An assessment of the component-based view for metaheuristic research

by

Thimershen Achary
(216005583)

Submitted in fulfilment of the academic requirements of

Master of Science

in Computer Science

School of Mathematics, Statistics and Computer Science

College of Agriculture, Engineering and Science

University of KwaZulu-Natal

Westville

South Africa

February 2023

Supervisor: Mr. Anban W. Pillay

Co-supervisor: Dr. Edgar Jembere

PREFACE

The research contained in this dissertation was completed by the candidate while based in the Discipline of Computer Science, School of Mathematics, Statistics and Computer Science of the College of Agriculture, Engineering and Science, University of KwaZulu-Natal, Westville, South Africa. The research was financially supported by Centre for Artificial Intelligence Research.

The contents of this work have not been submitted in any form to another university and, except where the work of others is acknowledged in the text, the results reported are due to investigations by the candidate.

[redacted]

Signed: Anban Pillay

Date: 9 February 2023

DECLARATION 1: PLAGIARISM

Note that two declaration sections are required if there are papers emanating from the dissertation/thesis. The first (obligatory) declaration concerns plagiarism and the second declaration specifies your role in the published papers.

I, Thimershen Achary, declare that:

(i) the research reported in this dissertation, except where otherwise indicated or acknowledged, is my original work;

(ii) this dissertation has not been submitted in full or in part for any degree or examination to any other university;

(iii) this dissertation does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons;

(iv) this dissertation does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:

a) their words have been re-written but the general information attributed to them has been referenced;

b) where their exact words have been used, their writing has been placed inside quotation marks, and referenced;

(v) where I have used material for which publications followed, I have indicated in detail my role in the work;

(vi) this dissertation is primarily a collection of material, prepared by myself, published as journal articles or presented as a poster and oral presentations at conferences. In some cases, additional material has been included;

(vii) this dissertation does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the dissertation and in the References sections.

[redacted]

Signed: Thimershen Achary

Date: 9 February 2023

DECLARATION 2: PUBLICATIONS

My role in each paper and presentation is indicated.

1. Achary Thimershen and Pillay Anban W. 2022. A Taxonomy Guided Method to Identify Metaheuristic Components. Computational Science - ICCS 2023 - 22nd International Conference, London, UK, June 21-23, 2022, Proceedings, Part III. https://doi.org/10.1007/978-3-031-08757-8_41. Paper presented by Thimershen Achary.

2. Achary Thimershen, Pillay Anban W. and Jembere, Edgar. 2022. Towards Rigorous Foundations for Metaheuristic Research. Proceedings of the 14th International Joint Conference on Computational Intelligence, IJCCI 2022, Valletta, Malta, October 24-26, 2023. Paper presented by Thimershen Achary. https://doi.org/10.5220/0011552600003332. Paper presented by Thimershen Achary.

3. Achary Thimershen, Pillay Anban W. and Jembere, Edgar. 2023. A New Metaheuristic-Algorithm Similarity Measure Using Signal Flow Diagrams. Under review at: The Genetic and Evolutionary Computation Conference (GECCO 2023)

The research reported on in the above papers is based on work carried for the Master of Science degree. I designed and carried out all experiments, analysed data and reported the results. The co-authors, supervised the work and assisted with language editing.

[redacted]

Signed: Thimershen Achary

Date: 9 February 2023

ABSTRACT

Several authors have recently pointed to a crisis within the metaheuristic research field, particularly the proliferation of metaphor-inspired metaheuristics. Common problems identified include using non-standard terminology, poor experimental practices, and, most importantly, the introduction of purportedly new algorithms that are only superficially different from existing ones. These issues make similarity and performance analysis, classification, and metaheuristic generation difficult for both practitioners and researchers. A component-based view of metaheuristics has recently been promoted to deal with these problems. A component based view argues that metaheuristics are best understood in terms of their constituents or components. This dissertation presents three papers that are thematically centred on this view. The central problem for the component-based view is the identification of components of a metaheuristic. The first paper proposes the use of taxonomies to guide the identification of metaheuristic components. We developed a general and rigorous method, TAXONOG-IMC, that takes as input an appropriate taxonomy and guides the user to identify components. The method is described in detail, an example application of the method is given, and an analysis of its usefulness is provided. The analysis shows that the method is effective and provides insights that are not possible without the proper identification of the components. The second paper argues for formal, mathematically sound representations of metaheuristics. It introduces and defends a formal representation that leverages the component based view. The third paper demonstrates that a representation technique based on a component based view is able to provide the basis for a similarity measure. This paper presents a method of measuring similarity between two metaheuristic-algorithms, based on their representations as signal flow diagrams. Our findings indicate that the component based view of metaheuristics provides valuable insights and allows for more robust analysis, classification and comparison.

ACKNOWLEDGMENTS

I wish to thank my Mom (Kogie) and Dad (Dean) for all their love and support during the undertaking of this project.

I wish to thank my supervisors Mr. Anban W. Pillay and Dr. Edgar Jembere for all their guidance, support, and mentoring; without which, the completion of this project would not have been possible.

I wish to thank my grade 10, 11, and 12 Information Technology teacher, Mr. Padayachee, for the initial guidance that set me on the path of computer science, and who made computer programming an exhilarating experience.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Background

It is useful to consider metaheuristic research as comprising three inter-connected layers : the foundational layer, the experimental layer, and the application layer. The foundational layer deals with the philosophy of metaheuristics and the artefacts of the metaheuristic field as well as theory that is relevant to all sub-domains in the metaheuristic field. The experimental layer relies on the foundational layer for a theoretical basis on which experiments can be conducted to derive insight. The theoretical basis provides context to analyse any obtained experimental data within metaheuristic research. The application layer is a grey area, where scholarly work and work carried out by practitioners co-exist. Ideally, this layer should use the insight derived from the experimental layer as guidance when trying to solve real-world problems.

The work undertaken by this dissertation resides within the foundational layer of metaheuristic research. This was motivated by the many issues afflicting the metaheuristic research space that has received increased attention in literature. These issues afflicting the metaheuristic research include inconsistent metaphor usage, non-standard terminology [1, 2], and use of poor experimental setups, validation, and comparisons [1–3]. These factors have contributed to challenges in the field such as a proliferation of novel metaheuristics and 'novel' approaches being very similar to existing approaches [1, 2, 4].

Irresponsible metaphor usage is the use of sources of inspiration, e.g., nature, physics, and human behaviour, to be the most, if not the only, pivotal aspect to justify the algorithm as a "new" metaheuristic [1, 2]. These works include practices that obscure details by using non-standard terminology (terminology specific to the metaphor/inspiration used). Doing so adds to the challenge of positioning the proposed contribution in literature and may give the impression that the research output is novel. Symptoms of this activity are, according to [1, 2, 4–6], a flood of metaheuristics, numerous cases of very similar/overlapping work, lack of novelty, and according to [5] instances where inspirational source and algorithm behaviour are disconnected.

Researchers have also pointed to poor experimental practices. Reports such as [1, 2, 4, 7] suggest unfair and biased comparisons such as comparing new proposals to older metaheuristics instead of state-of-the-art and tweaking hyperparameters in favour of a metaheuristic to lift its performance above the rest. Comparative studies are not transparent enough, resulting in difficulties in extending past studies and existing data [1, 2]. A lack of proper motivation for selecting metaheuristics to compare is common [7]. There is also a lack of rigorous data analytics [1]. Competitive studies produce very little insight and do not answer or aid in answering the how and why [8, 9], yet comparative studies are still widely setup as competitive ones [1, 10].

The proliferation of metaphor-inspired metaheuristics is also a cause for concern. A GitHub project called the Evolutionary Computational Bestiary lists a vast and ever-growing number of bioinspired metaheuristics (with only a few exceptional bio-inspired metaheuristics being exempt) [10]. The aforementioned project opposes the flood of metaheuristics, especially the creation of new bio-inspired metaheuristics. Articles and other projects that criticize certain metaheuristic research trends are listed, some of which are intended to parody or ridicule the fact that these trends still exist.

1.1.2 Metaheuristics

The term 'meta-heuristic' was coined by Glover in [11], where the authors suggested that Tabu Search could be viewed as a metaheuristic "superimposed" on another heuristic. The suggestion is that metaheuristics operate on a higher level than heuristics. There have been several definitions of the term metaheuristic, each having some distinct attributes but as analyzed in [12], these definitions usually suggest that a metaheuristic is a higher-level strategy that guides subordinate heuristics with some auxiliary constituents in the mix such as information for the guiding process. Metaheuristics are described in [13] as frameworks that can be used to derive heuristic optimization algorithms and notes that, in literature, the frameworks and the heuristic optimization algorithms are both referred to as metaheuristics.

This study adopts the definition of metaheuristics provided in **[13]** that views metaheuristics as frameworks from which metaheuristic-algorithms are derived. An elaboration of why the distinction between framework and algorithm is essential can be found in **[14]**. It can be inferred that, often, a novelty at the algorithm level is hardly a significant feat. It is also the case that the term metaheuristics is used for both the optimization algorithm and the framework from which it is created; in order to know which is being studied, and the methods, resources, and theory needed to analyse it, a clear differentiation is needed. It would not be good practise to draw conclusions on a framework by analysing only one or two of the derived optimization algorithms.

An important concept in the field of metaheuristics is 'General Metaheuristic'. General metaheuristics, also known as general metaheuristic frameworks [8], unified metaheuristic frameworks [15], and generalized metaheuristic models [16] are used for tasks such as metaheuristic generation [16], performance analysis [17, 18], metaheuristic-similarity analysis [6], and classification of metaheuristics [7]. General metaheuristics are an abstraction of a set of metaheuristics, i.e., they are generalizations of the components, structure, and information utilized by a set of metaheuristics [18, 19]. General metaheuristics make use of a set of component-types, also referred to as general metaheuristics structures [18], component-categories [19], main ingredients [20], or key components [21].

1.1.3 Component-based view for metaheuristic research

A promising approach to mitigate the concerns described above is a component based view of metaheuristics. This component-based view for metaheuristic research is promoted in [1]. This view first requires differentiating between metaheuristics and metaheuristic-algorithms, thereafter drawing attention to the heuristic and structural components of metaheuristics and metaheuristic-algorithms, and implicitly away from the inspiration source. A component-based view is especially important for general metaheuristics, which rely a lot on the generalizations of components, structures and information used by metaheuristics.

Studies or contributions that aligned with the component-based view for metaheuristic research are not necessarily immune to the ill-effects that are currently affecting the metaheuristic research space. Even general metaheuristics that emphasises metaheuristic components are still susceptible to the challenges such as those resulting from non-standard terminology and metaphors that obfuscate details. In other words, the usage of general metaheuristics requires that components are properly identified. Thus, the identification of components takes on special importance for general metaheuristics, and the same could be said for any study aligning with the component-based view for metaheuristic research.

1.2 Problem Statement

Several researchers have thus proposed that a component-based view of metaheuristics that explicitly lists metaheuristic components, will assist in identifying novel components [1–5], promote component-based performance comparison and analyses, and facilitate component-wise selection of metaheuristics for comparative studies [1, 2, 7, 19]. However, these advantages have not been conclusively demonstrated. The central problems in this view of metaheuristics is the identification of components, their representation and how best to leverage these for analysis and comparisons. This work considers the following problems: a) how best to rigorously identify metaheuristic components b) how best to capture the component based view in a formal representation of metaheuristics and c) how well does this view assist in rigorous comparison studies.

1.3 Aims

The main aim of this work was to develop and evaluate representations of metaheuristics that leverage the component based view and to determine the effectiveness of the component based view for metaheuristics in solving recently identified foundational problems.

1.4 Objectives

- Develop a formal representation of metaheuristics that is aligned to the component based view.
- Develop a method for the identification of metaheuristic components.
- Develop a method that uses a component based view to measure similarity between metaheuristics.

1.5 Contributions

This dissertation presents three papers that are thematically centred on this view. The central problem for the component-based view is the identification of components of a metaheuristic.

- The first paper proposes the use of taxonomies to guide the identification of metaheuristic components. We developed a general and rigorous method, TAXONOG-IMC, that takes as input an appropriate taxonomy and guides the user to identify components. The method is described in detail, an example application of the method is given, and an analysis of its usefulness is provided. The analysis shows that the method is effective and provides insights that are not possible without the proper identification of the components.

- The second paper argues for formal, mathematically sound representations of metaheuristics. It introduces and defends a formal representation that leverages the component based view.
- The third paper demonstrates that a representation technique based on a component based view is able to provide the basis for a similarity measure. This paper presents a method of measuring similarity between two metaheuristic-algorithms, based on their representations as signal flow diagrams.

Our findings indicate that the component based view of metaheuristics provides valuable insights and allows for more robust analysis, classification and comparison.

1.6 Outline of dissertation structure

The rest of this dissertation is structured as follows:

Chapter 2 is a reprint of the following paper:

> Achary Thimershen and Pillay Anban W. 2022. A Taxonomy Guided Method to Identify Metaheuristic Components. Computational Science - ICCS 2023 - 22nd International Conference, London, UK, June 21-23, 2022, Proceedings, Part III. https://doi.org/10.1007/978-3-031-08757-8_41. Paper presented by Thimershen Achary.

Chapter 3 is a reprint of the following paper:

> Achary Thimershen, Pillay Anban W. and Jembere, Edgar. 2022. Towards Rigorous Foundations for Metaheuristic Research. Proceedings of the 14th International Joint Conference on Computational Intelligence, IJCCI 2022, Valletta, Malta, October 24-26, 2023. Paper presented by Thimershen Achary. https://doi.org/10.5220/0011552600003332. Paper presented by Thimershen Achary.

Chapter 4 is a reprint of the following paper:

> Achary Thimershen, Pillay Anban W. and Jembere, Edgar. 2023. A New Metaheuristic-Algorithm Similarity Measure Using Signal Flow Diagrams. Under review at: The Genetic and Evolutionary Computation Conference (GECCO 2023)

Chapter 5 concludes.

CHAPTER 2: Paper One: A Taxonomy Guided Method to Identify Metaheuristic Components

A Taxonomy Guided Method to Identify Metaheuristic Components

Thimershen Achary[1][0000-0002-6033-7065] and Anban W. Pillay[1][0000-0001-7160-6972]

[1]University of KwaZulu-Natal, Durban, South Africa
thimershenzn@gmail.com

Abstract. A component-based view of metaheuristics has recently been promoted to deal with several problems in the field of metaheuristic research. These problems include inconsistent metaphor usage, non-standard terminology and a proliferation of metaheuristics that are often insignificant variations on a theme. These problems make the identification of novel metaheuristics, performance-based comparisons, and selection of metaheuristics difficult. The central problem for the component-based view is the identification of components of a metaheuristic. This paper proposes the use of taxonomies to guide the identification of metaheuristic components. We developed a general and rigorous method, TAXONOG-IMC, that takes as input an appropriate taxonomy and guides the user to identify components. The method is described in detail, an example application of the method is given, and an analysis of its usefulness is provided. The analysis shows that the method is effective and provides insights that are not possible without the proper identification of the components.

Keywords: Metaheuristic, General metaheuristic, Taxonomy.

1 Introduction

The metaheuristic research field has been criticized for inconsistent metaphor usage, non-standard terminology [1, 2], and use of poor experimental setups, validation, and comparisons [1–3]. These factors have contributed to challenges in the field such as a proliferation of novel metaheuristics and 'novel' approaches being very similar to existing approaches [1, 2, 4]. Several researchers have thus proposed that a component-based view of metaheuristics that explicitly lists metaheuristic components, will assist in identifying novel components [1, 5], promote component-based performance comparison and analyses, and facilitate component-wise selection of metaheuristics for comparative studies [1, 2, 6, 7].

A component-based view is especially important for general metaheuristics, which has enjoyed increasing popularity in recent literature. General metaheuristics, also known as general metaheuristic frameworks [8], unified metaheuristic frameworks [9], and generalized metaheuristic models [10] are used for tasks such as metaheuristic generation [10], performance analysis [11, 12], metaheuristic-similarity analysis [13], and classification of metaheuristics [7]. General metaheuristics are an abstraction of a set of metaheuristics, i.e., they are generalizations of the components, structure, and information utilized by a set of metaheuristics [6, 12]. They thus also take a component-based view. General metaheuristics make use of a set of component-types, also referred to as general metaheuristics structures [12], component-categories [6], main ingredients [14], or key components [15].

However, general metaheuristics still suffer the challenges outlined above viz. inconsistent metaphor usage and non-standard terminology. They also suffer from similar problems if components are not properly identified. Thus, the identification of components takes on special importance.

This work promotes the systematic use of taxonomies to guide the identification of components. Our proposed method uses formal taxonomy theory, which appears to be absent in several recent metaheuristic studies that involve the creation or incorporation of taxonomies such as [7, 16–19]. Taxonomies, ideally, are built using a rigorous taxonomy building-method e.g. [20, 21]. Taxonomies are intrinsic prerequisites to understanding a given domain, differentiating between objects, and facilitating discussion on the state and direction of research in a domain [22]. Taxonomies may thus help solve the issues affecting metaheuristic research, such as non-standard terminology and nomenclature.

This work proposes the use of taxonomies to guide the identification of metaheuristic components. We developed a general and rigorous method, TAXONOG-IMC, that takes as input an appropriate taxonomy and guides the user to identify components. TAXONOG-IMC promotes the use of taxonomies to guide component identification for any metaheuristic subset, and provides guidance for the proper use of taxonomies to perform component identification.

This paper presents the method, provides an example of its application, and gives an analysis of its usefulness. The rest of the paper is structured as follows: section 2 provides a literature review, section 3 comprehensively describes TAXONOG-IMC, section 4 demonstrates the use of the method by applying it to two taxonomies to showcase its effectiveness, section 5 provides an analysis of the method by showing its effectiveness in analysing nature-inspired, population-based metaheuristics. Section 6 concludes the study.

2 Literature Review

The need for a component-based view is best appreciated in general metaheuristics. However, many general metaheuristics lack a rigorous method for identifying components. Many studies proposing a general metaheuristic provide guidance through examples of their usage. Several broad-scoped general metaheuristics follow this trend, such as general metaheuristics for population-based metaheuristics [9] and metaheuristics in general [10, 11, 13]. The general metaheuristics proposed by [6, 9, 10, 13] use mathematical formulations for their component-types. Since these mathematical formulations are sometimes in-part derived from text, the researcher can choose how to formulate a component based on their judgement and interpretation. However, this process can be negatively impacted by inconsistent metaphor usage and non-standard terminology. Components that are essentially the same can be regarded as different. Using examples for guidance may not account for all contingencies.

A general metaheuristic built on the assumption that differentiating the components in detail and using relatable terminology may help resolve challenges in component identification, is presented in [12]. However, most of their component-types of the general metaheuristic were a renaming of the components in [13] and may consequently face the same challenges. Some component-categories in literature were listed, but using them for the general metaheuristic may be difficult; if they consist of combinations of components, then they themselves need to be decomposed, which requires expert knowledge.

Several studies used taxonomies and/or classification-schemes to support the design of general metaheuristics. The advantage of using a taxonomy for this purpose is that it declares a convention by which the components will be identified. It provides a list of possible components that a component-type encompasses. If an issue is taken with the convention, then it can be argued at the taxonomy level. There are studies, such as [23, 24], that propose general metaheuristics whose components make use of a presented taxonomy, and there are studies that make use of existing taxonomies for a proposed general metaheuristic, such as [7, 15]. The studies that proposed both a general metaheuristic and a taxonomy are likely to work well, as the taxonomy is built for the general metaheuristic; however, taxonomies are not necessarily built with general metaheuristics in mind.

Works that use existing taxonomies lack guidance on how to use taxonomies effectively. Existing taxonomies and viewpoints were used in [15] to create a new taxonomy to guide the usage of a proposed general metaheuristic. The taxonomy presented used examples at the lowest level of its hierarchy to illustrate its usage. However, examples do not account for every contingency. The essence of the multi-level classification method proposed in [7] is meritorious; however, a misuse of the behaviour taxonomy presented in [5], led to a classification that is questionable in terms of the taxonomy used, i.e., tabu search is depicted as possessing the differential vector movement behaviour. Some studies consider tabu search as population-based but viewing tabu search as being single-solution based has a stronger consensus [25] and appears to be followed by [5], i.e., the behaviour taxonomy presented by [5] is not applicable to tabu search in its canonical sense.

The study in [14] presents a taxonomy for evolutionary algorithms based on their main components. The same study uses the taxonomy to facilitate the expression of evolutionary algorithms in terms of their main components, and the distinguishing between various evolutionary algorithm classes. This study is notable for its use of a vector representation for its components. Our work uses a similar representation.

3 Taxonomy Guided Identification of Metaheuristic Components: TAXONOG-IMC

This section proposes TAXONOG-IMC (see Fig. 1), a general, rigorous method that guides the identification of metaheuristic components using taxonomies.

We use the definition of a taxonomy provided in [20] that lends itself to a flat representation of the metaheuristics or metaheuristic component-types, which facilitates tabular analysis. A taxonomy T is formally defined in [20] as:

$$T = \left\{D_i, (i = 1, \dots, n) \middle| D_i = \left\{C_{ij}, (j = 1, \dots, k_i); k_i \geq 2\right\}\right\} \quad (1)$$

where T is an arbitrary taxonomy, D_i is an arbitrary dimension of T, $k_i \geq 2$ is the number of possible characteristics for dimension D_i, C_{ij} an arbitrary characteristic for dimension D_i. Characteristics for every dimension are mutually exclusive and collectively exhaustive, i.e., each object under consideration must have one and only one C_{ij} for every D_i.This organization, using dimensions and characteristics, is likely to be relevant in all cases since they are fundamental to understanding the properties of objects in a domain; hence the definition (1) is used.

Some important terms concerning taxonomies are explained below:

1. Dimensions: A dimension represents some attribute of an object and can be thought of as a variable that has a set of possible values.
2. Characteristics: The characteristics of a given dimension are the possible values that can be assigned to a particular dimension.
3. Taxonomy dimension: A taxonomy dimension refers to a dimension that is part of the taxonomy under consideration. The method has steps where dimensions are proposed – these are not part of the taxonomy but are under consideration to be included. We refer to these as candidate dimensions that may then become part of the taxonomy.
4. Specialized dimension: A specialized dimension is a characteristic of a taxonomy that is promoted to dimension status; specialized dimensions are candidate dimensions.
5. Generalized dimension: A generalized dimension is created by partitioning characteristics of a taxonomy dimension or partitioning the combination of characteristics from multiple taxonomy dimensions. A generalized dimension is a candidate dimension.

To illustrate each term, consider the following dimensions of some metaheuristic: initializer, search operator, and selection. Characteristics of search operator may be, e.g., genetic crossover, swarm dynamic, differential mutation. A taxonomy for evolutionary algorithms in [14] has population, structured population, information sources etc., as its dimensions. Then population would be a taxonomy dimension. Using the behaviour taxonomy presented in [5], solution creation can be thought of as a generalized dimension of the combination and stigmergy dimensions. If we use solution-creation as a taxonomy dimension, then combination would be a specialized dimension.

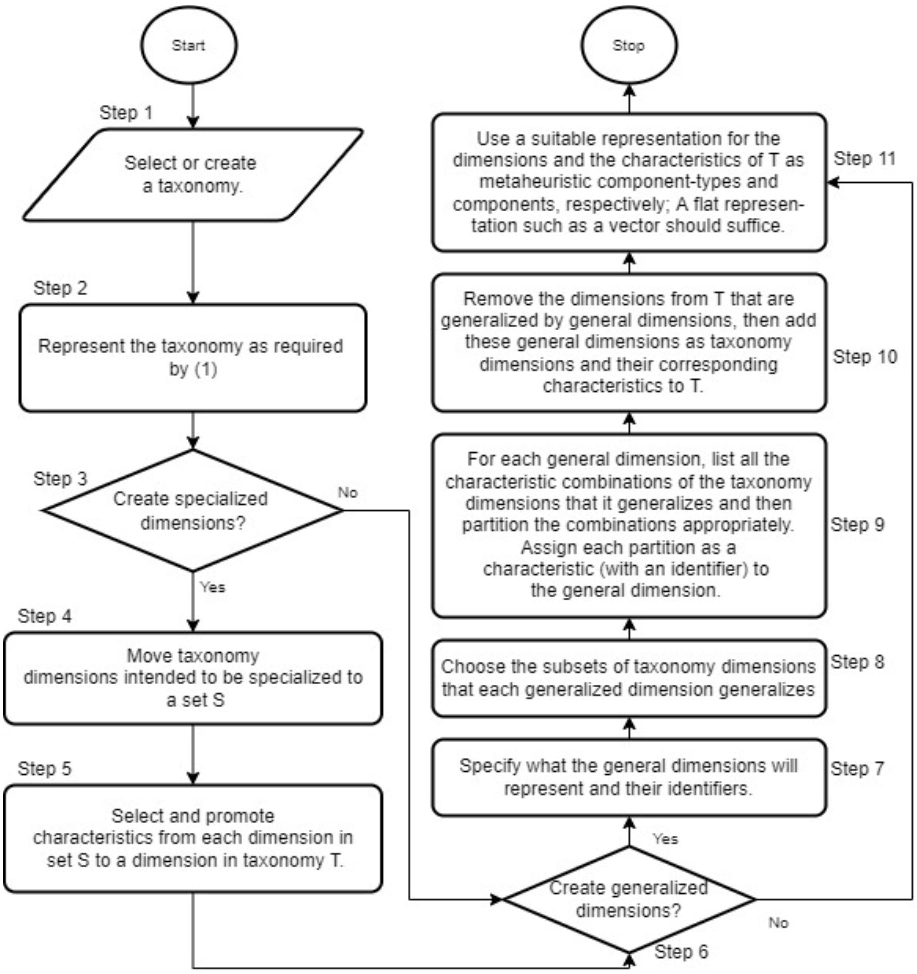

Fig. 1. Flowchart depicting the processes of TAXONOG-IMC

3.1 Comprehensive Description of Method Process

A good start for step 1 (select or create a taxonomy), is to conduct a literature search for relevant taxonomies using keywords, key-phrases, publication titles, etc. However, if no appropriate taxonomy is found, then an appropriate taxonomy building method should be used to create a taxonomy.

Expressing a taxonomy using definition (1), ensures the taxonomy is in a standard format for subsequent steps. The dimensions, and the dimensions' characteristics must be clearly stated to avoid ambiguity.

Steps 3 to 5 guides the creation of specialized dimensions. Using specialized dimensions will allow for focusing on specific components. The role of set S, introduced in step 4, is to store a collection of dimensions that are to be replaced by one of their characteristics in taxonomy T. In the metaheuristic context, a dimension may be replaced by more than one of its characteristics; this decision accommodates for hybrid-metaheuristics that have more than one characteristic for a dimension. When characteristics become dimensions, they will each need a set of possible characteristics of their own that will be derived from literature or the expertise of the researcher.

The addition of specialized dimensions to the Taxonomy may result in an overwhelmingly large number of taxonomy dimensions. Generalizing an appropriate number of taxonomy dimensions may help with this challenge.

Creating generalized dimensions is guided by steps 7 to 10. It is essential to name the general dimensions clearly and their characteristics to ensure no ambiguities nor confusion arises as to which dimension or characteristic a trait falls under. It is important to note that each subset of taxonomy dimensions, chosen in step 8, must be disjoint. Note that not every taxonomy dimension needs to be integrated into a general dimension.

As an example of when and how general dimensions can be used, consider a chosen set of metaheuristics that have a large diversity on certain taxonomy dimensions. They may be grouped by their characteristic combinations on these dimensions. A generalized dimension could then have two possible values, 1 representing a metaheuristic having a required combination of characteristics for those dimensions, and 0 representing a metaheuristic not having such a combination of characteristics for those dimensions.

4 Application of method

To demonstrate the method, we use it to generate binary component vectors to represent nature-inspired, population-based metaheuristics in terms of their inspiration and behaviour components. We use the behaviour and natural-inspiration taxonomies provided in [5]. In this study, we consider the metaphor/inspiration of a metaheuristic to be a component, but more specifically, a non-functional component. The nature-inspiration taxonomy was created to ascertain the natural-inspiration category of a metaheuristic without ambiguity. The behavioural taxonomy is based on the metaheuristic behaviour, i.e., focusing on the means by which new candidate solutions are obtained, and disregarding its natural inspiration. See section 4.3 for descriptions of all dimensions used by the behaviour and natural-inspiration taxonomies.

4.1 Behavior taxonomy

- Step 1: We use the behavior taxonomy from [5].
- Step 2: We express the taxonomy using the definition given in (1) as follows. A characteristic of 1 means that it is present and 0 means it is not.
  - b_1 - Combination (characteristics are {0, 1})
  - b_2 - Stigmergy (characteristics are {0; 1})
  - b_3 - All population Differential Vector Movement (DVM) (characteristics are {0; 1})

— b_4 - Groups-based (DVM) (characteristics are {0; 1})
— b_5 - Representative based (DVM) (characteristics are {0; 1})

- Step 3: We create specialized dimensions.
- Step 4: S = {Groups-based (DVM)}, The step at this phase dictates that we only select one characteristic to promote to dimension status, but with regards to metaheuristics, which can be hybridized and still be metaheuristics, an exception can be made such that numerous characteristics can be promoted during specialization (this depends on the characteristics, if the characteristics are single-solution and population-based then these can't both be used as component-types for a metaheuristic at the same time, since there is a possibility that both can be set to 1, which does not make intuitive sense). Therefore, we promote both Sub-population (DVM) and Neighborhood (DVM) to dimensions with their characteristics being binary {0; 1}. b_4 is set to Sub-population (DVM) and b_5 is set to Neighborhood (DVM), b_6 is set to Representative based (DVM).
- Step 5: Groups-based (DVM) is not referenced by any dimension and can thus be discarded. T = $\{b_1; b_2; b_3; b_4; b_5; b_6 \mid b_i = \{0; 1\}; (i = 1, 2, 3, 4, 5, 6)\}$
- Step 6: We do not create generalized dimension.
- Step 11: The vector representation derived from the behavior taxonomy is:

$$[b_1 \quad b_2 \quad b_3 \quad b_4 \quad b_5 \quad b_6] \tag{2}$$

4.2 Natural-inspiration taxonomy

- Step 1: We use the natural-inspiration taxonomy from [5].
- Step 2: We express the taxonomy using the definition given in (1) as follows:

— n_1 - Breeding-based evolution (characteristics are {0; 1})
— n_2 - Aquatic animals (characteristics are {0; 1})
— n_3 - Terrestrial animals (characteristics are {0; 1})
— n_4 - Flying animals (characteristics are {0; 1})
— n_5 - Microorganisms (characteristics are {0; 1})
— n_6 - Others (characteristics are {0; 1})
— n_7 - Physics-based (characteristics are {0; 1})
— n_8 - Chemistry-based (characteristics are {0; 1})
— n_9 - Social human behaviour algorithms (characteristics are {0; 1})
— n_{10} - Plants based (characteristics are {0; 1})
— n_{11} - Miscellaneous (characteristics are {0; 1})

- Step 3: We do not create specialized dimensions.
- Step 6: We create general dimensions.
- Step 7: We create two general dimensions that will be identified as Swarm-intelligence and Physics and Chemistry Based. (This is already done in the taxonomy, but we are redoing it in this process for demonstration).
- Step 8: Aquatic animals, Terrestrial animals, Flying animals, Microorganisms, Others are allocated to the Swarm-intelligence general dimension. Physics-based,

Chemistry-based are allocated to the Physics and Chemistry Based general dimension.

- Step 9: The characteristics of Swarm-intelligence are {0; 1}. 1 indicating that either Aquatic animals, Terrestrial animals, Flying animals, Microorganisms, or Others are present, 0 indicating that Aquatic animals, Terrestrial animals, Flying animals, Microorganisms, and Others are absent. The characteristics of Physics and Chemistry Based are {0; 1}. 1 indicating that either Physics-based or Chemistry-based is 1, 0 indicating that Physics-based and Chemistry-based are absent.
- Step 10: Since n_2 to n_8 are removed, n_2 will be the dimension for Swarm-intelligence, n_3 will be the dimension for Physics and Chemistry Based, n_4 will be the dimension for Social human behavior algorithms, n_5 will be the dimension for Plants based, n_6 will be the dimension for Miscellaneous; n_7 to n_{11} do not refer to any dimensions so they can be discarded. $T = \{n_1; n_2; n_3; n_4; n_5; n_6 \mid n_i = \{0; 1\}, (i = 1, 2, 3, 4, 5, 6)\}$
- Step 11: The vector representation definition derived from the selected taxonomy is:

$$[n_1 \quad n_2 \quad n_3 \quad n_4 \quad n_5 \quad n_6] \tag{3}$$

4.3 Dimension Descriptions

In this sub-section, the nodes of each hierarchal taxonomy presented in [5] are unambiguously defined as dimensions using the descriptions of each node provided in the same study; from these definitions, we can define the dimensions in the initial steps and proceed to modify them in subsequent steps by adding and/or dropping these dimensions due to using generalized or specialized dimensions.

Behaviour Dimensions

- Differential vector movement: New solution is obtained by movement relative to an existing solution
- All population Differential Vector Movement (DVM): All individuals in the population are used to generate the movement of each solution.
- Representative-based (DVM): The movements of each solution are only influenced by a small group of representative solutions, e.g., the best solutions found
- Group-based (DVM): Sub-populations or subsets of the populations are considered, without representative solutions.
- Sub-population (DVM): The movements of each solution are influenced by a subset or group of solutions in the population, and no representative solutions are determined and used in the trajectory calculation at hand.
- Neighborhood (DVM): Each solution is only influenced by solutions in its local neighborhood.
- Combination: New solutions are selected and combined via some method to create new solutions.
- Stigmergy: An indirect communication and coordination strategy is used between different solutions to create new solutions.

- Creation: Exploration of search domain by generating new solution, differential vector movement not present.

Natural-Inspiration Dimensions

- Breeding-based evolution: Inspired by the principle of natural evolution and references to producing offspring, successive generations.
- Swarm Intelligence: Inspired by the collective behavior of animal societies.
- Flying animals: Agent movements inspired by flying movements.
- Terrestrial animals: Agent movements inspired by foraging or movements of terrestrial animals.
- Aquatic animals: Agent movements inspired by animals living in aquatic ecosystems.
- Microorganisms: Agent movements inspired by food search by bacteria or how viruses spread infection.
- Others: Very low popularity inspiration sources from the collective behavior of animals.
- Physics and Chemistry Based: Imitate the behavior of physical/chemical phenomena (field of physics and chemistry).
- Social Human Behavior Algorithms: Inspired by human social concepts.
- Plants Based: Inspired by plants, where there is no communication between agents.
- Miscellaneous: Not inspired by any identified category.

5 Analysis and Discussion

We now demonstrate the use of the method. Information showing the application frequency of different nature-inspired metaheuristics to feature selection in disease diagnosis is depicted in Table 10 taken from the study in [26]. It is stated that data for the table was obtained by executing various search queries on google scholar. RA is not population-based, and thus is ignored since it is out of scope for the vector derived in the current paper. In this section, the amount of information extracted from Table 10 in [26] is extended using the derived vector. The aim is to reconfigure the table to attribute the frequencies to the component-types of the derived vector. This task is accomplished via the following steps:

1. List all metaheuristic abbreviations and ascertain their full name.
2. Represent each of the nature-inspired, population-based metaheuristics using the vector formats derived, i.e., (2) and (3), as shown in Table 1. If the metaheuristics were not present in the tables in [5], the descriptions of the dimensions of the taxonomies presented in [5] would have to be used to derive their vector representation.
3. Let B be a matrix representing the data of Table 1, i.e., $B[p][q]$ will indicate whether the component-type at column index q is present in the metaheuristic at row index p. Let D be a matrix where each intersection of row i and column j is the frequency of application of metaheuristic at row index i to the disease at col-

umn index j (D holds the data of Table 10 in [26]). Let F be the matrix that holds the component-type to disease diagnosis application frequencies (Table 2), i.e., where j is index number of the disease in the columns of Table 10 presented in [26] and q is the index number of the component-type in the vector:

$$F[j][q] = \sum_{x=0}^{N} B[x][q] \times D[x][j] \tag{4}$$

4. Matrix F contains the data of Table 2 that depicts the table of frequency of appl cation of a component-type to disease diagnosis. From this table, further analysis can be done.

Table 1. Representation of nature-inspired, population-based metaheuristics in terms of derived vector formats. The red and green colours assist the reader in viewing the 0s and 1s.

KEY: Harmony search (HS), Artificial bee colony (ABC), Glow-worm swarm optimization (GSO), Ant colony optimization (ACO), Firefly algorithm (FA), Monkey algorithm (MA), Cuckoo search (CS), Bat algorithm (BA), Dolphin echolocation (DE), Flower pollination algorithm (FPA), Grey wolf optimizer (GWO), Dragonfly algorithm (DA), Krill herd algorithm (KHA), Elephant search algorithm (ESA), Ant lion optimizer (ALO), Moth-flame optimization (MFO), Multi-verse optimizer (MVO), Runner-root algorithm (RRA), Laying chicken algorithm (LCA), Killer whale algorithm (KWA), Butterfly optimization algorithm (BOA).

PMBH	b1	b2	b3	b4	b5	b6	n1	n2	n3	n4	n5	n6
HS	1	0	0	0	0	0	0	0	1	0	0	0
ABC	0	0	0	0	0	1	0	1	0	0	0	0
GSO	0	0	0	0	0	1	0	1	0	0	0	0
ACO	0	1	0	0	0	0	0	1	0	0	0	0
FA	0	0	1	0	0	0	0	1	0	0	0	0
MA	0	0	0	0	0	1	0	1	0	0	0	0
CS	1	0	0	0	0	0	0	1	0	0	0	0
BA	0	0	0	0	0	1	0	1	0	0	0	0
DE	1	0	0	0	0	0	0	1	0	0	0	0
FPA	0	0	0	0	0	1	0	0	0	0	1	0
GWO	0	0	0	0	0	1	0	1	0	0	0	0
DA	0	0	0	0	0	1	0	1	0	0	0	0
KHA	0	0	0	0	0	1	0	1	0	0	0	0
ESA	0	0	0	0	0	1	0	1	0	0	0	0
ALO	0	0	0	0	0	1	0	1	0	0	0	0
MFO	0	0	0	0	0	1	0	1	0	0	0	0
MVO	0	0	0	0	0	1	0	0	1	0	0	0
RRA	0	0	0	0	0	1	0	0	0	0	1	0
LCA	1	0	0	0	0	0	0	1	0	0	0	0
KWA	0	0	0	0	0	1	0	1	0	0	0	0
BOA	0	0	0	0	0	1	0	1	0	0	0	0

Table 2. Frequencies of component-type usage, in literature, in various disease diagnosis applications

Disease diagnosis	b_1	b_2	b_3	b_4	b_5	b_6	n_1	n_2	n_3	n_4	n_5	n_6
Breast cancer	413	619	216	0	0	893	0	1859	236	0	46	0
Prostate cancer	35	73	9	0	0	68	0	161	21	0	3	0
Lung cancer	105	157	41	0	0	154	0	400	51	0	6	0
Oral cancer	4	3	2	0	0	6	0	12	3	0	0	0
Neck cancer	4	4	0	0	0	9	0	13	3	0	1	0
Skin cancer	19	4	15	0	0	53	0	81	8	0	2	0
HIV	40	114	24	0	0	80	0	237	18	0	3	0
Stroke	116	120	36	0	0	129	0	330	60	0	11	0
Schizophrenia	8	44	9	0	0	16	0	72	4	0	1	0
Parkinson	91	144	52	0	0	233	0	434	62	0	24	0
Heart disease	129	34	58	0	0	234	0	390	55	0	10	0
Anxiety	17	65	9	0	0	50	0	135	5	0	1	0
Insomnia	1	6	0	0	0	2	0	9	0	0	0	0
Sum	982	1387	471	0	0	1927	0	4133	526	0	108	0

It can be observed from Table 2 that b_6 (Representative-based (DVM)) is the dominant behaviour and n_2 (Swarm intelligence) is the dominant natural-inspiration. It is interesting to note that in [26], it is stated that ACO is dominant in the use of diagnosis of different human disorders. However, the behaviour associated with ACO is Stigmergy (b_2) is not the dominant behaviour; instead, representative-based differential movement (b_6) is the dominant behaviour for this application domain.

Literature such as [1] has shown that the names and metaphors of metaheuristics sometimes mask the substantial similarities between the metaheuristics and their differences are so minute that they can be considered marginal variants. ACO is popular, but the problem could lie with many metaheuristics, which have behavioural component-type b_6, being diverse in names as this trend is either diluting the core algorithm's popularity or is misguiding users to believe that different metaheuristic names entail that they have nearly orthogonal behaviours.

From Table 2, it can be ascertained that scope for future research lies in applying metaheuristics with behavioural component-types: sub-population (DVM), neighbourhood (DVM), breeding-based evolution, social-human behaviour algorithms, and miscellaneous to disease diagnosis. Even though the three latter component-types are natural-inspirations, and literature has motivated that this category of component-types has little contribution to performance. Applying them increases their presence in a population, from which data can be sampled, i.e., a diverse population is good.

The taxonomies in [5] organized the metaheuristics using their canonical versions. This study relies on the assumption that if two or more metaheuristic-algorithms are associated with the same metaheuristic, then they should possess the behaviour of that metaheuristic. The proposed method can be used to select components for metaheuristic frameworks, classification schemes, representations, and comparative analysis.

6 Conclusion

This study proposes TAXONOG-IMC, a structured method that provides guidance for metaheuristic component identification using taxonomies. An example application is provided to showcase how TAXONOG-IMC can aid in metaheuristic analysis.

Identification of metaheuristic components is an important task for the effective use of general metaheuristics, and the metaheuristic component-based view by and large. General metaheuristic publications use strategies such as providing examples, using finer-grain component-types, relying on existing taxonomies or creating new ones to assist in component identification. However, examples don't account for all contingencies that a researcher may encounter, and finer-grain components can also be affected by non-standard terminology and inconsistent metaphor usage. There are general metaheuristic publications that use taxonomies to assist in component identification; some propose their own taxonomy, and others use an existing taxonomy. The ones that propose their own taxonomy are likely to be compatible with the general metaheuristic since they are created for that purpose; however, some of the publications that use existing taxonomies made questionable decisions during the demonstration of general metaheuristic use – indicating a lack of proper use of taxonomy.

Future research lies in using taxonomies for component-identification for many other metaheuristic subsets, metaheuristics analysis, and use in general metaheuristics.

References

1. Sörensen, K.: Metaheuristics-the metaphor exposed. International Transactions in Operational Research. 22, 3–18 (2015). https://doi.org/10.1111/itor.12001.
2. Aranha, C., Camacho Villalón, C.L., Campelo, F., Dorigo, M., Ruiz, R., Sevaux, M., Sörensen, K., Stützle, T.: Metaphor-based metaheuristics, a call for action: the elephant in the room. Swarm Intelligence. (2021). https://doi.org/10.1007/s11721-021-00202-9.
3. García-Martínez, C., Gutiérrez, P.D., Molina, D., Lozano, M., Herrera, F.: Since CEC 2005 competition on real-parameter optimisation: a decade of research, progress and comparative analysis's weakness. Soft Comput. 21, 5573–5583 (2017). https://doi.org/10.1007/s00500-016-2471-9.
4. Tzanetos, A., Dounias, G.: Nature inspired optimization algorithms or simply variations of metaheuristics? Artificial Intelligence Review. 54, 1841–1862 (2021). https://doi.org/10.1007/s10462-020-09893-8.
5. Molina, D., Poyatos, J., Ser, J.D., García, S., Hussain, A., Herrera, F.: Comprehensive Taxonomies of Nature- and Bio-inspired Optimization: Inspiration Versus Algorithmic Behavior, Critical Analysis Recommendations. Cognitive Computation. 12, 897–939 (2020). https://doi.org/10.1007/s12559-020-09730-8.

6. Peres, F., Castelli, M.: Combinatorial Optimization Problems and Metaheuristics: Review, Challenges, Design, and Development. Applied Sciences. 11, 6449 (2021). https://doi.org/10.3390/app11146449.
7. Stegherr, H., Heider, M., Hähner, J.: Classifying Metaheuristics: Towards a unified multi-level classification system. Natural Computing. (2020). https://doi.org/10.1007/s11047-020-09824-0.
8. Birattari, M., Paquete, L., Stützle, T.: Classification of Metaheuristics and Design of Experiments for the Analysis of Components. (2003).
9. Liu, B., Wang, L., Liu, Y., Wang, S.: A unified framework for population-based metaheuristics. Annals of Operations Research. 186, 231–262 (2011). https://doi.org/10.1007/s10479-011-0894-3.
10. Cruz-Duarte, J.M., Ortiz-Bayliss, J.C., Amaya, I., Shi, Y., Terashima-Marín, H., Pillay, N.: Towards a Generalised Metaheuristic Model for Continuous Optimisation Problems. Mathematics. 8, 2046 (2020). https://doi.org/10.3390/math8112046.
11. De Araujo Pessoa, L.F., Wagner, C., Hellingrath, B., Buarque De Lima Neto, F.: Component Analysis Based Approach to Support the Design of Meta-Heuristics for MLCLSP Providing Guidelines. In: 2015 IEEE Symposium Series on Computational Intelligence. pp. 1029–1038. IEEE, Cape Town (2015). https://doi.org/10.1109/SSCI.2015.149.
12. Stegherr, H., Heider, M., Luley, L., Hähner, J.: Design of large-scale metaheuristic component studies. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. pp. 1217–1226. ACM, Lille France (2021). https://doi.org/10.1145/3449726.3463168.
13. de Armas, J., Lalla-Ruiz, E., Tilahun, S.L., Voß, S.: Similarity in metaheuristics: a gentle step towards a comparison methodology. Natural Computing. (2021). https://doi.org/10.1007/s11047-020-09837-9.
14. Calégari, P., Coray, G., Hertz, A., Kobler, D., Kuonen, P.: A Taxonomy of Evolutionary Algorithms in Combinatorial Optimization. Journal of Heuristics. 5, 145–158 (1999). https://doi.org/10.1023/A:1009625526657.
15. Raidl, G.R.: A Unified View on Hybrid Metaheuristics. In: Almeida, F., Blesa Aguilera, M.J., Blum, C., Moreno Vega, J.M., Pérez Pérez, M., Roli, A., and Sampels, M. (eds.) Hybrid Metaheuristics. pp. 1–12. Springer Berlin Heidelberg, Berlin, Heidelberg (2006). https://doi.org/10.1007/11890584_1.
16. Kaviarasan, R., Amuthan, A.: Survey on Analysis of Meta-Heuristic Optimization methodologies for node Network Environment. In: 2019 International Conference on Computer Communication and Informatics (ICCCI). pp. 1–4. IEEE, Coimbatore, Tamil Nadu, India (2019). https://doi.org/10.1109/ICCCI.2019.8821838.
17. Fister, I., Perc, M., Kamal, S.M., Fister, I.: A review of chaos-based firefly algorithms: Perspectives and research challenges. Applied Mathematics and Computation. 252, 155–165 (2015). https://doi.org/10.1016/j.amc.2014.12.006.
18. Diao, R., Shen, Q.: Nature inspired feature selection meta-heuristics. Artif Intell Rev. 44, 311–340 (2015). https://doi.org/10.1007/s10462-015-9428-8.
19. Donyagard Vahed, N., Ghobaei-Arani, M., Souri, A.: Multiobjective virtual machine placement mechanisms using nature-inspired metaheuristic algorithms in cloud environments: A comprehensive review. Int J Commun Syst. 32, e4068 (2019). https://doi.org/10.1002/dac.4068.

20. Nickerson, R.C., Varshney, U., Muntermann, J.: A method for taxonomy development and its application in information systems. European Journal of Information Systems. 22, 336–359 (2013). https://doi.org/10.1057/ejis.2012.26.
21. Usman, M., Britto, R., Börstler, J., Mendes, E.: Taxonomies in software engineering: A Systematic mapping study and a revised taxonomy development method. Information and Software Technology. 85, 43–59 (2017). https://doi.org/10.1016/j.infsof.2017.01.006.
22. Szopinski, D., Schoormann, T., Kundisch, D.: BECAUSE YOUR TAXONOMY IS WORTH IT: TOWARDS A FRAMEWORK FOR TAXONOMY EVALUATION. Research Papers. (2019).
23. Krasnogor, N., Smith, J.: A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issues. IEEE Transactions on Evolutionary Computation. 9, 474–488 (2005). https://doi.org/10.1109/TEVC.2005.850260.
24. Stork, J., Eiben, A.E., Bartz-Beielstein, T.: A new taxonomy of global optimization algorithms. Natural Computing. (2020). https://doi.org/10.1007/s11047-020-09820-4.
25. Glover, F., Laguna, M.: Tabu Search Background. In: Tabu Search. pp. 1–24. Springer US, Boston, MA (1997). https://doi.org/10.1007/978-1-4615-6089-0_1.
26. Sharma, M., Kaur, P.: A Comprehensive Analysis of Nature-Inspired Meta-Heuristic Techniques for Feature Selection Problem. Archives of Computational Methods in Engineering. 28, 1103–1127 (2021). https://doi.org/10.1007/s11831-020-09412-6.

CHAPTER 3: Paper Two: Towards Rigorous Foundations for Metaheuristic Research

Towards Rigorous Foundations for Metaheuristic Research

Thimershen Achary[1,2][a], Anban W. Pillay[1,2][b] and Edgar Jembere[1,2][c]
[1]*University of KwaZulu-Natal, Westville 4001, South Africa*
[2]*Centre for AI Research (CAIR), South Africa*
thimershenzn@gmail.com {pillayw4, jemberee}@ukzn.ac.za

Keywords: Metaheuristics, Component-Based View, Definition, Rigor

Abstract: Several authors have recently pointed to a crisis within the metaheuristic research field, particularly the proliferation of metaphor-inspired metaheuristics. Common problems identified include using non-standard terminology, poor experimental practices, and, most importantly, the introduction of purportedly new algorithms that are only superficially different from existing ones. In this paper, we argue that although metaphors may be good sources of inspiration and creativity, being the only reason for publication is insufficient. Instead, adopting a formal, mathematically sound representation of metaheuristics is a valuable path to follow. We believe this will lead to more insightful research.

1 INTRODUCTION

The recent past has seen an increase in research that is critical of numerous trends and practices observed in the field of metaheuristics (Aranha et al., 2021; Fister jr et al., 2016; Molina et al., 2020; Sörensen, 2015; Stegherr et al., 2020; Tzanetos & Dounias, 2021). An influential study by (Sörensen, 2015) points out several broad issues, including irresponsible metaphor usage, poor experimental practices, and misconceptions of what a metaheuristic is.

Others have lamented the poor quality and lack of rigor and insights in published works (see https://github.com/fcampelo/EC-Bestiary).
According to (Campelo & Aranha, 2021; Fister Jr et al., 2016; Sörensen, 2015), this has severe consequences for productivity, the credibility of the field, and the capability to stimulate new, valuable insights effectively.

In this paper, we review the issues raised by various researchers, consider proposed solutions, and argue that metaheuristic studies should adopt a mathematically formulated metaheuristic definition where the underlying philosophy is mindful of the issues affecting the metaheuristic field. We also agree with recent sentiments that metaphors are useful to inspire creativity but are insufficient on their own. We then propose a mindful and rigorous core understanding of metaheuristics.

1.1 Metaheuristics

The term 'meta-heuristic' was coined by Glover in (Glover, 1986), where the authors suggested that Tabu Search could be viewed as a metaheuristic "superimposed" on another heuristic. The suggestion is that metaheuristics operate on a higher level than heuristics.

Early definitions of the term metaheuristic were critically analyzed in (Voß, 2001). These definitions generally suggest that a metaheuristic is a higher-level strategy that guides subordinate heuristics, with some auxiliary constituents such as information for the guiding process and intelligent combinations of various exploration and exploitation concepts.

The meta-level is described as dealing with applying control and strategy to a given domain (Ostrowski & Schleis, 2008). In the context of heuristics being the domain, metaheuristics can then be defined as entities that apply control and strategy to heuristics, as depicted in Figure 1. Metaheuristics consists of a base plan, an integrated learning component, and strategic heuristics. The base plan and integrated learning component are utilized only

[a] https://orcid.org/ 0000-0002-6033-7065
[b] https://orcid.org/ 0000-0001-7160-6972
[c] https://orcid.org/ 0000-0003-1776-1925

in hyper-parameter tuning, while strategic heuristics are required for all metaheuristic activities. The strategic heuristics apply control and strategy to the heuristics.

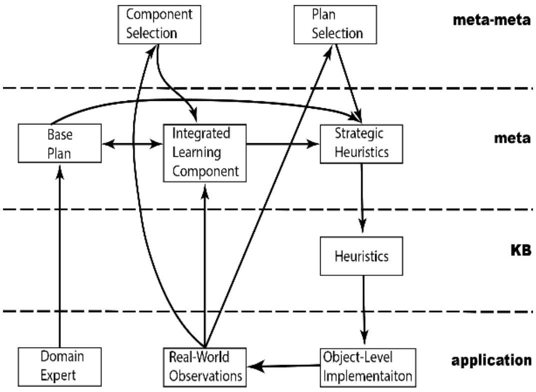

Figure 1: Metarules framework (Ostrowski & Schleis, 2008)

Metaheuristics are described in (Sörensen & Glover, 2013) as frameworks that can be used to derive heuristic optimization algorithms and notes that, in literature, the frameworks and the heuristic optimization algorithms are both referred to as metaheuristics. An elaboration of why the distinction between framework and algorithm is essential when discussing novelty can be found in (Lones, 2020); it can be inferred that, often, a novelty at the algorithm level is hardly a significant feat.

The rest of the paper is structured as follows: a review of several criticisms of the field is given in Section 2. Section 3 briefly reviews potential solutions to these problems discussed in literature. A proposal for instilling rigor in the metaheuristics research space is given in Section 4. This is discussed in Section 5, and Section 6 concludes.

2 METAHEURISTICS AND ITS DISCONTENTS

Several authors have recently pointed to problems afflicting metaheuristic research. This section summarizes these issues.

Irresponsible metaphor usage, in the metaheuristic field, is the use of sources of inspiration, e.g., nature, physics, and human behavior, to be the most, if not the only, pivotal aspect to justify the algorithm as a "new" metaheuristic to the field (Aranha et al., 2021; Sörensen, 2015). These works usually include practices that obscure details by using non-standard terminology (terminology specific to the metaphor/inspiration used). Doing so adds to the challenge of positioning the proposed contribution in literature and may give the impression that the research output is novel. Symptoms of this activity are, according to (Aranha et al., 2021; de Armas et al., 2021; Molina et al., 2020; Sörensen, 2015; Tzanetos & Dounias, 2021), a flood of metaheuristics, numerous cases of very similar/overlapping work, lack of novelty, and according to (Molina et al., 2020) instances where inspirational source and algorithm behaviour are disconnected.

Researchers have also pointed to poor experimental practices. Reports such as (Aranha et al., 2021; Sörensen, 2015; Stegherr et al., 2020; Tzanetos & Dounias, 2021) suggest unfair and biased comparisons such as comparing new proposals to older metaheuristics instead of state-of-the-art and tweaking hyperparameters in favor of a metaheuristic to lift its performance above the rest.

Comparative studies are not transparent enough, resulting in difficulties in extending past studies and existing data (Aranha et al., 2021; Sörensen, 2015). A lack of proper motivation for selecting metaheuristics to compare is common (Stegherr et al., 2020). There is also a lack of rigorous data analytics (Sörensen, 2015). Competitive studies produce very little insight and do not answer or aid in answering the how and why (Birattari et al., 2003; Hooker, 1995), yet comparative studies are still widely setup as competitive ones (Campelo & Aranha, 2021; Sörensen, 2015).

The proliferation of metaphor-inspired metaheuristics is also a cause for concern. A GitHub project called the Evolutionary Computational Bestiary lists a vast and ever-growing number of bio-inspired metaheuristics (with only a few exceptional bio-inspired metaheuristics being exempt) (Campelo & Aranha, 2021). The aforementioned project opposes the flood of metaheuristics, especially the creation of new bio-inspired metaheuristics. Articles and other projects that criticize certain metaheuristic research trends are listed, some of which are intended to parody or ridicule the fact that these trends still exist.

The above criticisms have not been universally accepted. One such counter-argument is that metaheuristics are currently being applied in various domains from numerous disciplines and have also been applied to real-world problems (Torres-Jiménez & Pavón, 2014). The view that metaheuristic research is of poor quality may very well be overly pessimistic and aims to make capital out of flaws in research techniques that are merely pragmatic. The pursuit of being theoretically optimal has little benefit to the real world.

Also, the argument goes, there is a long history of using nature to inspire the development of metaheuristic algorithms. Thus, to reject work that uses natural inspiration is to hinder creativity. The researcher pool has a diverse skill set, i.e., not all possess an advanced mathematical background, and researchers have skills/talents which may lie more in creativity than analytics. Therefore, the move to abandon natural inspiration or inspirational sources, in general, can be interpreted as a move to discriminate against researchers that are more creative than analytical.

To refute these arguments, we refer to a study by (Ven & Johnson, 2006) that explores the relationship between scholarly and practical knowledge. It analyses ways in which the discrepancies between these domains have been framed and discusses methods to address this, such as a method of engaged scholarship (proposed by the aforementioned study). From the study, it can be understood that practical and scholarly knowledge have different contexts and objectives. Practical knowledge deals with specific circumstances in certain scenarios, while scholarly knowledge deals with viewing specific circumstances as instances of a more general case to further understand and explain how what is done works. Reaping both benefits can be achieved through methods of communication between both spaces. This entails that the scholarly domain must be robust so that new knowledge can be framed efficiently amongst existing knowledge and communicated effectively to practical domains and other scholarly domains.

Recent studies have shown instances of scholarly work claiming to be novel, but the novelty does not stand up to scrutiny. Comparative studies have been questioned regarding their transparency and choice of experimental practices. The overloading of well-known concepts with non-standard terminology is creating confusion in literature. In summary, the issues highlighted by several publications are indicators that the metaheuristic research space falls extremely short of ideal conditions for a scholarly domain.

According to (Swan et al., 2015), expressing metaheuristics via mathematical formulations facilitates a rigorous definition of the term metaheuristic. Some may criticize and label this decision as systematically marginalizing creative research because mathematical definitions often use cryptic notation that may not be friendly to researchers without an advanced mathematical background and with a different skill set. However, the benefits of using mathematical formulations (more specifically, functional descriptions), as listed and discussed in (Swan et al., 2015), include promoting better communicability, reproducibility, interoperability, facilitating automated metaheuristic assembly, and promoting scientific advancement. Therefore, using mathematical formulations does not marginalize creative research; instead, it guides creativity.

The No Free Lunch theorem (Wolpert & Macready, 1997) being a valid premise in the argument for justifying the existence of a vast number of metaheuristics in the research space, is viewed as unclear in (Lones, 2020). The study also speculates that the argument may have substance as the performance of different optimizers varies when subjected to different problems. However, a discussion is presented in (Camacho-Villalón et al., 2022) that criticizes the aforementioned argument as being based on a misunderstanding of the No Free Lunch theorem for optimization and that the vast number of published metaheuristics based on metaphors are creating confusion in the research space, leading it away from proper scientific goals. Therefore, relying on the No Free Lunch theorem is not advisable to support the creation of a novel metaheuristic.

3 A REVIEW OF POTENTIAL SOLUTIONS

Several authors have not only given critical commentary on the field but have also suggested potential solutions.

The solutions to the metaheuristic research quality issues require adoption by researchers so that their impact, as argued in the respective research publications, may influence the metaheuristic research space. Increasing awareness about issues associated with metaphor-based research is therefore essential to stimulate the adoption of these solutions (Campelo & Aranha, 2021), and it is a recurring theme in many such publications, e.g., (Lones, 2020; Sörensen, 2015; Stegherr et al., 2020; Tzanetos & Dounias, 2021). Projects such as the Evolutionary Computational Bestiary are also ways to raise awareness.

A component-based view of metaheuristics, as a solution to the issues afflicting the metaheuristic research space, is highlighted in (Sörensen, 2015). This view suggests understanding metaheuristics as sets of general concepts, accompanied by the decision to distinguish metaheuristics from the optimization algorithms derived from them. Its widespread adoption may help resolve several of the problems discussed above. The component-based view of metaheuristics deals with conceptualization at the foundational layer, i.e., where definitions, taxonomies, ontologies etc., are crucial.

Applying mathematical formulations to express metaheuristics facilitates a rigorous definition of the term metaheuristic (Swan et al., 2015). Several definitions of the term metaheuristic incorporate tuples. Tuples encapsulate the specifications, main components, and sometimes structures that hold the relationships between the specifications and components.

The study by (Wang, 2010) provides worded definitions for the terms metaheuristic and metaheuristic computing. The study provides a rigorous definition of metaheuristic computing using tuples, in which the elements are concept algebra structures.

A tuple definition for population-based metaheuristics is presented as part of the unified framework for population-based metaheuristics introduced in (Liu et al., 2011).

The work done in (Cruz-Duarte et al., 2020) defines a metaheuristic as a map (expressible in terms of three components: initializer, search operator, and finalizer heuristics) from an arbitrary domain to a feasible domain of an optimization problem.

As part of the proposed design of a software framework to solve combinatorial optimization problems presented in (Peres & Castelli, 2021), a metaheuristic – actually an abstract metaheuristic – is defined as a map from a domain of specifications (encapsulated in a tuple) to a set of possible variations of the metaheuristic.

Swan et al. (Swan et al., 2015) advocate for metaheuristics to be described entirely in terms of functions (which are essentially maps), in which metaheuristics are parameterized by their environment, state, and the environments of the employed components. The environment, in this sense, refers to information required during execution, and the state refers to the solution in chosen representation form. The component heuristics are also parameterized with their environment and state.

The component-based view proposed by (Sörensen, 2015) is meritorious but has drawbacks if not used properly (Achary & Pillay, 2022). Definitions such as those presented by (Cruz-Duarte et al., 2020; Liu et al., 2011) express metaheuristics in terms of components, but as emphasized above, the ambiguity present in the definitions by (Cruz-Duarte et al., 2020) may lead to conflicting understandings. The definition by (Liu et al., 2011) uses biological terminology and thereby promotes the metaphor-based philosophy of metaheuristics. However, metaphor usage, non-standard terminology, and natural inspiration have been criticized in literature, indicating that the perspective used may nullify the long-term advantages of using the component-based view.

The framework proposed in (Peres & Castelli, 2021) resolves this ambiguity by providing mathematically formulated definitions of conceptual-level and concrete-level metaheuristics. Both are formulated as maps. The former maps from a tuple of abstract specifications to a set of concrete heuristic optimization algorithms, and the concrete heuristic optimization algorithms map from their concrete specifications to an optimal solution.

4 A PROPOSED SOLUTION

4.1 TOWARDS A RIGOROUS FOUNDATION FOR METAHEURISTIC RESEARCH

Conducting meaningful metaheuristic research for both the long and short term requires metaheuristic research to adopt strong foundations and a rigorous core.

The study by (Campelo & Aranha, 2021) summarizes some promising alternative approaches to conducting research in metaheuristics rather than relying on metaphor-based techniques. They propose understanding metaheuristics as frameworks of semi-independent modules that influence one or more intrinsic algorithmic structures. This is similar to the proposal made in (Sörensen, 2015) to see metaheuristics as frameworks and not concrete heuristic optimization algorithms. Defining metaheuristics as functions is advocated in (Swan et al., 2015), which also suggested a specific template for expressing these functions. Describing metaphor-based metaheuristics using standard terminology that effectively describes similarities and differences between metaheuristics is motivated in (Lones, 2020). Comparing metaheuristics with structure-wise similarity metrics, which facilitates determining special-case and general-case relationships between metaheuristics, is made possible by the work in (de Armas et al., 2021). Using existing taxonomies from literature rigorously is facilitated by work done in (Achary & Pillay, 2022).

Each of the above contributions has little overlap and a strict scope. Using these contributions together may be effective for establishing strong foundations for metaheuristic research and stimulating good quality, insightful research.

A rigorous foundation for metaheuristic research that makes use of the contributions, advice, and guidelines of existing literature is proposed below.

A philosophy of metaheuristics that is mindful of the issues affecting the field is provided by (Sörensen & Glover, 2013) and further explained in (Sörensen, 2015). In this view, metaheuristics are problem-independent frameworks that provide a set of guidelines to create heuristic optimization algorithms and are not the heuristic optimization algorithms themselves.

Mathematical definitions are known to be rigorous, and there are also added benefits to expressing metaheuristics as functions (Swan et al., 2015). Metaheuristics could be formulated as:

$$M: S \rightarrow A \quad (1)$$

Where *M* is an arbitrary metaheuristic and *S* is a set of tuples of specifications. The metaheuristic *M* has an influence on the tuple format, and a tuple of the set *S* must contain at least one heuristic operator. *A* is the set of heuristic optimization algorithms, each of which the rules of *M* can construct using a certain element of *S*. A proof-of-concept for the formulation in (1) can be found in Section 4.2.

The format and values of the tuples in the set *S* may be determined using the works of (Lones, 2020) and (Achary & Pillay, 2022). The novelty and influence of metaheuristics can be determined by applying the work of (de Armas et al., 2021) to metaheuristics defined in terms of (1).

This map formulation (1) aligns with the component-based view, as it guides the researcher to elucidate which components are variable in the specification tuple, thus providing scope for experiments in future research.

The restriction that an element of *S* must contain at least one heuristic operator enforces the component-based view and avoids scenarios where hyper-parameter values are the only elements of a specification tuple.

This map is very abstract and does not have many restrictions on how one may specialize it with details. Its intended use is to be a rigorous underlying conceptualization of what a metaheuristic is when proposing a concrete formulated definition for future research; this underlying conceptualization enforces alignment with the component-based view and considers the insights, advice, suggestions, and guidelines from existing literature on the problems within the metaheuristic field.

4.2 Proof of concept

The Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Bat Algorithm (BAT), and Differential Evolution (DE) metaheuristics are used to illustrate how the formulation in (1) could be used; a description of each of the aforementioned metaheuristics can be found in (Yang, 2020).

4.2.1 Genetic Algorithm

1. Substitute GA in place of *M*.
2. An element of *S* would then contain the initializer, crossover operator, mutation operator, selector, and terminating condition.
3. An element of *A* will be a resulting concrete Genetic Algorithm.

4.2.2 Particle Swarm Optimization

1. Substitute PSO in place of *M*.
2. An element of *S* would then contain the initializer, location update, velocity update, and terminating condition.
3. An element of *A* will be a resulting concrete Particle Swarm Optimization algorithm.

4.2.3 Bat algorithm

1. Substitute BAT in place of *M*.
2. An element of *S* would then contain the initializer, position update, velocity update, local search technique, and terminating condition.
3. An element of *A* will be a resulting concrete Bat Algorithm.

4.2.4 Differential Evolution

1. Substitute DE in place of *M*.
2. An element of *S* would then contain the initializer, crossover operator, mutation operator, and terminating condition.
3. An element of *A* will be a resulting concrete Differential Evolution algorithm.

5 DISCUSSION

The definition of metaheuristics adopted by a researcher will significantly influence their metaheuristic research.

A contributing factor to the proliferation of novel metaheuristics is arguably the ambiguity of whether metaheuristics are frameworks, concrete heuristic optimization algorithms, or both. The study in (Sörensen, 2015) remarks that it is unfortunate that the term "metaheuristic" is used for both general, problem-independent, algorithmic frameworks and concrete heuristic optimization algorithms derived from these frameworks and further expresses that metaheuristics are not algorithms, but they are each a set of ideas, concepts, and operators from which heuristic optimization algorithms can be derived.

The definitions presented in (Cruz-Duarte et al., 2020; Liu et al., 2011; Voß, 2001; Wang, 2010) fail to resolve this ambiguity. Difficulty in determining the novelty of new proposals may result from this ambiguity since a heuristic optimization algorithm could be related to a few or many concepts, ideas, or operators of a framework, garnished with a metaphor and non-standard terminology, then published as a novel metaheuristic.

Comparative studies of metaheuristics have received criticism in the literature (Aranha et al., 2021; Sörensen, 2015). A flaw that has been highlighted is that the implementations of metaheuristics, whose selections are poorly motivated (Stegherr et al., 2020), are compared, and the results could be misunderstood as representative of the framework.

Using metaphors and natural sources of inspiration has led to the creation of well-known, influential, and disruptive contributions such as Particle Swarm Optimization, Genetic Algorithm, Simulated Annealing, and Ant Colony Optimization, as indicated in (Camacho-Villalón et al., 2022). However, the incorporation of natural inspiration in research must outweigh the cost.

Research into the trends of metaphor and inspirational source usage (Aranha et al., 2021; Campelo & Aranha, 2021; Fister jr et al., 2016; Sörensen, 2015; Tzanetos & Dounias, 2021) has shown that metaphors and non-standard terminology introduce challenges when trying to frame metaheuristics amongst existing literature. It facilitates work similar to existing literature to be published as novel work. Non-standard terminology confuses readers and clouds the relevance and the link of the phenomenon described by the terms to the metaheuristic.

A flood of metaheuristics has been linked to metaphor and inspiration source usage. Research by (Molina et al., 2020) showed that there are many more inspiration sources than algorithmic behaviors. Hence, it can be said that inspiration source usage is a heuristic, in the general sense, for creativity, similar to the exploration of ideas. However, there is too much exploration and not enough rigor. Since metaphor/inspiration usage enhances creativity, it is insufficient on its own; this analogy is similar to those used in (Fister jr et al., 2016; Lones, 2020) with the similar computational optimization terminology

Although various publications argue that new novel metaheuristics are not needed at this point in the field's timeline, if a metaheuristic is to be published, it should be accompanied by a formulation of the metaheuristic in the format of (1). M represents the abstract pseudocode, ideas, and concepts that make up the metaheuristic. The format of elements of S will convey which components are variable, i.e., different concrete components can be substituted in their respective placeholders, which is then passed to M to create a concrete optimization algorithm of the set denoted by A in the formulation.

6 CONCLUSION

In this study, it is argued that metaheuristics studies should adopt a mathematically formulated metaheuristic definition where the underlying philosophy is mindful of the issues affecting the metaheuristic field; in other words, adopt definitions that sustain good quality research. Mathematical formulated definitions are rigorous and leave less room for vagueness that can lead to convenient interpretations. Ambiguities in adopted or proposed definitions can potentially allow choosing a definition/perspective/interpretation of the shelf that suits a requirement for publication, leading to low-quality research. The underlying philosophy of the mathematically formulated definition must be mindful of issues affecting metaheuristic research to prevent the definition from having the potential to stimulate problematic trends.

This work takes the stance that inspiration source usage is a good heuristic for creativity but is not needed right now; it has the capacity to become saturated, which is detrimental to the field. Intensifying research on existing work would be a better practice at present.

Increasing theoretical insight, better analytical techniques, and solid foundations should be a top priority of metaheuristic research.

REFERENCES

Achary, T., & Pillay, A. W. (2022). A Taxonomy Guided Method to Identify Metaheuristic Components. In D. Groen, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, & P. M. A. Sloot (Eds.), *Computational Science – ICCS 2022* (Vol. 13352, pp. 484–496). Springer International Publishing. https://doi.org/10.1007/978-3-031-08757-8_41

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*. https://doi.org/10.1007/s11721-021-00202-9

Birattari, M., Paquete, L., & Stützle, T. (2003). *Classification of Metaheuristics and Design of Experiments for the Analysis of Components*.

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*, itor.13176. https://doi.org/10.1111/itor.13176

Campelo, F., & Aranha, C. (2021). Sharks, Zombies and Volleyball: Lessons from the Evolutionary Computation Bestiary. *Proceedings of the LIFELIKE Computing Systems Workshop 2021*, *3007*.

Cruz-Duarte, J. M., Ortiz-Bayliss, J. C., Amaya, I., Shi, Y., Terashima-Marín, H., & Pillay, N. (2020). Towards a Generalised Metaheuristic Model for Continuous Optimisation Problems. *Mathematics*, *8*(11), 2046. https://doi.org/10.3390/math8112046

de Armas, J., Lalla-Ruiz, E., Tilahun, S. L., & Voß, S. (2021). Similarity in metaheuristics: A gentle step towards a comparison methodology. *Natural Computing*. https://doi.org/10.1007/s11047-020-09837-9

Fister jr, I., Mlakar, U., Brest, J., & Fister, I. (2016, October). *A new population-based nature-inspired algorithm every month: Is the current era coming to the end?*

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, *13*(5), 533–549. https://doi.org/10.1016/0305-0548(86)90048-1

Hooker, J. N. (1995). Testing heuristics: We have it all wrong. *Journal of Heuristics*, *1*(1), 33–42. https://doi.org/10.1007/BF02430364

Liu, B., Wang, L., Liu, Y., & Wang, S. (2011). A unified framework for population-based metaheuristics. *Annals of Operations Research*, *186*(1), 231–262. https://doi.org/10.1007/s10479-011-0894-3

Lones, M. A. (2020). Mitigating Metaphors: A Comprehensible Guide to Recent Nature-Inspired Algorithms. *SN Computer Science*, *1*(1), 49. https://doi.org/10.1007/s42979-019-0050-8

Molina, D., Poyatos, J., Ser, J. D., García, S., Hussain, A., & Herrera, F. (2020). Comprehensive Taxonomies of Nature- and Bio-inspired Optimization: Inspiration Versus Algorithmic Behavior, Critical Analysis Recommendations. *Cognitive Computation*, *12*(5), 897–939. https://doi.org/10.1007/s12559-020-09730-8

Ostrowski, D., & Schleis, G. (2008). New Approaches for MetaHeuristic Frameworks: A Position Paper. *AAAI Workshop - Technical Report*.

Peres, F., & Castelli, M. (2021). Combinatorial Optimization Problems and Metaheuristics: Review, Challenges, Design, and Development. *Applied Sciences*, *11*(14), 6449. https://doi.org/10.3390/app11146449

Sörensen, K. (2015). Metaheuristics-the metaphor exposed. *International Transactions in Operational Research*, *22*(1), 3–18. https://doi.org/10.1111/itor.12001

Sörensen, K., & Glover, F. W. (2013). Metaheuristics. In S. I. Gass & M. C. Fu (Eds.), *Encyclopedia of Operations Research and Management Science* (pp. 960–970). Springer US. https://doi.org/10.1007/978-1-4419-1153-7_1167

Stegherr, H., Heider, M., & Hähner, J. (2020). Classifying Metaheuristics: Towards a unified multi-level classification system. *Natural Computing*. https://doi.org/10.1007/s11047-020-09824-0

Swan, J., Adriaensen, S., Bishr, M., Burke, E. K., Clark, J. A., Durillo, J. J., Hammond, K., Hart, E., Johnson, C. G., Kocsis, Z. A., Kovitz, B., Krawiec, K., Martin, S., Merelo, J. J., Minku, L. L., Pappa, G. L., Pesch, E., Garc, P., Schaerf, A., … Wagner, S. (2015). *A Research Agenda for Metaheuristic Standardization*.

Torres-Jiménez, J., & Pavón, J. (2014). Applications of metaheuristics in real-life problems. *Progress in Artificial Intelligence*, *2*(4), 175–176. https://doi.org/10.1007/s13748-014-0051-8

Tzanetos, A., & Dounias, G. (2021). Nature inspired optimization algorithms or simply variations of metaheuristics? *Artificial Intelligence Review*, *54*(3), 1841–1862. https://doi.org/10.1007/s10462-020-09893-8

Ven, A., & Johnson, P. (2006). Knowledge for Theory and Practice. *Academy of Management Review*, *31*, 802–821. https://doi.org/10.2307/20159252

Voß, S. (2001). Meta-heuristics: The State of the Art. In G. Goos, J. Hartmanis, J. van Leeuwen, & A. Nareyek (Eds.), *Local Search for Planning and Scheduling* (Vol. 2148, pp. 1–23). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45612-0-1

Wang, Y. (2010). A Sociopsychological Perspective on Collective Intelligence in Metaheuristic Computing: *International Journal of Applied Metaheuristic Computing*, *1*(1), 110–128. https://doi.org/10.4018/jamc.2010102606

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82. https://doi.org/10.1109/4235.585893

Yang, X.-S. (2020). *Nature-inspired optimization algorithms* (2nd ed.). Elsevier Inc.

Chapter 4: Paper Three: A New Metaheuristic-Algorithm Similarity Measure Using Signal Flow Diagrams.

A New Metaheuristic-Algorithm Similarity Measure Using Signal Flow Diagrams

FirstName Surname[†]
Department Name
Institution/University Name
City State Country
email@email.com

FirstName Surname
Department Name
Institution/University Name
City State Country
email@email.com

FirstName Surname
Department Name
Institution/University Name
City State Country
email@email.com

ABSTRACT

Methods and measures for assessing the similarity between metaheuristic-algorithms are important tools for practitioners and researchers. This will allow informed comparisons when assessing new metaheuristics and when choosing metaheuristics for problems. The component-based view of metaheuristics has been promoted to improve comparison analyses between metaheuristics. This view promotes the identification of structural components of metaheuristics and metaheuristic-algorithms for any analysis pertaining to metaheuristics and metaheuristic-algorithms. In this paper, we present a new similarity measure for metaheuristic-algorithms based on a modified version of a signal flow representation of metaheuristic-algorithms that is aligned with the component-based view. It involves taking two metaheuristic-algorithms and decomposing them into their heuristic components whilst taking note of the order of execution of the heuristics. Thereafter, the features of the heuristics are extracted and subjected to a feature-based similarity calculation that also considers the position of the heuristics in the metaheuristic-algorithms' composition to obtain an overall similarity score. We demonstrate the use and effectiveness of the method by applying it to several algorithms from the Niapy collection.

CCS CONCEPTS

• Computing methodologies ~ Artificial intelligence ~ Search methodologies • Computing methodologies ~ Machine learning ~ Machine learning approaches ~ Bio-inspired approaches

KEYWORDS

Metaheuristic, Similarity, Component-based view, Features

*Article Title Footnote needs to be captured as Title Note
†Author Footnote to be captured as Author Note

GECCO'21, July, 2021, Lille, France

1 Introduction

There has been a growing number of publications reporting on the proliferation of novel nature-inspired metaheuristics where overlapping work is prevalent [3, 25]. The flood of so-called novel metaheuristics [3] has been attributed to the use of non-standard terminology, poor experimental practices, and stretched narratives of phenomena observed in nature that obfuscate important details needed to conceptualize a metaheuristic and position it in the existing literature. This makes the development of similarity and comparison methods particularly difficult.

The component-based view for metaheuristic research is promoted in [23] as a possible mitigation of these problems. This view first requires differentiating between metaheuristics and metaheuristic-algorithms, thereafter, drawing attention to the heuristic and other structural components of metaheuristics and metaheuristic-algorithms, and implicitly away from the inspiration source. When developing a similarity or comparison method for metaheuristic-algorithms, employing a representation is a crucial first step. The recently proposed signal flow diagram [10] for representing metaheuristics and metaheuristic-algorithms has been shown to account for intricate metaheuristic structures. It has already been applied in metaheuristic-algorithm analysis [8] and metaheuristic-algorithm generation [9]. The signal flow diagram focuses on the heuristic components of a metaheuristic-algorithm, or, in the case of a metaheuristic, it focuses on abstractions of the heuristic components.

This paper adopts the definition of metaheuristics provided in [24] that views metaheuristics as frameworks from which metaheuristic-algorithms are derived. A mathematical formulation is given in [1]:

$$M: S \rightarrow A \tag{1}$$

where M is an arbitrary metaheuristic, S is a set of tuples of specifications, and A is the set of metaheuristic-algorithms from which any element can be constructed by following the rules of M with a particular element from the set S. The rules of metaheuristic M determines the tuple format, and a tuple of the set S must contain at least one heuristic operator. In this view, metaheuristics output metaheuristic-algorithms, while metaheuristic-algorithms output solutions.

Representation via signal flow diagram was chosen due to its alignment with the component-based view, effective incorporation of structural information of a metaheuristic-algorithm, and the usability of the signal flow diagram in various types of analyses.

This paper thus presents a method of measuring the similarity between metaheuristic-algorithms and used a modified signal flow representation. We demonstrate the use and effectiveness of the method by applying it to several algorithms from the NiaPy collection [26].

The rest of this paper is structured as follows: a background to the work is given in section 2, the proposed similarity measure is discussed and demonstrated in section 3, applying the similarity measure to actual metaheuristic-algorithm implementations is showcased in section 4, a discussion is given in section 5, section 6 concludes the study.

2 Background

This section provides some preliminaries and positions the study among the existing literature in the metaheuristic field. The background focuses on metaheuristic-algorithm representation, measuring the similarity of metaheuristic-algorithms, the adopted representation technique, and a similarity measure appropriate for feature-based similarity.

2.1 Representations in metaheuristics

There are several metaheuristic and metaheuristic-algorithm representation techniques in the literature that differ in focus, granularity, and organization, which attribute the representation techniques with different levels of effectiveness pertaining to the task at hand, such as metaheuristic/metaheuristic-algorithm illustration, analysis, and generation. Representations of metaheuristics and metaheuristic-algorithms include flowcharts, employing a pool template, block diagrams, signal flow diagrams, assorted types of digraphs, such as in [29], visualizations of the metaheuristics in terms of their natural-inspiration, vector representations, and grammar-based representations.

Flowcharts like those in [11, 21, 27] are just a visualization of the algorithms and do not stimulate much insight. The visualizations of metaheuristics in terms of their natural-inspiration, such as describing the heart and circulation system in [12] or the ant-lion's trap and hunting behavior in [16], are illustrative of the inspiration source for a metaheuristic. However, given the inherent issues with natural-inspiration usage and the often tenuous link between models and phenomena in the metaheuristic field [3], the visualization of metaheuristics in terms of their natural-inspiration are unlikely to provide insights for analysis. The pool template format used in [4] uses a comprehensive set of metaheuristic features, but encapsulating the perturbator heuristics (heuristics that modify a solution [10]) and their arrangements within the updating mechanism field leads to sequences and branches being stored as non-atomic data in a row-column intersection. This introduces challenges in row-to-row comparisons; thus, structural information from the metaheuristic-algorithm is not optimally accommodated.

The block diagram presented as part of the work done in [10] focuses on rigorously defined categories of heuristic components and their precedence but the study also reports that the block diagram representation cannot accommodate intricate component arrangements, such as those arising from certain population topologies. The signal flow diagram (based on signal flow theory), presented by the same study, i.e. the study published in [10], does allow for operators to be sequential and/or branched in their arrangement. The signal flow diagram and block diagram lack emphasis on the features of the heuristics. Instead, an identifier of the concrete heuristics is given, with little to no focus on the similarity and relationships between heuristics.

Grammar-based representations, such as those in [18, 19], have mainly been used for parsing a formal text to generate or evolve metaheuristic-algorithms. However, they may not be suitable for component-wise analysis without extra work to extract details. Vector representations intended for analysis and classification have been showcased in [5, 6, 15]. Vector representations are a general representation form since a vector, with the necessary information and format, can be used to generate block diagrams, grammar-based representations, and pool template representations.

2.2 Metaheuristic-algorithm similarity

According to [13], in general, a measure of similarity quantifies the degree of association or likeness between two objects; this quantification usually results in a value positioned between 0 (indicating complete dissimilarity) and 1 (indicating complete similarity) inclusive. Distance-based, feature-based, and probabilistic similarity measures are the major approaches to measuring similarities. The feature-based similarity measure category is relevant to the current study.

A component-based similarity measure for metaheuristic-algorithms has recently been presented [4]. It utilizes a pool template format as its underlying representation. The pool template representation involves extracting features of perturbator heuristics; this shows commonalities between perturbator heuristics under different names. An issue with the aforementioned similarity measure is that it does not take into account that many perturbator heuristics may have the same feature and that a repetition of a perturbator heuristic in an arrangement should also be considered, thus the aforementioned issue leads to neglecting where and how many times a perturbator heuristic feature is employed during the similarity calculation; this issue can be attributed to the chosen representation technique, i.e., the pool template format. The pool template's design in [4] results in a flat vector representation of a metaheuristic-algorithm, but certain arrangements of components of a metaheuristic-algorithm lead to non-atomic data being stored in elements of the vector. This owes to the aforementioned similarity measure neglecting the arrangement and usage frequency of perturbators heuristic features during the relevant calculations.

Studies in measuring metaheuristic-algorithm similarity using empirical methods have been reported in [20, 28], but these techniques of measuring similarity are outside the scope of this study.

2.3 Modified Signal Flow Diagram Representation

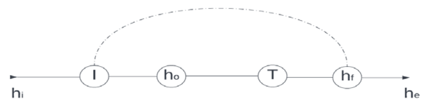

Figure 1: Signal flow diagram with the addition of demarcating nodes *I* and *T*

Figure 2: Cascading arrangement of a search operator.

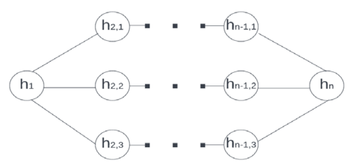

Figure 3: Parallel composition of a search operator that diverges from and merges to other search operators.

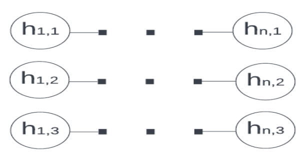

Figure 4: Parallel composition of a search operator.

The representation technique used for our similarity measure is the signal flow diagram [10]. The signal flow diagram shown in figure 1 is capable of representing a metaheuristic or metaheuristic-algorithm; it represents a metaheuristic when the heuristic components are abstract, and it represents a metaheuristic-algorithm when the abstract heuristic components are concretized. The heuristic components h_i, h_o, h_f, and h_e, which are the initializer, search operator, finalizer, and identity heuristics, respectively, are rigorously defined in [10].

A few changes were made to the signal flow diagram. These are the addition of demarcating nodes *I* and *T* and looping from finalizer to node *I* instead of h_o. This was to accommodate the contingency where the parallel composition proceeds immediately after the initializer, as will be the case if h_o is expanded into the arrangement shown in figure 4 as opposed to the arrangement in figure 3, where branches diverge from a search operator. Constructing loops from h_f to several search operators would be cumbersome. In this work, when the search operator node is expanded into arrangements such as those depicted in figures 2 – 4 or hybridizations of such arrangements, the nodes refer to feature vectors. Although in [10], the search operator comprises a perturbator and selector (post-selector) heuristic, we take into account three categories of heuristics viz. pre-selector, perturbator, and post-selector as done in [4, 29], but a feature representation of each node is used to measure similarity since features of heuristics allude to commonalities between heuristics under different names.

The modified signal flow diagram can effectively account for where and how many times a heuristic feature is employed. This is an advantage over the pool template format used in [4], for which the similarity measure only accounts for the presence or absence of a heuristic feature.

Some important terminology for understanding the use of the modified signal flow diagram in this study are:

- Merging point: This is a common node where the outgoing arcs of the last nodes of the branches of a parallel composition join. The common node could be a search operator node or node *T*.
- IT diagram: This diagram is the isolation of the portion of the signal flow diagram between nodes *I* and *T* (inclusive of nodes *I* and *T*); the composite search operator is expanded to showcase all the search operators that compose it and their arrangement.
- Pre-selector: A heuristic that determines whether or not a solution will be operated on by a given perturbator heuristic such as selecting a solution for perturbation if in the top ten fittest solutions, described in [4, 29].
- Perturbator heuristic: Rigorously defined in [10], basically a heuristic that modifies a solution such as genetic crossover.
- Post-selector: After a perturbation has occurred, we have what is called a candidate solution. The post-selector is a heuristic that selects whether or not the candidate solution will be added to the population or used in subsequent perturbations; in many cases, if the candidate solution is selected to be added to the population, it replaces the solution it was generated from, rigorously defined in [10] but is just called a selector.

2.4 Jaccard Index

In this work, binary features are extracted from the heuristics employed by metaheuristic-algorithms to determine the similarity between the heuristics used. The choice of using binary features does not have to be a standard, as features that have many possible feature-values could be used. However, feature-engineering, and determining how best to organize features and which features to use, are beyond the scope of this study. The features used in this study are roughly determined from previous work that uses metaheuristic features. Binary feature vectors are used for demonstration in this study; hence the Jaccard index for bags (sourced from [7, 22, 30]) is employed due to its applicability.

A binary feature vector can be interpreted as a set where the features that have the value 1 in the vector are present in the set, and the features that have the value 0 are absent in the set. Therefore, the Jaccard index for sets can be applied to determine the similarity between two binary feature vectors.

A sum of two or more binary feature vectors results in a bag/multiset, i.e., a set that removes the uniqueness constraint on its elements. Therefore, the Jaccard index for bags can be applied to calculate the similarity of two vectors where both vectors are possible sums of two or more binary feature vectors.

Given two bags, A, and B, the Jaccard index of the bags may be calculated as below:

$$J(A,B) = \frac{|A \cap B|}{|A \uplus B|} \tag{2}$$

The intersection and union operators used in the formula are for bags, i.e., bag-intersection and bag-union respectively. The bag-intersection between two bags returns all the elements common to both bags, with the number of occurrences of a common element being equal to that of the bag with the fewest occurrences of the element. The bag-union of two bags returns all elements of both bags, with the number of occurrences of an element being equal to the sum of the occurrences of that element from both bags. Note that the maximum value of the Jaccard index for bags is ½, as opposed to the Jaccard index for sets which is 1.

3 Method of Measuring Similarity

3.1 Step-By-Step Method

- **Step 1**: Select a metaheuristic-algorithm *X* and a metaheuristic-algorithm *Y* to measure their similarity.
- **Step 2**: Represent metaheuristic-algorithms *X* and *Y* using the signal flow diagram isolated between nodes *I* and *T* (inclusive); call these diagrams the IT diagrams for metaheuristic-algorithms *X* and *Y*.
- **Step 3**: For the IT diagram for metaheuristic-algorithm *X*, identify parallel composition arrangements for nodes between *I* and *T*. For each parallel composition arrangement, first, identify the branch with the most nodes before the merging point. Then pad the branches with fewer nodes starting from the tail end (before the merging point) with nodes that refer to default feature vectors until all branches in a parallel composition arrangement have the same number of nodes before the merging point.
- **Step 4**: Repeat step 3 but for the IT diagram of metaheuristic-algorithm *Y*.
- **Step 5**: Perform level-wise aggregation of feature vectors referred to by nodes of the IT diagram for metaheuristic-algorithm *X*. The result is a vector called the summary vector for metaheuristic-algorithm *X*, in which the elements are the aggregated vectors for each level in respective order. Note: If the IT diagram for metaheuristic-algorithm *X* is solely a cascading arrangement, then the summary vector is just a vector of the feature vectors referred to by the nodes of the IT diagram in respective order.
- **Step 6**: Repeat step 5 for the IT diagram of metaheuristic-algorithm *Y*.
- **Step 7**: After obtaining a summary vector for metaheuristic-algorithm *X* and a summary vector for metaheuristic-algorithm *Y*, pad the summary vector that is the shortest of the two at the tail end with default vectors so that both summary vectors have the same length.
- **Step 8**: Apply an appropriate similarity measure between the corresponding aggregated vectors at each index of the summary vectors for metaheuristic-algorithms *X* and *Y*. The result will be a single vector with the similarity scores for each index of the summary vectors; call this single vector the similarity vector.
- **Step 9**: For each index of the similarity vector, there will be a maximum possible score. Add up the maximum scores for each index. The result is the total max score.
- **Step 10**: Sum the elements of the similarity vector, then divide the sum by the total max score. The result is an overall similarity between metaheuristic-algorithms *X* and *Y*. Note: The overall similarity's maximum value is 1, and the minimum value is 0.

3.2 Similarity Calculation Demonstration

Let *F1*, *F2*, *F3*, *F4*, *F5* be arbitrary perturbator heuristic features, then consider the hypothetical metaheuristic-algorithms *X* and *Y*, given by the IT diagrams below:

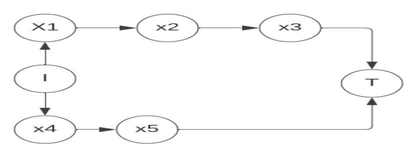

Figure 5: IT diagram for metaheuristic-algorithm *X*.

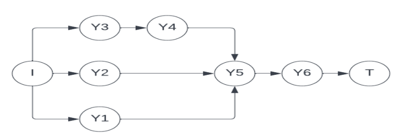

Figure 6: IT diagram for metaheuristic-algorithm *Y*.

Table 1: Feature vectors (the column vectors) for each node of the IT diagram in figure 5.

Features	X1	X2	X3	X4	X5
F1	0	1	1	1	0
F2	1	1	0	1	0
F3	0	0	0	1	0
F4	1	1	0	0	1
F5	0	0	1	0	0

Table 2: Feature vectors (the column vectors) for each node of the IT diagram in figure 6.

Features	Y1	Y2	Y3	Y4	Y5	Y6
F1	1	0	0	1	1	0
F2	1	1	1	0	1	1
F3	1	1	0	1	1	1
F4	1	1	1	1	1	1
F5	1	0	0	1	0	1

Table 3: Summary vector (Each row is an element of the summary vector, i.e., vector of vectors) for the IT diagram in figure 5.

Level	F1	F2	F3	F4	F5	Operation
1	1	2	1	1	0	$(X1 + X4)^T$
2	1	1	0	2	0	$(X2 + X5)^T$
3	1	0	0	0	1	$(X3 + \mathbf{0})^T$
4	0	0	0	0	0	$(\mathbf{0})^T$

Table 4: Summary vector (Each row is an element of the summary vector, i.e., vector of vectors) for the IT diagram in figure 6.

Level	F1	F2	F3	F4	F5	Operation
1	1	3	2	3	1	$(Y1 + Y2 + Y3)^T$
2	1	0	1	1	1	$(Y4 + \mathbf{0} + \mathbf{0})^T$
3	1	1	1	1	0	$(Y5)^T$
4	0	1	1	1	1	$(Y6)^T$

Table 5: Result of bag intersection operation performed between summary vectors given in tables 3 & 4.

Level	F1	F2	F3	F4	F5
1	1	2	1	1	0
2	1	0	0	1	0
3	1	0	0	0	0
4	0	0	0	0	0

Table 6: Result of bag union operation performed between summary vectors given in tables 3 & 4.

Level	F1	F2	F3	F4	F5
1	2	5	3	4	1
2	2	1	1	3	1
3	2	1	1	1	1
4	0	1	1	1	1

Table 7: Calculation of the Jaccard index for each level of the IT diagrams given in figures 5 & 6.

Level	The cardinality of Bag intersection (column-wise summation of table 5)	The cardinality of Bag union (column-wise summation of table 6)	Jaccard Index	MAX
1	5	15	5/15	½
2	2	8	2/8	½
3	1	6	1/6	½
4	0	4	0	½
Overall Similarity = (3/4)/2 = 0.375				

The nodes of the IT diagrams for metaheuristic-algorithms *X* and *Y* refer to feature vectors given in table 1 and table 2, respectively (Each row-column intersection is a feature-value, and columns are the feature vectors). The level of a node is the path length from node *I* to that node (determined after padding is completed in step 3 and step 4). Steps 1 and 2 are completed by considering hypothetical metaheuristic-algorithms *X* and *Y* and representing them via IT diagrams and their corresponding feature vectors (figure 5, figure 6, table 1, and table 2 – respectively). Step 3 is completed for metaheuristic-algorithm *X*; a node referring to the zero vector is appended to the branch with nodes *X4* and *X5*; this is accommodated for in step 5. Step 4 is completed for metaheuristic-algorithm *Y*; zero vectors are appended to the shorter branches, i.e., branches with nodes *Y1* and *Y2*; this is accommodated for step 6. In step 5, the level-wise aggregation of feature vectors within metaheuristic-algorithm *X* is given in table 3 and proceeds as follows:

- Level 1: Nodes *X1* and *X4* are on the same level; their features are aggregated.
- Level 2: Nodes *X2* and *X5* are on the same level; their features are aggregated.
- Level 3: Nodes *X3* and the zero-vector padded on in step 3 are on the same level; their feature vectors are aggregated (see level 3 in table 3).
- Level 4: This level is created due to step 7 and is touched on when that step is reached in this demonstration.

In step 6, the level-wise aggregation of feature vectors within metaheuristic-algorithm *Y* is given in table 4 and proceeds as follows:

- Level 1: Nodes *Y1*, *Y2*, and *Y3* are on the same level; their feature vectors are aggregated.
- Level 2: Nodes *Y4* and the zero vectors padded on in step 4 are on the same level; their feature vectors are aggregated (see intersection of level 2 and column labelled Operation in table 4).
- Level 3: Node *Y5* is the only node on level 3, so it is added as is to the summary vector.

- Level 4: Node *Y6* is the only node on level 4, so it is added as is to the summary vector.

For step 7, metaheuristic-algorithm *X* has the fewest number of levels (3 levels), so its summary vector is padded with an additional level to bring it to 4 levels; this is shown with a zero vector at level 4 in table 3.

In step 8, the Jaccard index for bags is applied to corresponding aggregated vectors of the summary vectors for metaheuristic-algorithms *X* and *Y*; tables 5 and 6 show the results of the bag-intersection and bag-union operations of the summary vectors, respectively. The Jaccard index for each level is shown in table 7. The column labeled Jaccard index in table 7 is the similarity vector; each element has a maximum value of 1/2; therefore, the total maximum similarity is equal to 2 (step 9 completed). Step 10 proceeds by totaling the elements of the similarity vector and dividing the sum by the total maximum similarity; the result is the overall similarity value of metaheuristic-algorithm *X* and metaheuristic-algorithm *Y,* shown in the last row of table 7, which lies between 0 and 1.

4 Applying Similarity Measure to Actual Metaheuristic Algorithms

Five metaheuristic-algorithms of varying structure and heuristic features were selected from the NiaPy library [26] to showcase the usage of the presented comparison methodology.

Representing metaheuristic-algorithms directly from source code is a challenge. There are many instances where one can fall into subjectivity traps. This is due to source-code creation being dependent on a programmer's coding style.

An ad-hoc method is used to derive the representation, whereby any function that calls to a particular heuristic or another function that eventually calls the perturbator heuristic is identified in the source code. The pre-selector and post-selector are then worked out by observing the code that will be executed prior to and post perturbator execution, respectively.

The features of the perturbator and its associated pre-selector, post-selector, and miscellaneous information are recorded for the respective metaheuristic-algorithm to be used for the comparison. Deciding what features to use for extraction and when they apply is a challenge – some features overlap and may have a relationship, e.g., hierarchical. Taxonomies would be extremely useful for this task, but relevant taxonomies are lacking. A relevant taxonomy is used, but it is not comprehensive/robust enough to account for the encountered contingencies. Thus, features were created in an ad-hoc manner, leading to some challenges as to which feature applies – best effort was used to decide. The IT diagrams for implementations of HHO (HarrisHawksOptimization), BA (BatAlgorithm), CRO (CoralReefsOptimization), FSS (FishSchoolSearch), and PSO (ParticleSwarmAlgorithm), from the NiaPy library [26], are shown in figures 7, 8, 9, 10, and 11, respectively. Tables 9 and 10 touch briefly on some elements of the calculation. The blank cells in table 9 and 10 represent cells where the value is 0, and the colored cells in tables 9 and 10 are only used to improve readability – they don't convey any meaning. Table 11 shows the overall similarity scores between the metaheuristic implementations focused on in this section.

Table 8 shows the features that will be extract for each node and their definitions. The choice of features is based on features used in literature. The features found in the literature are not comprehensive enough to accommodate all encountered scenarios, so some features had to be modified in an ad-hoc manner to some extent. Features used were sourced from [4, 17]; some features were created intuitively, if features found in the literature were not detailed enough. Re-occurring terms in the feature definitions are:

- Differential vector movement (DVM) [17]: Generating a candidate solution via a linear combination of vectors.
- Guided insertion (GI): Inserting values into specific elements of a solution vector.
- Parameter setting: A value that influences how a metaheuristic-algorithm is run in some aspect, usually set prior to a metaheuristic-algorithms execution but may be updated during the runtime.

The overall similarity values, shown in table 11, are very low. However, consideration should be given to the fact that the features used for demonstration in this study are treated as having an orthogonal relationship with every other feature. However, similarities between features do exist; for example, all the features that have the Differential Vector Movement (DVM) trait have it has one of their similarities. Creating or implementing a similarity measure between bags that considers non-orthogonal features is a direction for future research.

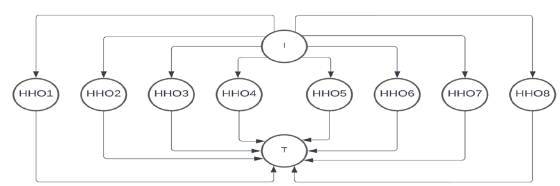

Figure 7: IT diagram for HHO metaheuristic-algorithm

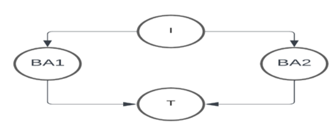

Figure 8: IT diagram for BA metaheuristic-algorithm

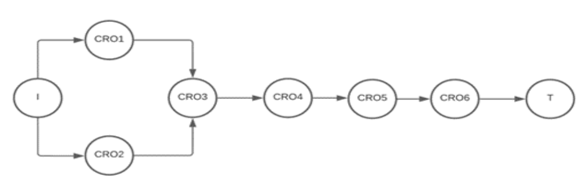

Figure 9: IT diagram for CRO metaheuristic-algorithm

Table 8: Definitions of features used for calculating similarity. All features are binary features – 0 indicating that a feature is absent, 1 indicating that a feature is present.

Features	Definition
Reference-current	The perturbator uses the current solution in the calculation of a candidate solution.
Reference-representative	The perturbator uses an individual representative of the population, e.g., the best solution or a solution in the top 3 solutions, for calculating a candidate solution (the current solution is not utilized).
Solution(GI)	Inserting values into particular elements of a solution to generate a candidate solution. The values inserted are determined using information from a solution of the population.
Self(GI)	Inserting values into particular elements of a solution to generate a candidate solution. The values inserted are determined using information from the solution (specified) being operated on.
Random(GI)	Inserting values into particular elements of a solution to generate a candidate solution. The values inserted are randomly generated.
All population(DVM)	Uses all individuals of the population individually to generate a candidate solution through differential vector movement.
Derived representative solution (DVM)	Uses a significant, calculated solution (using individuals of the population) to generate a candidate solution through differential vector movement.
Representative(DVM)	Using an individual that represents some statistic of the population to generate a candidate solution through differential vector movement.
Random movement(DVM)	Using a randomly generated vector to generate a candidate solution through differential vector movement.
Arbitrary Invididual(DVM)	Using a random individual of the population to generate a candidate solution through differential vector movement.
Select all	Select all individuals of the population to be operated on to generate candidate solutions.
Fitness influenced selection	Selection of individuals from the population to be operated on to generate candidate solutions is determined by their fitness.
Random selection	Selection of individuals from the population to be operated on to generate candidate solutions is influenced by some random distribution.
Parameter-based selection	Selection of individuals from the population to be operated on to generate candidate solutions is influenced by some parameter setting.
Allow all	Allow all candidate solutions to update their respective solutions for the next generation or to be used in subsequent perturbations.
Parameter influenced	The allowance of candidate solutions to update their respective solutions for the next generation or to be used in subsequent perturbations is influenced by some parameter setting.
Allow better than current	Allowance of candidate solutions to update their respective solutions for the next generation or to be used in subsequent perturbations is partially or fully determined by the candidate solution being better than its respective solution.
Allow best of all candidates	For a pre-selected solution, many candidate solutions are generated, and only the best candidate solution can update the respective solution for the next generation or be used in subsequent perturbations.
Discard Individuals	Individuals not selected for the update are discarded, i.e., they do not carry over to subsequent iterations nor are used in subsequent perturbations.
Identity heuristic	Update does not change the solution.
Iteration activated	Update is performed at a specified iteration.
Personal best in update	The update of an individual solution uses its best-known solution found during runtime.

Table 9: Feature vectors (the column vectors) for each node of the IT diagrams in figures 7, 8, 9, 10 & 11

Features	HHO1	HHO2	HHO3	HHO4	HHO5	HHO6	HHO7	HHO8	CRO1	CRO2	CRO3	CRO4	CRO5	CRO6	FSS1	FSS2	FSS3	FSS4	PSO1	BA1	BA2
Reference-current	1		1	1	1	1			1	1	1	1	1	1	1	1	1	1	1	1	
Reference-representative		1					1	1													1
Solution(GI)									1												
Self(GI)											1		1								
Random(GI)										1		1									
All population(DVM)																1	1	1			
Derived statistic (DVM)		1					1	1													
Representative(DVM)			1	1	1	1													1	1	
Random movement(DVM)		1				1		1			1		1		1						1
Arbitrary Invididual(DVM)	1																				
Select all															1	1			1		
Fitness influenced selection												1		1			1	1			
Random selection	1	1	1	1	1	1	1	1	1	1										1	1
Parameter-based selection	1	1	1	1	1	1	1	1	1	1	1	1	1							1	1
Allow all	1	1	1	1	1	1	1	1	1	1		1				1	1	1	1		
Parameter influenced											1		1	1							
Allow better than current					1	1	1	1							1					1	1
Allow best of all candidates						1		1													
Discard Individuals											1		1	1							
Identity heuristic									1					1							
Iteration activated														1							
Personal best in update																			1		

Table 10: Level-wise aggregation of feature vectors of level for each of the metaheuristic-algorithms' IT diagrams given in figures 7, 8, 9, 10 & 11 (Each column is the result of aggregating the feature vectors at a particular level of specific metaheuristics)

Features	HHO Level 1	CRO Level 1	CRO Level 2	CRO Level 3	CRO Level 4	CRO Level 5	FSS Level 1	FSS Level 2	FSS Level 3	PSO Level 1	BA Level 1
Reference-current	5	2	1	1	1	1	1	1	2	1	1
Reference-representative	3										1
Solution(GI)		1									
Self(GI)			1		1						
Random(GI)		1		1							
All population(DVM)								1	2		
Derived statistic (DVM)	3										
Representative(DVM)	4									1	1
Random movement(DVM)	3		1		1		1				1
Arbitrary Invididual(DVM)	1										
Select all							1	1		1	
Fitness influenced selection				1		1			2		
Random selection	8	2									2
Parameter-based selection	8	2	1	1	1						2
Allow all	8	2		1				1	2	1	
Parameter influenced			1		1	1					
Allow better than current	4						1				2
Allow best of all candidates	2										
Discard Individuals			1		1	1					
Identity heuristic		1				1					
Iteration activated						1					
Personal best in update										1	

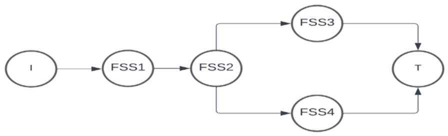

Figure 10: IT diagram for FSS metaheuristic-algorithm

Figure 11: IT diagram for PSO metaheuristic-algorithm

Table 11: Overall similarity values between metaheuristic-algorithms under consideration, using the proposed similarity measure.

Metaheuristic-algorithms	HHO	CRO	FSS	PSO	BA
HHO	1	0.0667	0.0377	0.1111	0.1695
CRO	0.0667	1	0.1589	0.05	0.09
FSS	0.0377	0.1589	1	0.1667	0.1429
PSO	0.1111	0.05	0.1667	1	0.4
BA	0.1695	0.09	0.1429	0.4	1

5 Discussion

The signal flow diagram required some modifications before being used in the similarity measure. The changes made were representing heuristics as features instead of using their names, as well as including the nodes *I* and *T* for convenience. The decision to represent heuristics as features is so that heuristics can be contrasted in terms of their attributes instead of regarding them as different due to having different names.

There are commonalities between the proposed similarity measure and the similarity measure based on the pool template representation in [4]. These include the use of features for representing heuristics, focusing on metaheuristic-algorithm components and some aspects underlying the similarity calculation (i.e., a feature-based similarity calculation that partly resembles the Jaccard index). The distinguishing elements of our proposed similarity measure are that we consider the arrangement of heuristic components, the usage of bags instead of sets to accommodate for repeated heuristic features in the similarity calculation, and the commutativity of the similarity measure.

Thus far, the similarity measure uses the components enclosed between the *I* and *T* nodes; however, the similarity measure can be extended to accommodate initializer and finalizer heuristics.

The digraph representation in [29] uses pre-selector, perturbator, and post-selector heuristics in its decision space; however, the work done in creating the signal flow diagram used only perturbators and post-selectors in the construction of a search operator heuristic. The feature vectors used in the current study employ features regarding pre-selector, perturbator, and post-selector heuristics, as well as some miscellaneous information.

The feature vector format used to showcase the similarity calculation is not a standard. Most of the features were selected in an ad-hoc manner, and thus the vector format has to be improved in future research. Constructing taxonomies using rigorous taxonomy-building methods increases the likelihood of having a robust and comprehensive set of features that can be extracted using methods like TAXONOG-IMC [2]. Different feature vector formats may entail that alternative, appropriate similarity measures will have to be used instead of the one implemented for the showcasing of the similarity calculation. Combining the vectors of each level for all branches of a metaheuristic-algorithm is done by simply adding them up. As feature vector formats incorporated in future research change, so may the method of combining the feature vectors and, subsequently the underlying similarity formula.

6 Conclusion

Metaheuristic-algorithm similarity determined via component-wise calculations may prove to be a valuable avenue for research, especially when the theory that bridges the gap between component-wise similarity and performance similarity starts to develop as it appears to be with [14].

In this study, a new metaheuristic-algorithm similarity measure based on the signal flow diagram representation of metaheuristic-algorithms is presented. It incorporates more structural information in the similarity calculation than a previous component-wise similarity measure using a pool template and can be extended to cover a comprehensive set of metaheuristic components – though, in this study, the pre-selector, perturbator, and post-selector heuristics were mainly given focused.

The method to calculate the similarity between two metaheuristic-algorithms is provided in step form accompanied by a calculation demonstration. Thereafter similarity scores of five concrete metaheuristic-algorithm implementations from the NiaPy library are calculated using the proposed similarity measure.

Future research directions are developing the theory that describes relationships between metaheuristic-component data and metaheuristic-performance data. Robust and comprehensive features are needed, as well as underlying similarity calculations that consider features that are related in some manner.

REFERENCES

[1] Achary, T. et al. 2022. Towards Rigorous Foundations for Metaheuristic Research: *Proceedings of the 14th International Joint Conference on Computational Intelligence* (Valletta, Malta, 2022), 151–157.

[2] Achary, T. and Pillay, A.W. 2022. A Taxonomy Guided Method to Identify Metaheuristic Components. *Computational Science – ICCS 2022*. D. Groen et al., eds. Springer International Publishing. 484–496.

[3] Aranha, C. et al. 2021. Metaphor-based metaheuristics, a call for action: the elephant in the room. *Swarm Intelligence*. (Nov. 2021). DOI:https://doi.org/10.1007/s11721-021-00202-9.

[4] de Armas, J. et al. 2021. Similarity in metaheuristics: a gentle step towards a comparison methodology. *Natural Computing*. (Feb. 2021). DOI:https://doi.org/10.1007/s11047-020-09837-9.

[5] Birattari, M. et al. 2003. Classification of Metaheuristics and Design of Experiments for the Analysis of Components. (Mar. 2003).

[6] Calégari, P. et al. 1999. A Taxonomy of Evolutionary Algorithms in Combinatorial Optimization. *Journal of Heuristics*. 5, 2 (1999), 145–158. DOI:https://doi.org/10.1023/A:1009625526657.

[7] Costa, L. da F. 2021. Further Generalizations of the Jaccard Index. (2021). DOI:https://doi.org/10.48550/ARXIV.2110.09619.

[8] Cruz-Duarte, J.M. et al. 2021. Automated Design of Unfolded Metaheuristics and the Effect of Population Size. *2021 IEEE Congress on Evolutionary Computation (CEC)* (Kraków, Poland, Jun. 2021), 1155–1162.

[9] Cruz-Duarte, J.M. et al. 2021. Global Optimisation through Hyper-Heuristics: Unfolding Population-Based Metaheuristics. *Applied Sciences*. 11, 12 (Jun. 2021), 5620. DOI:https://doi.org/10.3390/app11125620.

[10] Cruz-Duarte, J.M. et al. 2020. Towards a Generalised Metaheuristic Model for Continuous Optimisation Problems. *Mathematics*. 8, 11 (Nov. 2020), 2046. DOI:https://doi.org/10.3390/math8112046.

[11] Gandomi, A.H. and Alavi, A.H. 2012. Krill herd: A new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*. 17, 12 (Dec. 2012), 4831–4845. DOI:https://doi.org/10.1016/j.cnsns.2012.05.010.

[12] Hatamlou, A. 2014. Heart: a novel optimization algorithm for cluster analysis. *Progress in Artificial Intelligence*. 2, 2–3 (Jun. 2014), 167–173. DOI:https://doi.org/10.1007/s13748-014-0046-5.

[13] Ifenthaler, D. 2012. Measures of Similarity. *Encyclopedia of the Sciences of Learning*. N.M. Seel, ed. Springer US. 2147–2150.

[14] Kostovska, A. et al. 2023. Using Knowledge Graphs for Performance Prediction of Modular Optimization Algorithms. arXiv.

[15] Krasnogor, N. and Smith, J. 2005. A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issues. *IEEE Transactions on Evolutionary Computation*. 9, 5 (Oct. 2005), 474–488. DOI:https://doi.org/10.1109/TEVC.2005.850260.

[16] Mirjalili, S. 2015. The Ant Lion Optimizer. *Advances in Engineering Software*. 83, (May 2015), 80–98. DOI:https://doi.org/10.1016/j.advengsoft.2015.01.010.

[17] Molina, D. et al. 2020. Comprehensive Taxonomies of Nature- and Bio-inspired Optimization: Inspiration Versus Algorithmic Behavior, Critical Analysis Recommendations. *Cognitive Computation*. 12, 5 (Sep. 2020), 897–939. DOI:https://doi.org/10.1007/s12559-020-09730-8.

[18] Motta, F.A.A. et al. 2018. A Hybrid Approach of Grammar-based Genetic Programming and Differential Evolution for Symbolic Regression. *Proceedings XIII Brazilian Congress on Computational Inteligence* (Jan. 2018), 1–12.

[19] Mweshi, G. and Pillay, N. 2019. A Grammatical Evolution Approach for the Automated Generation of Perturbative Heuristics. *2019 IEEE Congress on Evolutionary Computation (CEC)* (Wellington, New Zealand, Jun. 2019), 2642–2648.

[20] Pereira Junior, J. and Aranha, C. 2022. Empirical Similarity Measure for Metaheuristics. *Bioinspired Optimization Methods and Their Applications*. M. Mernik et al., eds. Springer International Publishing. 69–83.

[21] Razmjooy, N. et al. 2016. A New Meta-Heuristic Optimization Algorithm Inspired by FIFA World Cup Competitions: Theory and Its Application in PID Designing for AVR System. *Journal of Control, Automation and Electrical Systems*. 27, 4 (Aug. 2016), 419–440. DOI:https://doi.org/10.1007/s40313-016-0242-6.

[22] Samanthula, B.K. and Jiang, W. 2016. Secure Multiset Intersection Cardinality and its Application to Jaccard Coefficient. *IEEE Transactions on Dependable and Secure Computing*. 13, 5 (Sep. 2016), 591–604. DOI:https://doi.org/10.1109/TDSC.2015.2415482.

[23] Sörensen, K. 2015. Metaheuristics-the metaphor exposed. *International Transactions in Operational Research*. 22, 1 (Jan. 2015), 3–18. DOI:https://doi.org/10.1111/itor.12001.

[24] Sörensen, K. and Glover, F.W. 2013. Metaheuristics. *Encyclopedia of Operations Research and Management Science*. S.I. Gass and M.C. Fu, eds. Springer US. 960–970.

[25] Tzanetos, A. and Dounias, G. 2021. Nature inspired optimization algorithms or simply variations of metaheuristics? *Artificial Intelligence Review*. 54, 3 (Mar. 2021), 1841–1862. DOI:https://doi.org/10.1007/s10462-020-09893-8.

[26] Vrbančič, G. et al. 2018. NiaPy: Python microframework for building nature-inspired algorithms. *Journal of Open Source Software*. 3, 23 (Mar. 2018), 613. DOI:https://doi.org/10.21105/joss.00613.

[27] Wang, G.-G. et al. 2019. Monarch butterfly optimization. *Neural Computing and Applications*. 31, 7 (Jul. 2019), 1995–2014. DOI:https://doi.org/10.1007/s00521-015-1923-y.

[28] Zhang, Y.-W. and Halgamuge, S.K. 2018. Metaheuristic Algorithm Similarity Analysis Based on Performance Metric Mapping of Fractional Ranking. *2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)* (Colombo, Sri Lanka, Dec. 2018), 1–6.

[29] Zhao, Q. et al. 2022. AutoOpt: A Methodological Framework of Automatically Designing Metaheuristic Algorithms for Optimization Problems. (2022). DOI:https://doi.org/10.48550/ARXIV.2204.00998.

[30] 2022. Jaccard index. *Wikipedia*.

CHAPTER 5: Conclusions and Future Work

5.1. Conclusions

There are several issues affecting the metaheuristic research field: the obfuscating nature of metaphors and non-standard terminology makes it difficult to position new research amongst existing literature. The poor experimental setups and comparisons creates challenges in reusing existing data. The ever-growing flood of supposedly novel metaheuristics makes evaluating and choosing fit for purpose metaheuristics difficult. Several works in the literature have proposed and defended a component-based view of metaheuristics.

The central problem for the component-based view is the identification of components of a metaheuristic. This work demonstrated the use of taxonomies to guide the identification of metaheuristic components. We developed a general and rigorous method, TAXONOG-IMC, that takes as input an appropriate taxonomy and guides the user to identify components. The method is described in detail, an example application of the method is given, and an analysis of its usefulness is provided. The analysis shows that the method is effective and provides insights that are not possible without the proper identification of the components.

This work also argues for formal, mathematically sound representations of metaheuristics. It introduces and defends a formal representation that leverages the component based view. A mathematical formulation of a core philosophy of what a metaheuristic is, is proposed. It is highlighted in the literature, that metaheuristics and metaheuristic-algorithms are often both referred to as metaheuristics which allows novel metaheuristic-algorithms to be published as novel metaheuristics but it is then also stated that novelty metaheuristic-algorithm is hardly a significant contribution. Thus rigorously differentiating between metaheuristic and metaheuristic-algorithm is important

The third contribution made in this work demonstrates that a representation technique based on a component based view is able to provide the basis for a similarity measure. We present a method of measuring similarity between two metaheuristic-algorithms, based on their representations as signal flow diagrams.

Our findings indicate that the component based view of metaheuristics provides valuable insights and allows for more robust analysis, classification and comparison.

5.2. Future Work

Increasing theoretical insight, better analytical techniques, and solid foundations should be a top priority of metaheuristic research. Pursuing these goals should consider the steps necessary to do so, e.g., developing robust and comprehensive taxonomies for the metaheuristic field – which are lacking. Theory to bridge the gap between component-data and performance-data is likely to be important. Investigating the conditions under which metaheuristic similarity in terms of components matches up with metaheuristic similarity in terms of performance will aid in selecting metaheuristics or metaheuristic-algorithms for application to real world problems.

References

1. Sörensen, K.: Metaheuristics-the metaphor exposed. International Transactions in Operational Research. 22, 3–18 (2015). https://doi.org/10.1111/itor.12001.
2. Aranha, C., Camacho Villalón, C.L., Campelo, F., Dorigo, M., Ruiz, R., Sevaux, M., Sörensen, K., Stützle, T.: Metaphor-based metaheuristics, a call for action: the elephant in the room. Swarm Intelligence. (2021). https://doi.org/10.1007/s11721-021-00202-9.
3. García-Martínez, C., Gutiérrez, P.D., Molina, D., Lozano, M., Herrera, F.: Since CEC 2005 competition on real-parameter optimisation: a decade of research, progress and comparative analysis's weakness. Soft Comput. 21, 5573–5583 (2017). https://doi.org/10.1007/s00500-016-2471-9.
4. Tzanetos, A., Dounias, G.: Nature inspired optimization algorithms or simply variations of metaheuristics? Artificial Intelligence Review. 54, 1841–1862 (2021). https://doi.org/10.1007/s10462-020-09893-8.
5. Molina, D., Poyatos, J., Ser, J.D., García, S., Hussain, A., Herrera, F.: Comprehensive Taxonomies of Nature- and Bio-inspired Optimization: Inspiration Versus Algorithmic Behavior, Critical Analysis Recommendations. Cognitive Computation. 12, 897–939 (2020). https://doi.org/10.1007/s12559-020-09730-8.
6. de Armas, J., Lalla-Ruiz, E., Tilahun, S.L., Voß, S.: Similarity in metaheuristics: a gentle step towards a comparison methodology. Natural Computing. (2021). https://doi.org/10.1007/s11047-020-09837-9.
7. Stegherr, H., Heider, M., Hähner, J.: Classifying Metaheuristics: Towards a unified multi-level classification system. Natural Computing. (2020). https://doi.org/10.1007/s11047-020-09824-0.
8. Birattari, M., Paquete, L., Stützle, T.: Classification of Metaheuristics and Design of Experiments for the Analysis of Components. (2003).
9. Hooker, J.N.: Testing heuristics: We have it all wrong. J Heuristics. 1, 33–42 (1995). https://doi.org/10.1007/BF02430364.
10. Campelo, F., Aranha, C.: Sharks, Zombies and Volleyball: Lessons from the Evolutionary Computation Bestiary. In: Proceedings of the LIFELIKE Computing Systems Workshop 2021. CEUR-WS.org (2021).
11. Glover, F.: Future paths for integer programming and links to artificial intelligence. Computers & Operations Research. 13, 533–549 (1986). https://doi.org/10.1016/0305-0548(86)90048-1.
12. Voß, S.: Meta-heuristics: The State of the Art. In: Goos, G., Hartmanis, J., van Leeuwen, J., and Nareyek, A. (eds.) Local Search for Planning and Scheduling. pp. 1–23. Springer Berlin Heidelberg, Berlin, Heidelberg (2001). https://doi.org/10.1007/3-540-45612-0-1.
13. Sörensen, K., Glover, F.W.: Metaheuristics. In: Gass, S.I. and Fu, M.C. (eds.) Encyclopedia of Operations Research and Management Science. pp. 960–970. Springer US, Boston, MA (2013). https://doi.org/10.1007/978-1-4419-1153-7_1167.
14. Lones, M.A.: Mitigating Metaphors: A Comprehensible Guide to Recent Nature-Inspired Algorithms. SN COMPUT. SCI. 1, 49 (2020). https://doi.org/10.1007/s42979-019-0050-8.
15. Liu, B., Wang, L., Liu, Y., Wang, S.: A unified framework for population-based metaheuristics. Annals of Operations Research. 186, 231–262 (2011). https://doi.org/10.1007/s10479-011-0894-3.
16. Cruz-Duarte, J.M., Ortiz-Bayliss, J.C., Amaya, I., Shi, Y., Terashima-Marín, H., Pillay, N.: Towards a Generalised Metaheuristic Model for Continuous Optimisation Problems. Mathematics. 8, 2046 (2020). https://doi.org/10.3390/math8112046.
17. De Araujo Pessoa, L.F., Wagner, C., Hellingrath, B., Buarque De Lima Neto, F.: Component Analysis Based Approach to Support the Design of Meta-Heuristics for MLCLSP Providing Guidelines. In: 2015 IEEE Symposium Series on Computational Intelligence. pp. 1029–1038. IEEE, Cape Town (2015). https://doi.org/10.1109/SSCI.2015.149.
18. Stegherr, H., Heider, M., Luley, L., Hähner, J.: Design of large-scale metaheuristic component studies. In: Proceedings of the Genetic and Evolutionary Computation

Conference Companion. pp. 1217–1226. ACM, Lille France (2021). https://doi.org/10.1145/3449726.3463168.
19. Peres, F., Castelli, M.: Combinatorial Optimization Problems and Metaheuristics: Review, Challenges, Design, and Development. Applied Sciences. 11, 6449 (2021). https://doi.org/10.3390/app11146449.
20. Calégari, P., Coray, G., Hertz, A., Kobler, D., Kuonen, P.: A Taxonomy of Evolutionary Algorithms in Combinatorial Optimization. Journal of Heuristics. 5, 145–158 (1999). https://doi.org/10.1023/A:1009625526657.
21. Raidl, G.R.: A Unified View on Hybrid Metaheuristics. In: Almeida, F., Blesa Aguilera, M.J., Blum, C., Moreno Vega, J.M., Pérez Pérez, M., Roli, A., and Sampels, M. (eds.) Hybrid Metaheuristics. pp. 1–12. Springer Berlin Heidelberg, Berlin, Heidelberg (2006). https://doi.org/10.1007/11890584_1.