejor.2019.01.018

Parejo, J. A., Ruiz-Cortés, A., Lozano, S., & Fernandez, P. (2011). Metaheuristic optimization frameworks: A survey and benchmarking. *Soft Computing, 16*(3), 527–561. https://doi.org/10.1007/s00500-011-0754-8

Rardin, R. L., & Uzsoy, R. (2001). Experimental evaluation of heuristic optimization algorithms: A tutorial. *Journal of Heuristics, 7*(3), 261–304.

Rechenberg, I. (1965). *Cybernetic solution path of an experimental problem*. Royal Aircraft Establishment, Library Translation 1122.

Reeves, W. T. (1983). Particle systems—A technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics, 2*(2), 91–108.

Rere, L. R., Fanany, M. I., & Arymurthy, A. M. (2015). Simulated annealing algorithm for deep learning. *Procedia Computer Science, 72*, 137–144.

Ruxton, G. D. (2006). The unequal variance t-test is an underused alternative to Student's t-test and the Mann–Whitney U test. *Behavioral Ecology, 17*(4), 688–690. https://doi.org/10.1093/beheco/ark016

Ruxton, G. D., & Beauchamp, G. (2008). Time for some a priori thinking about post hoc testing. *Behavioral Ecology, 19*(3), 690–693. https://doi.org/10.1093/beheco/arn020

Sandve, G. K., Nekrutenko, A., Taylor, J., & Hovig, E. (2013). Ten simple rules for reproducible computational research. *PLoS Computational Biology, 9*(10), e1003285. https://doi.org/10.1371/journal.pcbi.1003285

Saremi, S., Mirjalili, S. M., & Mirjalili, S. (2014). Chaotic krill herd optimization algorithm. *Procedia Technology, 12*, 180–185. https://doi.org/10.1016/j.protcy.2013.12.473

Sayed, G. I., Tharwat, A., & Hassanien, A. E. (2019). Chaotic dragonfly algorithm: An improved metaheuristic algorithm for feature selection. *Applied Intelligence, 49*(1), 188–205. https://doi.org/10.1007/s10489-018-1261-8

Schwefel, H. P. (1965). Kybernetische evolution als strategie der experimentellen forschung in der strömungstechnik (Diploma thesis). Technical University of Berlin.

Sheskin, D. J. (2020). Handbook of parametric and nonparametric statistical procedures. Chapman & Hall/CRC.

Singh, C., Sharma, N., & Kumar, N. (2019). Analysis of software maintenance cost affecting factors and estimation models. International Journal of Scientific and Technology Research, 8(9), 276–281.

Sörensen, K. (2015). Metaheuristics—the metaphor exposed. International Transactions in Operational Research, 22(1), 3–18.

Sörensen, K., Sevaux, M., & Glover, F. (2017). A history of metaheuristics. arXiv. https://arxiv.org/abs/1704.00853

Suárez, P., et al. (2019). Make robots be bats: Specializing robotic swarms to the bat algorithm. Swarm and Evolutionary Computation, 44, 113–129. https://doi.org/10.1016/j.swevo.2018.01.005

Swan, J., Adriaensen, S., Bishr, M., Burke, E. K., Clark, J. A., De Causmaecker, P., & Yao, X. (2015). A research agenda for metaheuristic standardization. In Proceedings of the XI Metaheuristics International Conference (pp. 1–3).

Swan, J., Adriaensen, S., Brownlee, A. E. I., Hammond, K., Johnson, C. G., Kheiri, A., Krawiec, F., Merelo, J. J., Minku, L. L., Özcan, E., Pappa, G. L., García-Sánchez, P., Sörensen, K., Voß, S., Wagner, M., & White, D. R. (2022). Metaheuristics "in the large". *European Journal of Operational Research, 297*(2), 393–406. https://doi.org/10.1016/j.ejor.2021.05.042

Tzanetos, A., & Dounias, G. (2021). Nature inspired optimization algorithms or simply variations of metaheuristics? Artificial Intelligence Review, 54(3), 1841–1862.

Utochkin, I. S. (2015). Ensemble summary statistics as a basis for rapid visual categorization. Journal of Vision, 15(4), Article 8. https://doi.org/10.1167/15.4.8

Velasco, L., et al. (2023). A literature review and critical analysis of metaheuristics recently developed. Archives of Computational Methods in Engineering, 31(1), 125–146. https://doi.org/10.1007/s11831-023-09975-0

Wagner, S., & Affenzeller, M. (2005). HeuristicLab: A generic and extensible optimization environment. In Computer Aided Systems Theory – EUROCAST 2005 (pp. 538–541). Springer. https://doi.org/10.1007/3-211-27389-1_130

Wagner, S. (2009). Heuristic optimization software systems: Modeling of heuristic optimization algorithms in the HeuristicLab software environment (Doctoral dissertation). Johannes Kepler University Linz.

Walpole, R. E., Myers, R. H., Myers, S. L., & Ye, K. (2002). *Probability & statistics for engineers & scientists* (7th ed.). Prentice Hall.

Wang, G.-G., Deb, S., Gandomi, A. H., Zhang, Z., & Alavi, A. H. (2016). Chaotic cuckoo search. *Soft Computing, 20*(9), 3349–3362. https://doi.org/10.1007/s00500-015-1726-1

Weyland, D. (2010). 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, 1(2), 50–60.

Weyland, D. (2015). A critical analysis of the harmony search algorithm—How not to solve sudoku. Operations Research Perspectives, 2, 97–105.

Wilcox, R. R. (2021). Introduction to robust estimation and hypothesis testing. Elsevier.

Wilkinson, L. (1999). Statistical methods in psychology journals: Guidelines and explanations. American Psychologist, 54(8), 594–604. https://doi.org/10.1037/0003-066X.54.8.594

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 1(1), 67–82.

Xue, F., Cai, Y., Cao, Y., Cui, Z., & Li, F. (2015). Optimal parameter settings for bat algorithm. International Journal of Bio-Inspired Computation, 7(2), 125–128.

Yale Law School Roundtable on Data and Code Sharing. (2010). Reproducible research: Addressing the need for data and code sharing in computational science. *Computing in Science & Engineering, 12*(5), 8–13. https://doi.org/10.1109/MCSE.2010.113

Yang, X. S. (2008). Nature-inspired metaheuristic algorithms. Luniver Press.

Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization (NICSO 2010) (pp. 65–74). Springer.

Yuan, Z., Stützle, T., Montes de Oca, M. A., Lau, H. C., & Birattari, M. (2013). An analysis of post-selection in automatic configuration. In Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation (pp. 525–532). https://doi.org/10.1145/2463372.2463562