mathematics

IMPACT FACTOR 2.3

CITESCORE 4.0

Review

Continuous Metaheuristics for Binary Optimization Problems: An Updated Systematic Literature Review

Marcelo Becerra-Rozas, José Lemus-Romani, Felipe Cisternas-Caneo, Broderick Crawford, Ricardo Soto, Gino Astorga, Carlos Castro and José García

MDPI

https://doi.org/10.3390/math11010129

mathematics

MDPI

Review

Continuous Metaheuristics for Binary Optimization Problems: An Updated Systematic Literature Review

Marcelo Becerra-Rozas [1,*], José Lemus-Romani [2], Felipe Cisternas-Caneo [1], Broderick Crawford [1,*], Ricardo Soto [1], Gino Astorga [3], Carlos Castro [4] and José García [5]

[1] Escuela de Ingeniería Informática, Pontificia Universidad Católica de Valparaíso, Avenida Brasil 2241, Valparaíso 2362807, Chile

[2] Escuela de Construcción Civil, Pontificia Universidad Católica de Chile, Avenida Vicuña Mackenna 4860, Macul, Santiago 7820436, Chile

[3] Escuela de Negocios Internacionales, Universidad de Valparaíso, Viña del Mar 2572048, Chile

[4] Departamento de Informática, Universidad Técnica Federico Santa María, Avenida España 1680, Valparaíso 2390123, Chile

[5] Escuela de Ingeniería de Construcción y Transporte, Pontificia Universidad Católica de Valparaíso, Avenida Brasil 2147, Valparaíso 2362804, Chile

* Correspondence: marcelo.becerra.r@mail.pucv.cl (M.B.-R.); broderick.crawford@pucv.cl (B.C.)

Abstract: For years, extensive research has been in the binarization of continuous metaheuristics for solving binary-domain combinatorial problems. This paper is a continuation of a previous review and seeks to draw a comprehensive picture of the various ways to binarize this type of metaheuristics; the study uses a standard systematic review consisting of the analysis of 512 publications from 2017 to January 2022 (5 years). The work will provide a theoretical foundation for novice researchers tackling combinatorial optimization using metaheuristic algorithms and for expert researchers analyzing the binarization mechanism's impact on the metaheuristic algorithms' performance. Structuring this information allows for improving the results of metaheuristics and broadening the spectrum of binary problems to be solved. We can conclude from this study that there is no single general technique capable of efficient binarization; instead, there are multiple forms with different performances.

Keywords: combinatorial problems; continuous metaheuristics; binarization; discretization methods; review

MSC: 68-02

check for updates

Citation: Becerra-Rozas, M.; Lemus-Romani, J.; Cisternas-Caneo F.; Crawford, B.; Soto, R.; Astorga, G.; Castro, C.; García, J. Continuous Metaheuristics for Binary Optimization Problems: An Updated Systematic Literature Review. *Mathematics* **2023**, *11*, 129. https://doi.org/10.3390/math11010129

Academic Editor: Simeon Reich

Received: 30 November 2022
Revised: 17 December 2022
Accepted: 22 December 2022
Published: 27 December 2022

1. Introduction

Information technologies have experienced exponential growth, generating multiple optimization problems. These problems can be classified into two main subfields: stochastic and deterministic optimization problems. The latter includes (a) unrestricted and constrained continuous optimization; (b) discrete optimization, which can be divided into integer programming and combinatorial optimization. The manuscript's focus is combinatorial optimization, which deals with problems where the set of feasible solutions is discrete or can be reduced to a discrete set.

The reason for focusing on combinatorial optimization problems is because of their great impact on real-world problems. For example, several authors have modeled combinatorial problems such as Emergency Humanitarian Logistics [1], Manufacturing Cell Design Problem [2], Brain Tumors Diagnosis and Prediction [3], Airline and Plant Location Selection [4], Antenna Positioning Problem [5], Disaster Management System [6], Electrocardiogram Arrhythmia Classification [7], Dynamic Vehicle Routing Problem [8], Sales Territory Design Problem [9], Buttressed Walls Problem [10], Facility Location Problem [11], Gene Selection Problem [12], Hierarchized Steiner Tree Problems in Telecommunication Networks [13], Susceptible-Exposed-Infectious Vigilant Epidemic Model [14].

Many techniques for solving combinatorial optimization problems can be classified into exact and approximation methods. The former includes branch and cut [15,16] and branch and bound [17–19]. The latter includes heuristic and metaheuristic approaches. The latter has gained enormous popularity over exact methods due to their simplicity and the results' robustness when the combinatorial problem is of high dimension.

Thanks to the no-free lunch theorem [20], which tells us that there is no supreme algorithm that solves all problems, researchers develop new metaheuristics every year. This great interest led to the development of metaheuristics with very good performance, such as Particle Swarm Optimization [21], Ant Colony Optimization [22], Artificial Bee Colony [23], Bat Algorithm [24], Cuckoo Search [25], Firefly Algorithm [26], Grey Wolf Optimizer [27], Sine Cosine Algorithm [28], Whale Optimization Algorithm [29], Dragonfly Algorithm [30], among others.

These metaheuristics have one feature in common: They were designed to solve problems with a continuous domain of variables. Therefore, the need arises to perform a binarization process. With binarization, we can use popular continuous metaheuristics with very good performance on binary combinatorics problems.

In this context and given the importance that metaheuristics have taken on in solving complex combinatorial problems, this paper presents a systematic review of articles published between 2017 and 2022 to deepen and extract techniques or mechanisms to binarize metaheuristic algorithms that they operate in continuous search spaces. This manuscript is an update of the literature review presented in [31]. It will provide a theoretical foundation for young researchers tackling combinatorial optimization using metaheuristic algorithms and for experienced researchers looking at the impact of the binarization mechanism on the performance of metaheuristic algorithms. The systematic review will be guided by the procedure described in 2004 by Kitchenham [32].

The systematic literature review results were analyzed from the following points of view:

1. Journals publishing paper on binarized continuous metaheuristics.
2. Scientific production by country.
3. Combinatorial problems solved by the authors.
4. Continuous metaheuristics used to solve binary combinatorial problems.
5. Classification and definition of the different binarization techniques

A brief summary of the structure of the content of the following sections: Section 2 explains the procedure and methodology used to perform this systematic review. Section 3 presents the results extracted and analyzed. In Section 4 we answer our research questions (presented in the methodology) and finally our conclusions in Section 5.

2. Methodology

This chapter is structured as follows, first, in Section 2.1, we identify the research questions that this work intends to clarify; the research questions guide and create a strong link of ideas that pivot on the main objective of the study. Next, in Section 2.2, the sources or search engines used to extract results are defined, then the search terms to be used are determined based on the summaries of some of the primary publications that are already available. Once the results are obtained, an a priori analysis is made to show trends in the results obtained. Then, exclusion and inclusion criteria are defined for the results in Section 2.3, focusing on the content of the title, keywords, and abstract of the article. Finally, each paper included in the review is assigned a score or value according to the scoring criteria Section 2.4. The above process is represented in Figure 1. In the last Section 2.5, we can find the data collected and used for our study and the presentation of the bigrams and maps shown in the later sections.

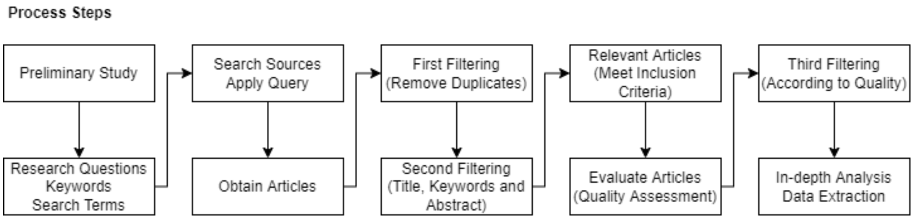

Figure 1. Systematic Literature Review Process Performed.

2.1. Research Questions

The first question from which the above-stated motivation of this study derives is: How to make a complete picture of the research on binarization techniques solving combinatorial problems in the binary domain? To answer it, we formulated two research questions (RQs) to consider in the collected literature. These questions are:

RQ1 What continuous metaheuristics have been used from 2017 to date to solve binary combinatorial problems?

RQ2 What techniques or forms of binarization have been used in metaheuristics from 2017 to date to solve binary combinatorial problems?

To address RQ-1, we identified the number of articles published per year, the journal/conference that published them, and whether they referred to continuous metaheuristics to solve problems that belonged to a binary domain. Regarding RQ-2, we considered the scope of the study, i.e., what techniques they used to transform a metaheuristic that specializes in working with continuous domains to a binary domain.

2.2. Search Process

The search process was a search through queries in well-known search engines such as Web of Science (Clarivate) and Scopus since 2017. It should be noted that only approved scientific articles in digital format and written in English were considered for the review.

To answer the questions posed and find the articles that interest us, search terms (STs) are used in the engines mentioned above:

ST1 (“binarization” OR “binary”)

ST2 (“optimization” OR “optimizer” OR “combinatorial” AND “problem*” OR “combinatorial” AND “optimization”)

ST3 (“metaheuristic*” OR “continuous” AND “metaheuristic*”)

Using such search terms, we have elaborated in the sources the respective queries subject to the dates Jan 1st, 2017 and Jan 30th, 2022.

QRY Scopus: (TITLE-ABS-KEY (binarization OR binary) AND TITLE-ABS-KEY (optimization OR optimizer OR combinatorial AND problem* OR combinatorial AND optimization) AND TITLE-ABS-KEY (metaheuristic* OR continuous AND metaheuristic*))

QRY Web of Science: ALL=(binarization OR binary) AND ALL=(optimization OR optimizer OR combinatorial AND problem* OR combinatorial AND optimization) AND ALL=(metaheuristic* OR continuous AND metaheuristic*)

2.3. Inclusion and Exclusion Criteria

For the results obtained from Jan 1st, 2017, and Jan 30th, 2022, a first selection is made, which consists of removing duplicates of the sum of both queries. Those articles that are related to applications of continuous metaheuristics and solve binary combinatorial problems are selected.

- Use of continuous metaheuristics.
- Solving binary domain combinatorial problems.
- Binarize a continuous metaheuristic.

- Describes advances in the way of binarizing continuous metaheuristics.

Items excluded were for the following topics.

- Duplicate articles.
- Articles not related to metaheuristics.
- Articles other than English.

2.4. Quality Assessment

At the end of the selection and storage of articles, a second selection is made, which consists of reading the title, abstract, and keywords and assigning a score concerning the degree of usefulness of this study. This score is calculated by adding the score obtained in each inclusion and exclusion criteria.

QA1 Are the authors using continuous metaheuristics?
QA2 Are the authors solving a binary domain combinatorial problem?
QA3 Do the authors binarize a continuous metaheuristic?
QA4 The authors describe advances in the binarization of continuous metaheuristics?

The scoring procedure corresponds to a binary evaluation, where $Y = 1$, $N = 0$. The criteria are scored as follows:

QA1: **Yes**, authors use a continuous metaheuristic; **No** authors do not make use of continuous metaheuristics.
QA2: **Yes**, authors solve at least 1 problem with binary domain; **No**, authors solve discrete or continuous problems.
QA3: **Yes**, the authors use techniques to binarize a continuous metaheuristic, that is, they adapt the values of the real domain to binary to work with the binary problem: **No**, the authors do not apply techniques to binarize the metaheuristic or they use one from the literature and do not explain the procedure.
QA4: **Yes**, the authors propose new or novel techniques to perform binarization and transform continuous values to binary in a continuous metaheuristic: **No**, the authors propose nothing and stick to more traditional techniques for binarization.

2.5. Data Collection

The data extracted from each article studied were:

- Full reference
- Quality evaluation
- Problem solved
- Metaheuristic used
- Transfer function used
- Categorization for the binarization technique used
- Year of publication
- Journal
- Cites by articles
- Authors’ countries

3. Results

In this section, a set of graphs presents the most relevant information about the selected articles, such as the number of articles, the journals of origin and their respective countries, the general concepts addressed, the quantification of the QA factors, and the popularity factors.

3.1. General and Bigrams Analysis

This section aims to conduct a general analysis of the metadata obtained from the select articles. First, a visualization will be generated that identifies the journals or conferences that are published the most in terms of binarizations of metaheuristics, and then a visualization will be developed indicating the countries that have generated the most significant contribution in this area. This last graph considers the country of all the authors. A bigram is a series of two contiguous elements of a chain of tokens, which correspond to words in this instance. The purpose is to conduct a statistical analysis of the frequency distribution of these bigrams in the various abstracts under consideration. The initial display represents the Treemap. This seeks to determine the frequency of the most prevalent bigrams in each theme. The thematic map is then utilized; this graph combines the concepts of density (internal associations) and centrality (external associations), [33,34]. Finally, the visualization corresponds to conceptual maps and dendrograms. Conceptual structure visualization creates a conceptual structure map. Specifically, the Correspondence Analysis (CA) is performed on terms extracted from the summaries of the documents. In addition to analyzing the relationship between the terms hierarchically, the conceptual structure is also displayed through a dendrogram.

The main journals and conferences are shown in Figure 2. As can be seen from the results, "advances in intelligent systems and computing together" with "IEEE Access" are the ones that publish the most in the area. Subsequently, reading notes and communication in computer and information science appear. All are generated by conferences except in the case of IEEE Access. Figure 3 shows the main countries publishing in the binarization area. The main country corresponds to Chile with 299 appearances, followed by China with 183 appearances, and India with 143.

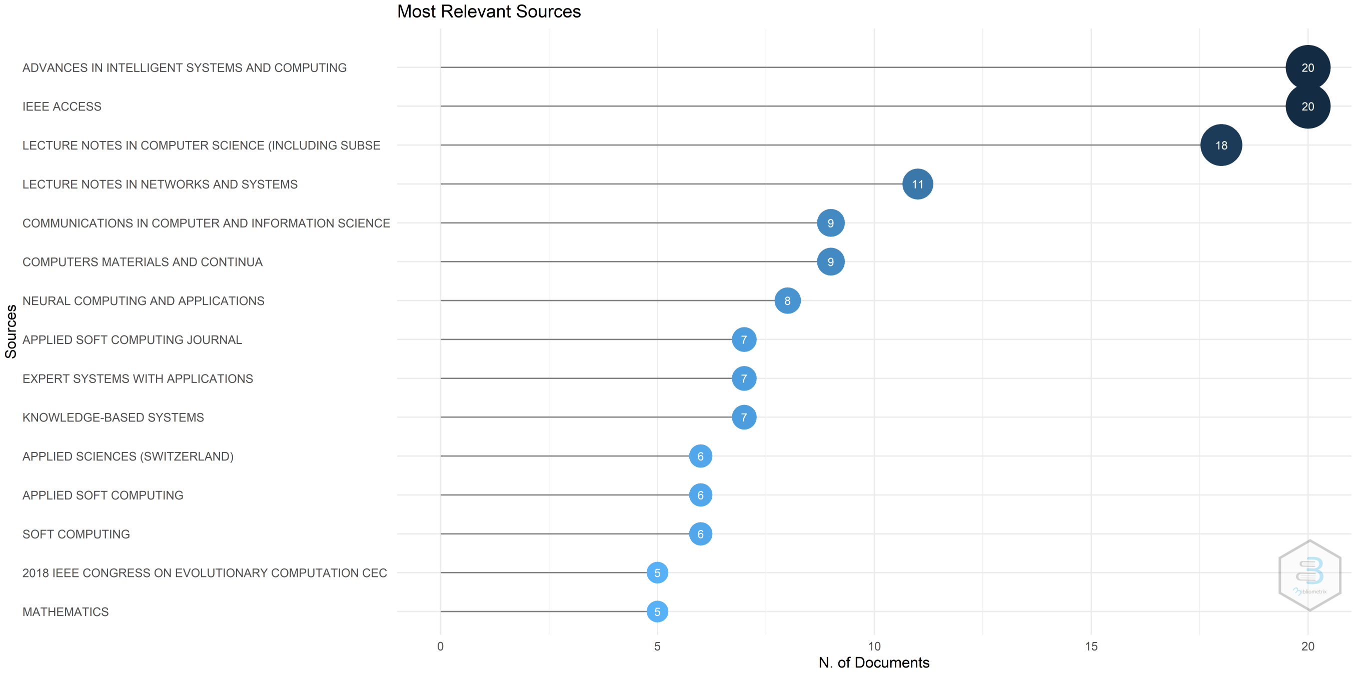

Figure 2. Most relevant journals.

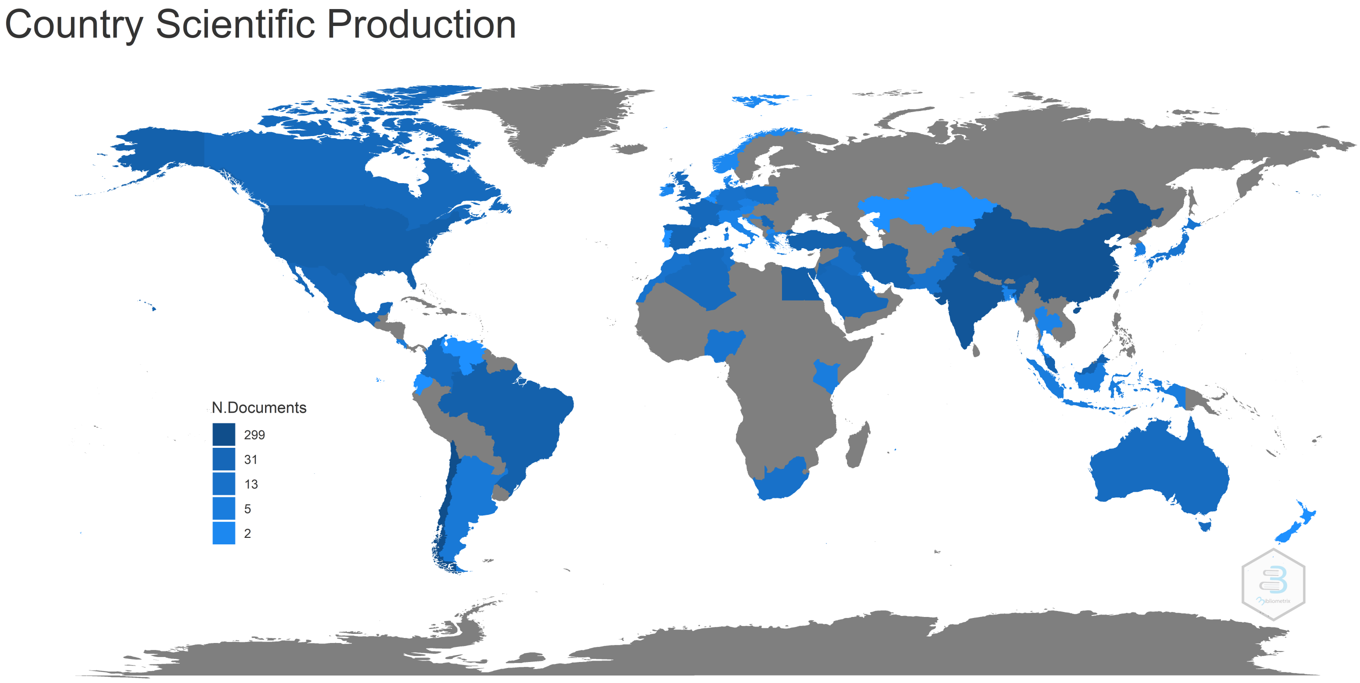

Figure 3. Scientific production by country.

When analyzing the Treemap Figure 4, three main concepts related to the techniques used stand out. In the first place, transfer functions appear as a binarization method, followed by machine learning techniques and, a little further down, techniques based on the concept of percentile. From the point of view of the problems used to verify the algorithms, the main problem corresponds to feature selection, followed by set-covering and unit commitment problems. When reviewing the techniques, we see that they are all of the swarm intelligence types where the most used corresponds to particle swarm, followed by bat algorithm, bee colony, and grey wolf. When analyzing the thematic map Figure 5, it is observed that a base theme, which has high centrality and low density, is related to the use of transfer functions. These transfer functions are associated with different swarm-like techniques. This can be seen in the lower right quadrant. On the other hand, the most important topics, which correspond to the high centrality and high-density quadrant (upper right quadrant), include machine learning techniques related to binarization and problems such as set covering and knapsack, as well as percentile techniques related to local search operators and swarm optimization techniques.

Finally, conceptual bigrams analysis, shown in Figure 6, returns two groups. In red, there is a cluster that mainly relates binarization techniques and swarm-type metaheuristics. Here, the transfer function and machine learning concepts stand out within the binarization techniques. On the other hand, the cluster in blue is mainly related to binary problems, among which set-covering and knapsack stand out.

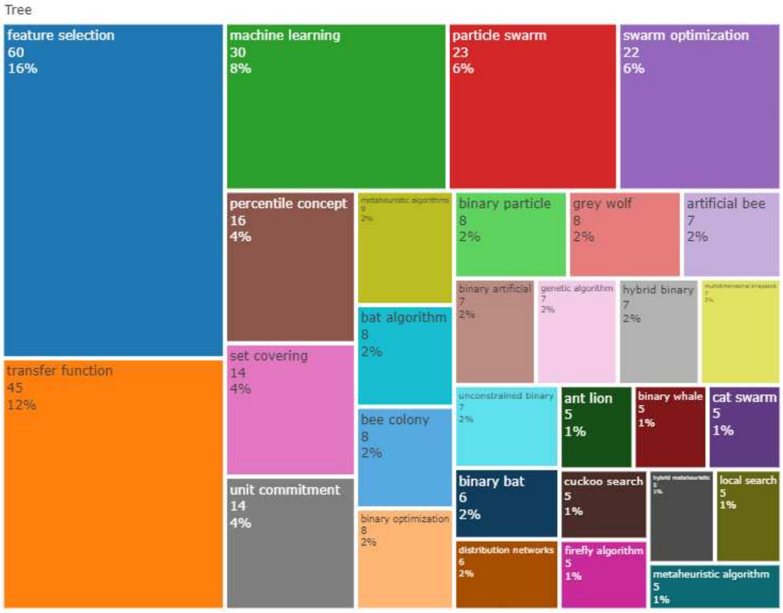

Figure 4. The most relevant bigrams, Treemap.

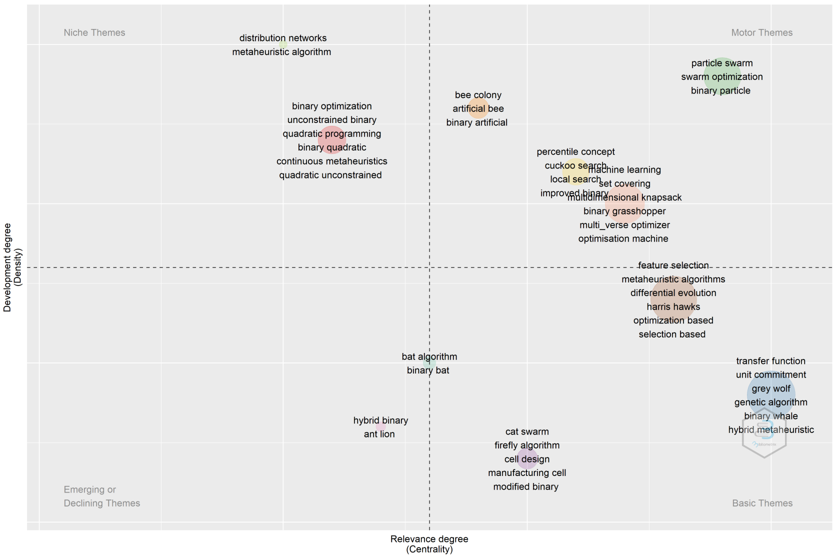

Figure 5. Clustering thematic bigrams analysis.

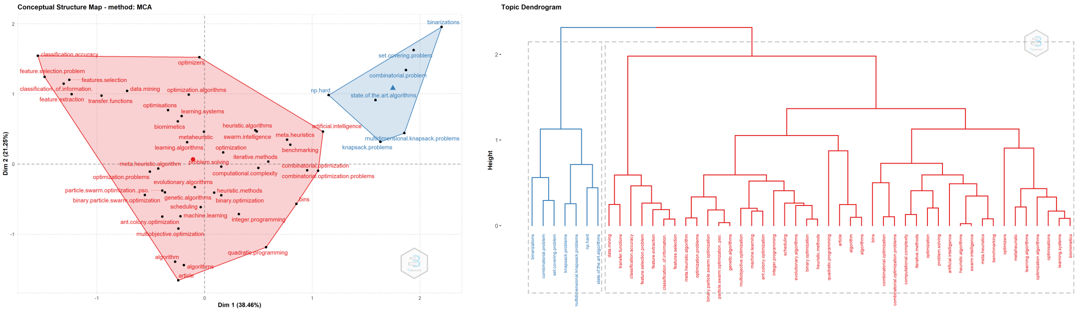

Figure 6. Conceptual bigrams analysis.

3.2. Search Results

Table 1 summarizes the results obtained from the search process shown in Figure 1. With the first search of the queries used, we obtained a total of 733 articles, from which it was necessary to remove the existing duplicates, leaving a total of 512 qualifying articles. For our second filter, we performed an analysis of the quality criteria defined in Section 2.4 based on the title, keywords, and abstract presented by the research, leaving a total of 283 potentially relevant articles for further analysis and to avoid false positives. From this list, we removed papers in a language other than English, those where it was impossible to obtain access to the article, and existing false positives. Finally, we obtained 195 unique studies.

Table 1. Sources searched for years 2017–2022 (including articles up to 30 January 2022).

	Amount of Results			
Used Query	**Sources Found**	**Assessed**	**Selected for In-Depth Analysis**	**Final Selectees**
QRY1	405	396	212	132
QRY2	328	116	71	63
Total	733	512	283	195

3.3. Quality Evaluation of Articles

As mentioned above, the quality of the research obtained was evaluated using the criteria defined in Section 2.4. The score of each article is present in Table 2, it is worth mentioning, which contains the selected articles. The last column shows the score the researchers agreed on; in this case, only those articles that achieved a score equal to or greater than 3 points. All disagreements were discussed and resolved. Based on the number of articles chosen, a total of 317 articles scored less than three or were discarded for the reasons specified in Section 2.3.

Table 2. Quality evaluation of articles.

Article	QA1	QA2	QA3	QA4	Total Score	Rater Agreement	ID	QA1	QA2	QA3	QA4	Total Score	Rater Agreement	ID	QA1	QA2	QA3	QA4	Total Score	Rater Agreement	ID	QA1	QA2	QA3	QA4	Total Score	Rater Agreement
A1 [35]	Y	Y	Y	Y	4	3	**A51** [36]	Y	Y	Y	Y	4	3	**A101** [37]	Y	Y	Y	N	3	3	**A151** [38]	Y	Y	Y	N	3	3
A2 [39]	Y	Y	Y	Y	4	3	**A52** [40]	Y	Y	Y	Y	4	3	**A102** [41]	Y	Y	Y	N	3	3	**A152** [42]	Y	Y	Y	N	3	3
A3 [43]	Y	Y	Y	N	3	3	**A53** [44]	Y	Y	Y	Y	4	3	**A103** [45]	Y	Y	Y	N	3	3	**A153** [46]	Y	Y	Y	N	3	3
A4 [47]	Y	Y	Y	Y	4	3	**A54** [48]	Y	Y	Y	Y	4	3	**A104** [49]	Y	Y	Y	Y	4	3	**A154** [50]	Y	Y	Y	N	3	3
A5 [51]	Y	Y	Y	N	3	3	**A55** [52]	Y	Y	Y	Y	4	3	**A105** [53]	Y	Y	Y	Y	4	3	**A155** [54]	Y	Y	Y	N	3	3
A6 [55]	Y	Y	Y	Y	4	3	**A56** [56]	Y	Y	Y	Y	4	3	**A106** [57]	Y	Y	Y	N	3	3	**A156** [58]	Y	Y	Y	N	3	3
A7 [59]	Y	Y	Y	Y	4	3	**A57** [60]	Y	Y	Y	Y	4	3	**A107** [61]	Y	Y	Y	Y	4	3	**A157** [62]	Y	Y	Y	Y	4	3
A8 [2]	Y	Y	Y	Y	4	3	**A58** [63]	Y	Y	Y	N	3	3	**A108** [64]	Y	Y	Y	Y	4	3	**A158** [65]	Y	Y	Y	Y	4	3
A9 [66]	Y	Y	Y	Y	4	3	**A59** [67]	Y	Y	Y	Y	4	3	**A109** [68]	Y	Y	Y	N	3	3	**A159** [69]	Y	Y	Y	N	3	3
A10 [70]	Y	Y	Y	N	3	3	**A60** [71]	Y	Y	Y	N	3	3	**A110** [72]	Y	Y	Y	N	3	3	**A160** [73]	Y	Y	Y	Y	4	3
A11 [74]	Y	Y	Y	Y	4	3	**A61** [75]	Y	Y	Y	N	3	3	**A111** [76]	Y	Y	Y	N	3	3	**A161** [77]	Y	Y	Y	N	3	3
A12 [78]	Y	Y	Y	N	3	3	**A62** [79]	Y	Y	Y	Y	4	3	**A112** [80]	Y	Y	Y	N	3	3	**A162** [81]	Y	Y	Y	Y	4	3
A13 [82]	Y	Y	Y	N	3	3	**A63** [3]	Y	Y	Y	N	3	3	**A113** [83]	Y	Y	Y	Y	4	3	**A163** [84]	Y	Y	Y	Y	4	3
A14 [85]	Y	Y	Y	Y	4	3	**A64** [86]	Y	Y	Y	N	3	3	**A114** [87]	Y	Y	Y	N	3	3	**A164** [12]	Y	Y	Y	N	3	3
A15 [88]	Y	Y	Y	Y	4	3	**A65** [89]	Y	Y	Y	N	3	3	**A115** [90]	Y	Y	Y	N	3	3	**A165** [91]	Y	Y	Y	N	3	3
A16 [92]	Y	Y	Y	Y	4	3	**A66** [93]	Y	Y	Y	N	3	3	**A116** [94]	Y	Y	Y	N	3	3	**A166** [95]	Y	Y	Y	N	3	3
A17 [96]	Y	Y	Y	N	3	3	**A67** [97]	Y	Y	Y	N	3	3	**A117** [98]	Y	Y	Y	N	3	3	**A167** [99]	Y	Y	Y	Y	4	3
A18 [100]	Y	Y	Y	N	3	3	**A68** [101]	Y	Y	Y	N	3	3	**A118** [102]	Y	Y	Y	N	3	3	**A168** [103]	Y	Y	Y	N	3	3
A19 [104]	Y	Y	Y	Y	4	3	**A69** [105]	Y	Y	Y	Y	4	3	**A119** [106]	Y	Y	Y	N	3	3	**A169** [107]	Y	Y	Y	N	3	3
A20 [31]	Y	Y	Y	Y	4	3	**A70** [108]	Y	Y	Y	N	3	3	**A120** [109]	Y	Y	Y	N	3	3	**A170** [110]	Y	Y	Y	N	3	3
A21 [111]	Y	Y	Y	N	3	3	**A71** [112]	Y	Y	Y	Y	4	3	**A121** [113]	Y	Y	Y	N	3	3	**A171** [114]	Y	Y	Y	N	3	3
A22 [115]	Y	Y	Y	Y	4	3	**A72** [116]	Y	Y	Y	N	3	3	**A122** [117]	Y	Y	Y	N	3	3	**A172** [118]	Y	Y	Y	Y	4	3
A23 [119]	Y	Y	Y	N	3	3	**A73** [120]	Y	Y	Y	N	3	3	**A123** [121]	Y	Y	Y	N	3	3	**A173** [122]	Y	Y	Y	N	3	3
A24 [123]	Y	Y	Y	Y	4	3	**A74** [124]	Y	Y	Y	Y	4	3	**A124** [125]	Y	Y	Y	N	3	3	**A174** [126]	Y	Y	Y	N	3	3
A25 [127]	Y	Y	Y	N	3	3	**A75** [128]	Y	Y	Y	N	3	3	**A125** [129]	Y	Y	Y	Y	4	3	**A175** [130]	Y	Y	Y	N	3	3
A26 [131]	Y	Y	Y	N	3	3	**A76** [132]	Y	Y	Y	N	3	3	**A126** [133]	Y	Y	Y	Y	4	3	**A176** [134]	Y	Y	Y	N	3	3
A27 [135]	Y	Y	Y	N	3	3	**A77** [136]	Y	Y	Y	N	3	3	**A127** [137]	Y	Y	Y	Y	4	3	**A177** [138]	Y	Y	Y	N	3	3
A28 [139]	Y	Y	Y	N	3	3	**A78** [140]	Y	Y	Y	N	3	3	**A128** [141]	Y	Y	Y	Y	4	3	**A178** [142]	Y	Y	Y	N	3	3
A29 [143]	Y	Y	Y	N	3	3	**A79** [144]	Y	Y	Y	N	3	3	**A129** [145]	Y	Y	Y	Y	4	3	**A179** [146]	Y	Y	Y	N	3	3
A30 [147]	Y	Y	Y	N	3	3	**A80** [148]	Y	Y	Y	N	3	3	**A130** [149]	Y	Y	Y	N	3	3	**A180** [150]	Y	Y	Y	N	3	3
A31 [151]	Y	Y	Y	N	3	3	**A81** [152]	Y	Y	Y	N	3	3	**A131** [153]	Y	Y	Y	Y	4	3	**A181** [5]	Y	Y	Y	Y	4	3
A32 [154]	Y	Y	Y	N	3	3	**A82** [155]	Y	Y	Y	N	3	3	**A132** [156]	Y	Y	Y	Y	4	3	**A182** [157]	Y	Y	Y	N	3	3
A33 [158]	Y	Y	Y	Y	4	3	**A83** [159]	Y	Y	Y	N	3	3	**A133** [160]	Y	Y	Y	Y	4	3	**A183** [161]	Y	Y	Y	N	3	3
A34 [162]	Y	Y	Y	N	3	3	**A84** [163]	Y	Y	Y	N	3	3	**A134** [164]	Y	Y	Y	Y	4	3	**A184** [165]	Y	Y	Y	Y	4	3
A35 [166]	Y	Y	Y	N	3	3	**A85** [167]	Y	Y	Y	N	3	3	**A135** [168]	Y	Y	Y	N	3	3	**A185** [169]	Y	Y	Y	Y	4	3
A36 [170]	Y	Y	Y	Y	4	3	**A86** [171]	Y	Y	Y	N	3	3	**A136** [172]	Y	Y	Y	Y	4	3	**A186** [10]	Y	Y	Y	Y	4	3
A37 [173]	Y	Y	Y	N	3	3	**A87** [174]	N	Y	Y	Y	3	3	**A137** [175]	Y	N	Y	Y	3	3	**A187** [176]	Y	Y	Y	Y	4	3
A38 [177]	Y	Y	Y	N	3	3	**A88** [178]	Y	Y	Y	N	3	3	**A138** [179]	Y	Y	Y	Y	4	3	**A188** [180]	Y	Y	Y	Y	4	3
A39 [181]	Y	Y	Y	N	3	3	**A89** [182]	Y	Y	Y	N	3	3	**A139** [183]	Y	Y	Y	Y	4	3	**A189** [184]	Y	Y	Y	Y	4	3
A40 [185]	Y	Y	Y	N	3	3	**A90** [186]	Y	Y	Y	N	3	3	**A140** [187]	Y	Y	Y	N	3	3	**A190** [188]	Y	Y	Y	Y	4	3
A41 [189]	Y	Y	Y	Y	4	3	**A91** [190]	Y	Y	Y	N	3	3	**A141** [191]	Y	Y	Y	N	3	3	**A191** [192]	Y	Y	Y	Y	4	3
A42 [193]	Y	Y	Y	N	3	3	**A92** [194]	Y	Y	Y	N	3	3	**A142** [195]	Y	Y	Y	Y	4	3	**A192** [196]	Y	Y	Y	Y	4	3
A43 [197]	Y	Y	Y	N	3	3	**A93** [198]	Y	Y	Y	Y	4	3	**A143** [199]	Y	Y	Y	Y	4	3	**A193** [200]	Y	Y	Y	Y	4	3
A44 [201]	Y	Y	Y	N	3	3	**A94** [202]	Y	Y	Y	N	3	3	**A144** [203]	Y	Y	Y	Y	4	3	**A194** [204]	Y	N	Y	Y	3	3
A45 [205]	Y	Y	Y	N	3	3	**A95** [206]	Y	Y	Y	N	3	3	**A145** [207]	Y	Y	Y	Y	4	3	**A195** [208]	Y	Y	Y	Y	4	3
A46 [209]	Y	Y	Y	N	3	3	**A96** [210]	Y	Y	Y	N	3	3	**A146** [211]	Y	Y	Y	Y	4	3							
A47 [212]	Y	Y	Y	N	3	3	**A97** [213]	Y	Y	Y	N	3	3	**A147** [214]	Y	Y	Y	N	3	3							
A48 [14]	Y	Y	Y	N	3	3	**A98** [215]	Y	Y	Y	Y	4	3	**A148** [216]	Y	Y	Y	N	3	3							
A49 [217]	Y	Y	Y	Y	4	3	**A99** [218]	Y	Y	Y	N	3	3	**A149** [219]	Y	Y	Y	N	3	3							
A50 [220]	Y	Y	Y	Y	4	3	**A100** [221]	Y	Y	Y	N	3	3	**A150** [7]	Y	Y	Y	N	3	3							

3.4. Quality Factors

The relationship between the quality score of a selected article and the date of publication of the article was investigated. The mean quality scores obtained from the studies in each year are shown in Table 3. According to this table, we can observe that there is an increasing trend in the use of continuous metaheuristics to solve problems in the binary domain, although in 2022, there are only 16 articles, it is necessary to remember that for this year, only the month of January was considered. Based on the trend, we can assume that there will be more articles by the end of the year than in 2021.

Table 3. Average quality scores for articles by publication date.

	Year						
	2017	**2018**	**2019**	**2020**	**2021**	**2022**	**Total**
Number of articles	15	22	33	53	56	16	195
Mean quality score	3.5	3.38	3.27	3.6	3.42	3.24	

3.5. Popularity Factors

To facilitate the reading of the metaheuristics, Table 4 has been created with the ID of each of the metaheuristics that were used in the respective works. On the other hand, the composition of the next tables is as follows: the first column indicates what we are analyzing (metaheuristics, problems, binarization technique used or transfer function), the next five columns correspond to the years that were considered in the study, from 2017 to 2021 (without considering the single month of January 2022, this way we avoid confusion with the values) and finally the last column corresponds to the total value of articles.

As well as quality, the relationship between the popularity of the problems, metaheuristics, binarization techniques, and transfer functions used in a selected article and the date of publication was investigated. According to Table 5, we can notice that the problems with more tendency to solve are Feature Selection, Set Covering Problem, and Knapsack Problem, while with Table 6 with the course of the years there is a tendency to use less conventional metaheuristics, as it is Particle Swarm Optimization or Cuckoo Search. The most popular techniques shown in Table 7 give indications that the most traditional is to use a simple transformation, which usually consists of two steps, going from continuous to discrete and then by means of an equation to binary, finally, the most popular transfer functions in the literature during the last five years, the most classical ones such as S and V type are maintained, however, there are some adventurous in using other less conventional ones as can be seen in the Table 8.

Table 4. Acronyms of the investigated metaheuristics.

ID	Metaheuristic	ID	Metaheuristic
AAA	Artificial Algae Algorithm	HOA	Horse Herd Optimization Algorithm
ABC	Artificial Bee Colony Algorithm	JAYA	JAYA
AFSA	Artificial Fish Swarm Optimization Algorithm	LSA	Lightning Search Algorithm
ALO	Ant Lion Optimization	MA	Monkey Search Algorithm
ASA	Artificial Sheep Algorithm	MAMH	Multi-Agent System as Metaheuristic
ASO	Atom Search Optimization	MFO	Moth-Flame Optimization
BA	Bat Algorithm	MOA	Magnetic Optimization Algorithm
BHO	Black Hole Optimization Algorithm	MRFO	Manta Ray Foraging Optimization Algorithm
BO	Bonobo Optimizer	MS	Moth Search
BSO	Brain Storm Optimization	MVO	Multi-verse Optimazer Algorithm
ChOA	Chimp Optimization Algorithm	OSA	Owl Search Algorithm
COA	Coyote Optimization Algorithm	PBO	Polar Bear Optimization Algorithm
CS	Cuckoo Search	PIO	Pigeon-Inspired Optimization Algorithm
CSA	Crow Search Algorithm	PO	Political Optimizer
CSO	Cat Swarm Optimization	PSO	Particle Swarm Optimization
DA	Dragonfly Algorithm	RA	Rao Algorithm
DE	Differential Evolution	RFSO	Red Fox Search Optimizer
EM	Electromagnetism-like Algorithm	SCA	Sine Cosine Algorithm
EO	Equilibrium Optimizer Algorithm	SFO	Sail Fish Optimization
EPO	Emperor Penguin Optimizer	SHO	Spotted Hyena Optimizer
EVOA	Egyptian Vulture Optimization Algorithm	SMA	Slime Mould Algorithm
FA	Firefly Algorithm	SMO	Spider Mokey Optimization
FMO	Fish Migration Optimization	SOA	Seagull Optimization Algorithm
FOA	Fruit Fly Optimization	SOS	Symbiotic Organism Search Algorithm
GBO	Gradient-Based Optimizer	SSA	Social Spider Algorithm
GEO	Golden Eagle Optimizer	SSA*	Salp Swarm Algorithm
Q GOA	Grasshopper Optimization Algorithm	TGA	Tree Growth Algorithm
GPC	Giza Pyramids Construction Algorithm	TLBO	Teaching-Learning-Based Optimization Algorithm
GSA	Gravitational Search Algorithm	WDO	Wind Driven Optimization Algorithm
GSK	Gaining-Sharing Knowledge-Based Optimization Algorithm	WOA	Whale Algorithm Optimization
GSO	Galactic Swarm ptimization	WSA	Weighted Superposition Attraction Algorithm
GWO	Grey Wolf Optimizer	WSA*	Wolf Search Algorithm
HGS	Hunger Games Search Optimization	WWO	Water Wave Optimization Algorithm
HHO	Harris Hawk Optimization	β HO	β -Hill Climbing Optimizer

Table 5. Problem per year.

Problem	2017	2018	2019	2020	2021	Total
Feature Selection	**3** [42,95,122]	**7** [161,162,165,166, 177,213,222]	**7** [46,76,101,111, 114,136,147]	**23** [38,54,74,80, 103,108,116,119, 125,132,134,135, 152,157,173,180, 185,187,189,194, 196,217,223]	**25** [7,36,39,45, 50,57,69,78,90, 110,126,127,138, 140,171,174,190–192,200,206,214, 216,224,225]	**65**
Set Covering Problem	**3** [44,53,55]	**2** [105,182]	**4** [83,168,183,199]	**8** [56,61,153,164, 198,215,218,220]	**11** [48,60,79,85, 104,137,145,170, 195,226,227]	**28**
Knapsack Problem	**3** [37,52,121]	**1** [98]	**6** [49,62,118,124, 197,228]	**5** [35,66,74,99,220]	**7** [39,65,203,229–232]	**22**
Multidimensional Knapsack Problem	-	**2** [112,160]	**1** [149]	**7** [74,94,156,158, 172,179,209]	**4** [75,133,141,176]	**14**
Unit Commitment	**1** [43]	-	**3** [93,178,207]	**2** [41,97]	**3** [148,151,221]	**9**
Classic Benchmark Function	**1** [175]	-	**2** [93,233]	**3** [115,234,235]	**1** [236]	**7**
Uncapacitated Facility Location Problem	-	**1** [51]	**1** [197]	**1** [123]	**1** [210]	**4**
Set Union Knapsack Problem	-	**1** [237]	-	**1** [123]	**1** [64]	**3**
Buttressed Walls Problem	-	-	-	**3** [10,188,211]	-	**3**
Binary Benchmarks Problems	-	**1** [205]	**2** [128,238]	-	-	**3**
Gene Selection Problem	-	-	**1** [12]	-	**1** [91]	**2**
Manufacturing Cell Design Problem	**1** [59]	-	**1** [2]	-	-	**2**
Crew Scheduling Problem	-	**1** [73]	**1** [183]	-	-	**2**
Security Audit Trails Analysis Problem	-	**1** [186]	**1** [102]	-	-	**2**
GNSS/Leveling points	-	-	-	-	**1** [169]	**1**
Hardware/Software Partitioning Problem	-	-	-	-	**1** [129]	**1**
Next Release Problem	-	-	-	-	**1** [72]	**1**
Scheduling Problem in Code-Base Computing	-	-	-	-	**1** [113]	**1**
Support Vector Classification	-	-	-	-	**1** [109]	**1**
Synthetic, Real Life and Higher-dimensional Datasets	-	-	-	-	**1** [40]	**1**
Workflow Scheduling	-	-	-	-	**1** [131]	**1**
Classification of Complex Diseases	-	-	-	**1** [71]	-	**1**
Design of Micro-Grooved Pipe	-	-	-	**1** [142]	-	**1**
Structural Topology Design Optimization	-	-	-	**1** [47]	-	**1**
Susceptible-Exposed-Infectious Vigilant Epidemic Model	-	-	-	**1** [14]	-	**1**
Transmission Expansion Planning Considering	-	-	-	**1** [143]	-	**1**

Table 5. *Cont.*

Problem	2017	2018	2019	2020	2021	Total
Welded Beam Design Problem	-	-	-	**1** [115]	-	**1**
Tension/Compression Design Problem	-	-	-	**1** [115]	-	**1**
Two-dimensional Bin Packing Problem	-	-	-	**1** [81]	-	**1**
Advertisement Problem	-	-	**1** [159]	-	-	**1**
Brain Tumors Diagnosis and Prediction Based	-	-	**1** [3]	-	-	**1**
Constrained Multiobjective Next Release Problem	-	-	**1** [154]	-	-	**1**
Distribution Network Reconfiguration with Large Number of Switches	-	-	**1** [212]	-	-	**1**
High Dimensional of Colon and Leukemia Datasets	-	-	**1** [63]	-	-	**1**
Load Restoration in Primary Distribution Networks	-	-	**1** [100]	-	-	**1**
Mesh-based Network-on-chip	-	-	**1** [89]	-	-	**1**
Optimal Scheduling of Appliances in Smart Home	-	-	**1** [120]	-	-	**1**
Binary Cutting Stock	-	**1** [70]	-	-	-	**1**
Computationally Efficient Electronic Nose System	-	**1** [96]	-	-	-	**1**
Cryptanalysis	-	**1** [155]	-	-	-	**1**
Fault Section Diagnosis of Power Systems	-	**1** [84]	-	-	-	**1**
Obnoxious p-median Problem	-	**1** [106]	-	-	-	**1**
Planning and Allocation Problem	-	**1** [181]	-	-	-	**1**
Antenna Positioning Problem	**1** [5]	-	-	-	-	**1**
Unconstrained Binary Quadratic Programming	**1** [117]	-	-	-	-	**1**
Compact Dual-Band Planar Monopole Antenna Design	-	-	-	-	**1** [230]	**1**
Electrocardiogram Arrhythmia Classification	-	-	-	-	**1** [7]	**1**
Electrocardiogram Signal Classification	-	-	**1** [46]	-	-	**1**
Distribution System	-	-	-	-	**1** [130]	**1**

Table 6. Metaheuristic per year.

Metaheuristic	2017	2018	2019	2020	2021	Total
PSO	**1** [52]	**10** [70,96,106,155,161,162,177,181,205,208]	**10** [3,12,89,100,120,146,154,183,204,207]	**11** [14,41,134,142,143,158,179,209,211,234,235]	**9** [39,40,69,130,138,191,214,232,236]	**41**
BA	**1** [5]	**1** [186]	**5** [63,102,147,212,228]	**6** [47,99,103,164,172,215]	**3** [40,219,230]	**16**
CS	**1** [55]	**3** [73,105,112]	**3** [178,183,199]	**5** [10,71,158,179,196]	**3** [48,64,176]	**15**
WOA	-	-	**2** [93,238]	**4** [38,97,115,185]	**7** [60,85,104,133,137,203,225]	**13**
GWO	-	-	**2** [63,197]	**3** [56,123,185]	**8** [109,110,126,131,169,170,214,227]	**13**
FA	**2** [59,92]	**1** [205]	**1** [111]	**1** [217]	**5** [65,79,129,141,145]	**10**
SCA	-	-	**2** [49,89]	**1** [119]	**4** [104,195,226,227]	**7**
HHO	-	-	**1** [76]	**1** [189]	**5** [36,126,127,206,227]	**7**
GSA	**1** [42]	**2** [165,177]	**3** [63,122,146]	**1** [198]	-	**7**
GOA	-	-	**3** [62,83,136]	**2** [153,156]	**2** [151,216]	**7**
ABC	-	**1** [98]	**2** [159,204]	**1** [187]	**2** [39,210]	6
BHO	**1** [55]	**1** [160]	-	**1** [188]	**2** [91]	**5**
DA	-	**1** [222]	-	-	**4** [40,91,229,231]	**5**
ALO	**1** [117]	-	**2** [118,168]	**1** [119]	-	**4**
CSA	-	**1** [213]	-	**4** [81,157,220]	-	**4**
CSO	**1** [44]	**1** [155]	**1** [2]	**1** [74]	-	4
MVO	**1** [53]	-	**2** [124,149]	**1** [54]	-	4
GSK	-	-	-	-	**2** [57,192]	**2**
WSA	-	**1** [205]	-	**1** [35]	-	**2**
JAYA	-	-	-	**1** [132]	**1** [113]	**2**
MA	-	**1** [51]	-	**1** [61]	-	**2**
MRFO	-	-	-	-	**2** [7,78]	**2**
PIO	-	-	**1** [128]	**1** [94]	-	**2**
SMO	-	-	**1** [114]	**1** [223]	-	2
SSA*	-	-	**1** [233]	-	**1** [200]	**2**
BSO	-	**2** [84,166]	-	-	-	**2**
AAA	-	-	-	-	**1** [72]	**1**
EO	-	-	-	**1** [194]	-	**1**
EPO	-	-	-	-	**1** [50]	**1**
GBO	-	-	-	-	**1** [190]	**1**
DE	-	-	-	**1** [142]	-	**1**
ASA	**1** [43]	-	-	-	-	**1**
ASO	-	-	-	**1** [180]	-	**1**
COA	-	-	-	**1** [152]	-	**1**
EM	-	**1** [182]	-	-	-	**1**
EVOA	**1** [59]	-	-	-	-	**1**

Table 6. *Cont.*

Metaheuristic	**2017**	**2018**	**2019**	**2020**	**2021**	**Total**
TGA	-	-	-	**1** [173]	-	**1**
FMO	-	-	-	-	**1** [148]	**1**
FOA	**1** [121]	-	-	-	-	**1**
GPC	-	-	-	-	**1** [140]	**1**
GSO	-	-	-	**1** [66]	-	**1**
LSA	**1** [175]	-	-	-	-	**1**
MAMH	-	-	-	-	**1** [90]	**1**
MFO	-	-	-	-	**1** [171]	**1**
MOA	-	-	**1** [46]	-	-	**1**
MS	-	**1** [237]	-	-	-	**1**
OSA	-	-	-	**1** [80]	-	**1**
PBO	-	-	-	-	**1** [221]	**1**
PO	-	-	-	**1** [116]	-	**1**
AFSA	-	-	**1** [204]	-	-	**1**
RA	-	-	-	-	**1** [224]	**1**
SFO	-	-	-	**1** [135]	-	**1**
SMA	-	-	-	-	**1** [75]	**1**
SOA	-	-	-	-	**1** [45]	**1**
SOS	-	-	**1** [101]	-	-	**1**
SPO	-	-	-	**1** [108]	-	**1**
SSA	**1** [37]	-	-	-	-	**1**
TLBO	-	-	-	**1** [218]	-	**1**
WSA*	**1** [95]	-	-	-	-	**1**
WWO	-	-	-	**1** [125]	-	**1**
βHO	-	-	-	-	**1** [174]	**1**

Table 7. Technique of Binarization per year.

Technique of Binarization	2017	2018	2019	2020	2021	Total
Simple Transformation	**11** [5,31,37,43,44,52,55,59,92,122,175]	**16** [70,96,106,155,161,162,165,166,177,181,182,186,205,213,222,237]	**26** [2,3,12,46,63,76,89,93,100–102,111,120,136,146,147,149,154,159,178,197,207,212,228,233,238]	**32** [14,41,47,54,61,71,74,81,97,103,108,115,116,119,132,134,135,142,143,152,157,180,185,187,189,194,217,218,220,223,234,235]	**39** [36,39,45,50,57,69,75,78,90,91,109,113,126,127,130,131,138,140,148,151,171,174,190–192,200,206,210,214,216,219,221,224,225,229–232,236]	**123**
Encoding Transformation	**4** [42,95,117,121]	**5** [51,84,165,205,208]	**6** [63,93,114,128,159,204]	**9** [35,80,94,125,173,188,194,196,209]	**5** [40,72,110,169,219]	**29**
Machine Learning Structure	-	**2** [73,160]	**2** [183,199]	**11** [10,56,66,153,156,158,164,172,179,211,220]	**13** [48,64,85,104,133,137,141,145,170,176,195,226,227]	**28**
Percentile Concept	**1** [53]	**2** [105,112]	**6** [49,62,83,118,124,168]	**3** [99,198,215]	**4** [60,65,79,203]	**16**
Crossover	-	**1** [98]	**2** [136,159]	**2** [38,123]	**4** [7,129,192,219]	**9**

Table 8. Transfer Function per year.

Transfer Function	2017	2018	2019	2020	2021	Total
S-Shaped	**7** [5,31,37,43,44,55,92]	**13** [70,96,106,155,161,162,165,166,177,181,205,222,237]	**20** [2,3,12,63,76,89,93,100,101,111,120,136,146,147,154,159,178,197,233,238]	**26** [14,54,56,61,71,74,81,97,103,115,116,119,132,134,135,142,143,157,180,185,187,189,194,218,220,234]	**32** [39,45,50,69,75,78,85,90,104,109,126,127,130,131,138,140,148,151,170,171,174,190–192,195,214,216,219,221,224,226,227]	**98**
V-Shaped	**6** [31,44,55,59,92,175]	**7** [63,165,177,182,186,213,237]	**12** [12,46,76,102,111,120,122,136,149,207,212,233]	**20** [41,47,54,56,93,103,108,115,116,119,146,152,180,189,194,195,217,218,220,234]	**20** [39,45,50,74,75,78,85,91,104,113,131,140,148,170,171,190,219,224,226,227]	**65**
U-Shaped	-	-	-	**1** [235]	**3** [39,75,171]	**4**
O-Shaped	-	**2** [205,237]	-	-	**2** [39,210]	**4**
Q-Shaped	-	-	**2** [76,120]	-	**2** [200,206]	**4**
S-Shaped Time Varying	**1** [52]	**1** [222]	-	-	**1** [225]	**3**
V-Shaped Time Varying	1 [52]	**1** [222]	-	-	**1** [36]	**3**
X-Shaped	-	-	-	**1** [223]	**1** [236]	**2**
Z-Shaped	-	-	-	**1** [234]	**1** [232]	**2**
Lineal	-	-	-	-	**1** [39]	**1**

4. Discussion

In this section, we will discuss the answers to our research questions.

4.1. What Continuous Metaheuristics Have Been Used from 2017 to Date to Solve Binary Combinatorial Problems?

In general, as mentioned in the previous section, we found a total of 195 out of 512 relevant articles in the sources we searched, a little more than 50% of the articles use continuous metaheuristics to solve binary combinatorial problems, this gives us good indications as they present good indicators that continuous metaheuristics give good results when solving problems of different binary nature. Of the total number of articles, researchers opted for more traditional metaheuristics (PSO, BA, and CS) while the rest decided to use less conventional metaheuristics (see Table 4 or Table 6).

4.2. What Techniques or Forms of Binarization Have Been Used in Metaheuristics from 2017 to Date to Solve Binary Combinatorial Problems?

Analyzing the results obtained, a classification of binarization techniques was made based on how they perform the binarization process to answer this question. They were classified into five categories, which are “Simple Transformation,” “Encoding Transformation,” “Machine Learning Structure,” “Percentile Concept,” and “Crossover.” Table 7 shows the number of times each category was used in each year. Next, we will proceed to define and exemplify each of the five categories mentioned above.

4.2.1. *Simple Transformation*

In the literature, our categorization of Simple Transformation corresponds to a generally sequential mechanism; this transformation works with continuous operators without modifying them. We say it is sequential because the first step fits the introduction of operators that transform the solution of $\mathbf{R}^n$ into $\{InterSpace\}$. For example, in *Great Value Priority*, our interspace is $\mathbf{Z}^n$; in the case of a transfer function, we have $[0,1]^n$ and $\{InterSpace\ Functions\}$ in the Angular Modulation. The second step is to transform from the intermediate space ($\mathbf{Z}^n$, $[0,1]^n$, $\{InterSpace\ Functions\}$) into a binary space $\{0,1\}^n$. See Figure 7 for a better overview of the general scheme. Within our category, we can find several techniques that follow these principles, which we will detail below.

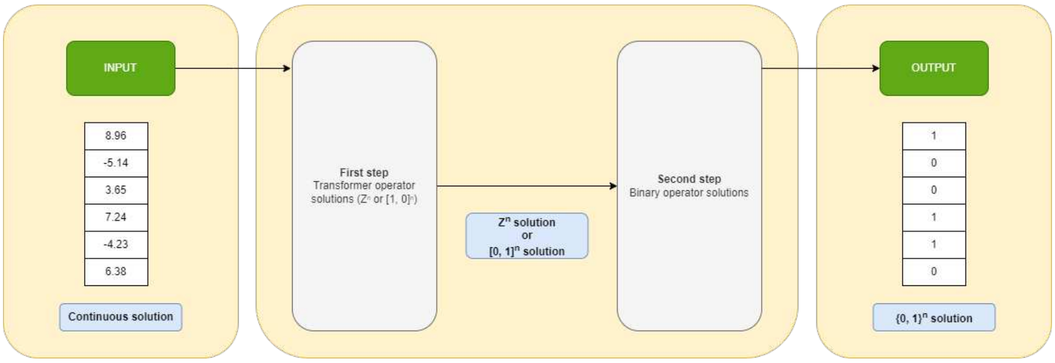

Figure 7. Simple Transformation — Sequential Scheme.

• Transfer Function and Binarization

The first step of this technique corresponds to the use of transfer functions, the most common normalization method and which was introduced in [239], the advantage of the transfer function is that it is a very cheap operator, providing a range of probabilities and attempts to model the transition of the particle position. These functions are responsible for the first step of the binarization method and for mapping the $\mathbf{R}^n$ solutions into $[0,1]^n$.

In our review, we found the use of different transfer functions: the S-shaped [88,163,167,202], V-shaped [68,82,144,201], U-shaped [184], O-shaped [205,237], Q-shaped [76,120,200,206], X-shaped [223,236], Z-shaped [232,234], S-shaped time-varying [52,67,222,225], V-shaped time-varying [36,52,222] and finally the authors in [39] used one called Linear.

The second step is to use a binarization rule, where the particle is transformed into a binary solution. In this review we could identify that a large percentage of the articles used standard (see Equation (1)), however, there are other rules, for example, in [31] four more are mentioned, static probability, elitist, elitist roulette [240,241], and finally complement the technique also used in [46,122]. The four mentioned techniques can be seen in Table 9.

$$Standard\begin{cases} 1 & if\ random \leq T(d_w^j) \\ 0 & otherwise. \end{cases} \tag{1}$$

Table 9. Binarization Functions.

Type	Binarization
Complement	$b_{new}^j = \begin{cases} complement(b_w^j) & if\ rand \leq T(d_w^j) \\ 0 & otherwise. \end{cases}$
Static Probability	$b_{new}^j = \begin{cases} 0 & if\ T(d_w^j) \leq \alpha \\ b_w^j & if\ \alpha < T(d_w^j) \leq \frac{1}{2}(1+\alpha) \\ 1 & if\ T(d_w^j) \geq \frac{1}{2}(1+\alpha) \end{cases}$
Elitist	$b_{new}^j = \begin{cases} b_{Best}^j & if\ rand < T(d_w^j) \\ 0 & else. \end{cases}$
Elitist Roulette	$b_{new}^j = \begin{cases} P[b_{new}^j = \zeta_j] = \frac{f(\zeta)}{\sum_{\delta \in Q_g} f(\delta)} & if\, rand \leq T(d_w^j) \\ P[b_{new}^j = 0] = 1 & else. \end{cases}$

- **Angle Modulation**

This approach is used in the telecommunications industry for carrier modulation of the [242] signal and uses trigonometric functions with four parameters capable of controlling the frequency and offset of the function itself.

$$g_i(x_j) = sin(2\pi(x_j - a_i)b_i \cdot cos(2\pi(x_j - a_i)c_i)) + d_i \tag{2}$$

Let us consider $X = \{x_1, x_2, ..., x_n\}$ a solution of an $n-$dimensional binary problem. We will start with a 4-dimensional search space, where each of these dimensions represents a coefficient of Equation (2). Considering this search space, the first step is to obtain a function in a function space, namely from each solution represented in the space as (a_i, b_i, c_i, d_i). Then we obtain a trigonometric function g_i that lies in a function space.

For the second step, for each of the x_j elements present in X, we apply a binarization rule (see Equation (3)) and obtain an $n-$dimensional binary solution.

$$b_{i,j} = \begin{cases} 1 & if\ g_i(x_j) \geq 0 \\ 0 & otherwise \end{cases} \tag{3}$$

In this way, for each initial solution, remembering (a_i, b_i, c_i, d_i), we will obtain an $n-$dimensional binary solution: $(b_{i,1}, b_{i,2}, ..., b_{i,n})$. In this review, we were able to find some papers that applied this technique, such as [228–231].

- **Binarization Conversion**

There are some works where the authors did not apply any traditional way as mentioned above. Failing that, these authors in their research indicate that they obtain the value of their particle and then randomly apply binarization. In [57,150] they update the population with the following Eq. $x_{tk}^0 = round(rand(0,1))$ and in [77,139] they use Equation (4).

$$(x_i^t)_{bin} = \begin{cases} 1 & if\ x_i^t > 0.5 \\ 0 & Otherwise \end{cases} \tag{4}$$

4.2.2. Encoding Transformation

Unlike the previous category, these methods classified as encoding are characterized by redefining the metaheuristic operators; we can identify two main groups. The first are those that modify the operations of the search space through modified algebraic or logical operations. The second group can be referred to as promising regions, and the operators, this time, are restructured based on the promising regions that were selected in the search space. The Quantum binary approach is an example of this group we found in the analyzed articles.

- **Algebraic or Logic Operations**

This method transforms real operators into binary or logical operators. This transformation is performed by Boolean operations, which causes the operators to be able to act on the binary solutions. This approach was proposed as a binarization technique in PSO [243].

The Boolean notation or also known as logic gates are as follows: "XOR" = $\bigotimes$, "AND" = $\bigoplus$ and "OR" = $\bigodot$. Equations (5) and (6) present the Boolean equations for velocity and position.

$$V_i(t+1) = w \oplus V_i(t) \odot c_1 \otimes (P_{best,i} \oplus X_i) \odot c_2 \otimes (P_{global} \oplus X_i) \tag{5}$$

$$X_i(t+1) = X_i(t) \oplus V_i(t+1) \tag{6}$$

where $V_i(t)$ and X_i are the representation of the velocity at time t and the particle position, c_1 and c_2 are random vectors. $P_{best,i}$ is the best position selected by the particle and P_{global} corresponds to the position of the best global solution.

This method has been used in the application in different binary optimization problems using GSK [107], PSO [40,86], BSO [84], TGA [173], BA [40,63], GWO [63], GSA [63], ABC [159], DA [40], SMO [114], BHO [188] and CS [196].

- **Quantum Binary Approach**

This approach is inspired by the uncertainty principle, where we cannot simultaneously determine position and velocity. Therefore, the PSO algorithm works differently for individual particles, and we need to rewrite the operators. In the quantum approach, each feasible solution has a position $X = \{x_1, x_2, ..., x_n\}$ and a quantum vector $Q = [Q_1, Q_2, ..., Q_n]$. The quantum vector Q_j represents the probability that x_j takes the value 1. For each dimension j, a random number between $[0,1]$ is generated and compared to Q_j (see Equation (7)).

$$x_j = \begin{cases} 1 & if\quad rand < Q_j \\ 0 & \quad otherwise \end{cases} \tag{7}$$

After obtaining the binary values of the solutions, the new P_{best} and P_{global} are calculated using the objective function. Finally, the transition probability is updated using Equation (8).

$$Q(t+1) = C_1 \cdot Q(t) + C_2 \cdot Q_{self}(t) + C_3 \cdot Q_{global}(t) \tag{8}$$

where Q_{self} and Q_{global} are calculated with Equations (9) and (10) respectively.

$$Q_{self}(t) = \alpha P_{best}(t) + \beta(1 - P_{best}) \tag{9}$$

$$Q_{social}(t) = \alpha P_{global}(t) + \beta(1 - P_{global}) \tag{10}$$

The quantum method has been applied to various problems such as Multidimensional Knapsack Problem with PSO [209], Next Release Problem con AAA [72], Feature Selection con OSA [80], GSA [42]. In [208], they apply this method to binarize PSO, while in [204] apart from PSO, also in ABC, AFSA.

4.2.3. Percentile Concept

This approach using percentile operators works directly with the own parameters of each metaheuristic $\Delta(x)$ (see Equation (11)), i.e., the binary percentile operator considers the displacements generated by the metaheuristic in each dimension for each of the particles.

$$x(t+1) = x(t) + \Delta_{t+1}(x(t)) \tag{11}$$

The parameter $\Delta^i(x)$ is the magnitude of the displacement $\Delta(x)$ at the i-th position for the particle x; subsequently, these displacements are grouped using the magnitude of the displacement ($\Delta^i(x)$) and an assigned percentile list. The binary percentile operator will have as input the parameters of the percentile list; suppose $Pr = \{20, 40, 60, 80, 100\}$ and a list of values. Given an iteration, the list of values corresponds to the magnitude of the displacement $\Delta^i(x)$ of the particles in each dimension. As a first step, the percentile operator uses the value list and obtains the given percentile values in Pr. Subsequently, each value in the value list is assigned the smallest percentile group to which it belongs. Finally, the list of the percentile to which each value belongs is obtained and assigned a transition probability by Equation (12).

$$P_{tr} = \begin{cases} 0.1 & if \quad x^i \in group\{0,1\} \\ 0.5 & if \quad x^i \in group\{2,3,4\} \end{cases} \tag{12}$$

The concept of percentiles has been applied to the classical Knapsack Problem and the Set Covering Problem in several articles with binary versions of ALO [118,168], BA [99,215], FA [65,79], GOA [62,83], SCA [49], WOA [60,203], MVO [53,124], GSA [198]. CS has also been applied in the Multidimensional Knapsack Problem [112] and in the Set Covering Problem [105].

4.2.4. Machine Learning Structure

This approach uses the great virtues of machine learning techniques to binarize. Machine Learning techniques have taken great prominence in recent years thanks to great technological advances, which allow us to have a great computational capacity. Machine learning techniques are classified into three groups, Supervised Learning, Unsupervised Learning, and Reinforcement Learning [244–246]. In the present analysis, the use of algorithms belonging to the groups of Unsupervised Learning and Reinforcement Learning was detected.

- **Unsupervised Learning**

In unsupervised learning, the output data is unknown from the beginning. The learning lies in recognizing patterns in the input data. This type of learning is also known as clustering learning.

The algorithms used to binarize by clustering are K-Means [64,133,137,141] and Db-Scan [66,179,183,211]. Were applied to binarize CS, PSO, BA, BHO, GOA, FA, WOA, CSA, GSO para resolver Buttressed Walls Problem, Multidimensional Knapsack Problem, Set Covering Problem, Crew Scheduling Problem, and Set Union Knapsack Problem.

- **Reinforcement Learning**

Reinforcement Learning consists of finding the best action to be taken by an agent in a given state. The performance of the action is judged through a reward. Therefore, Reinforcement Learning algorithms seek to maximize this reward accumulated over time.

In Section 4.2.1 we mentioned two-step techniques, a technique that combines a transfer function and a binarization function to binarize continuous solutions. To find the best combination of functions it is necessary to test them all, which is very time-consuming. To solve this, reinforced learning techniques are incorporated.

In [85,104,170,195] they use Q-Learning and in [85,227,247] they use SARSA as intelligent selectors of binarization schemes. These two techniques were applied to binarize GWO, HHO, SCA, and WOA to solve Set Covering Problem.

4.2.5. Crossover

Within the population metaheuristics, we have a group called Evolutionary Algorithms [248], which uses evolutionary theory to perturb the solutions. The operations used to perturb the solutions are crossover and mutation, where the most common algorithms within this group of metaheuristics are Genetic Algorithm (GA) and Differential Evolution (DE). Different proposals such as [7,88,129,219] use a continuous metaheuristic hybridized with an evolutionary algorithm to perturb the solutions.

The optimization process consists of initializing the solutions in a binary way to the metaheuristic, using its movement patterns in conjunction with the evolutionary algorithms to obtain binary perturbed solutions.

4.3. Closing of Discussions

Because of what has been presented and summarizing the answers proposed in Sections 4.1 and 4.2, it is clarified that the most used metaheuristics are those based on the swarm, since these have recently gained remarkable popularity, both for their good performance and their ease of implementation, both in continuous and discrete domains using the aforementioned techniques. Along with this, it is emphasized what was already indicated in the predecessor of this work, where the techniques or forms of binarization where the most used are the Simple Transformation or in other words, those that are sequential. But together with ratifying what was previously investigated, new alternatives of binarization are presented, such as Percentile Concept or Machine Learning Structure there are also novel transfer functions or hybridizations called in this work as crossovers, which present new and novel techniques or forms of binarization, enriching the options to be used to be able to operate with continuous metaheuristics in binary space.

5. Conclusions

This study investigated and compiled important binarization methods of continuous metaheuristics. Within our obtained conglomerate of binarization, we propose five main classifications; in Table 7, these groups are shown. The first group we call Simple Transformation sequential binarization mechanisms, which use an intermediate space from which the binarization is mapped. The second group we call Encoding Transformation, where the metaheuristic operator is adapted to a binary problem; by analyzing this adaptation of operators, we could find methods that transform algebraic operations and methods that use a probability to make the transition in the search space. The third group is called Percentile Concept and is based on statistics using percentile groups to the values obtained in the solutions. The fourth group was assigned the name Machine Learning Structure and, as its name indicates, incorporates machine learning techniques for binarization. In the review, two different groups were detected, Unsupervised Learning and Reinforcement Learning. On the Unsupervised Learning side, the authors used clustering techniques such as K-Means and DbScan to binarize. On the Reinforcement Learning side, the authors used Q-Learning and SARSA as intelligent selectors of binarization schemes coming from the two-step technique. The fifth group we call Crossover because continuous metaheuristics are hybridized with evolutionary algorithms to use operators such as Crossover to obtain perturbed binarization solutions.

In addition, we investigated which specific metaheuristics use these binarization techniques; the summary is shown in Table 6. Based on the study performed by all researchers, we can conclude that the most used method to binarize continuous metaheuristics is the transfer function, belonging to the Simple Transformation; a summary of the most used functions can be seen in Table 8. In addition, we looked for what the different techniques solved optimization problems; see Table 5.

This continuation of systematic study in this area is to find new forms of binarization and to understand how continuous metaheuristics' exploration and exploitation properties map to binary metaheuristics. This is of public utility since knowing this information gives way to improve results in metaheuristics and broaden the spectrum of binary problems to be solved. This survey of the compilation of binarization techniques leads to the conclusion that there is no general technique capable of binarizing efficiently, placing it above the others, which is on a par with the No Free Lunch theorem [20], which indicates the same principle.

Author Contributions: Conceptualization, M.B.-R. and B.C.; methodology, M.B.-R. and F.C.-C.; software, M.B.-R. and F.C.-C.; validation, M.B.-R., B.C., R.S., G.A., C.C. and J.G.; formal analysis, M.B.-R.; investigation, M.B.-R. and F.C.-C.; resources, M.B.-R. and F.C.-C.; data curation, M.B.-R., F.C.-C. and J.G.; writing—original draft preparation, M.B.-R., F.C.-C. and J.G.; writing—review and editing, M.B.-R., F.C.-C., J.G., J.L.-R., G.A. and C.C.; supervision, B.C., R.S. and J.L.-R.; project administration, M.B.-R. and B.C; funding acquisition, B.C. and C.C. All authors have read and agreed to the published version of the manuscript.

Funding: Broderick Crawford and Ricardo Soto are supported by Grant ANID/FONDECYT/REGULAR/1210810. Marcelo Becerra-Rozas is supported by the National Agency for Research and Development (ANID)/Scholarship Program/DOCTORADO NACIONAL/2021-21210740. Felipe Cisternas-Caneo is supported by Beca INF-PUCV.

Data Availability Statement: The study does not provide any data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Alizadeh, R.; Nishi, T. Hybrid Set Covering and Dynamic Modular Covering Location Problem: Application to an Emergency Humanitarian Logistics Problem. *Appl. Sci.* **2020**, *10*, 7110. [CrossRef]
2. Soto, R.; Crawford, B.; Aste Toledo, A.; Castro, C.; Paredes, F.; Olivares, R. Solving the manufacturing cell design problem through binary cat swarm optimization with dynamic mixture ratios. *Comput. Intell. Neurosci.* **2019**, *2019*, 4787856. [CrossRef] [PubMed]
3. Aly, R.H.M.; Rahouma, K.H.; Hamed, H.F. Brain Tumors Diagnosis and Prediction Based on Applying the Learning Metaheuristic Optimization Techniques of Particle Swarm, Ant Colony and Bee Colony. *Procedia Comput. Sci.* **2019**, *163*, 165–179. [CrossRef]
4. Liu, Z.; Xu, H.; Liu, P.; Li, L.; Zhao, X. Interval-valued intuitionistic uncertain linguistic multi-attribute decision-making method for plant location selection with partitioned hamy mean. *Int. J. Fuzzy Syst.* **2020**, *22*, 1993–2010. [CrossRef]
5. Dahi, Z.A.E.M.; Mezioud, C.; Draa, A. A 0-1 bat algorithm for cellular network optimisation: A systematic study on mapping techniques. *Int. J. Reason.-Based Intell. Syst.* **2017**, *9*, 22–42. [CrossRef]
6. Mandal, S.; Patra, N.; Pal, M. Covering problem on fuzzy graphs and its application in disaster management system. *Soft Comput.* **2021**, *25*, 2545–2557. [CrossRef]
7. Houssein, E.H.; Ibrahim, I.E.; Neggaz, N.; Hassaballah, M.; Wazery, Y.M. An efficient ECG arrhythmia classification method based on Manta ray foraging optimization. *Expert Syst. Appl.* **2021**, *181*, 115131. [CrossRef]
8. Xiang, X.; Qiu, J.; Xiao, J.; Zhang, X. Demand coverage diversity based ant colony optimization for dynamic vehicle routing problems. *Eng. Appl. Artif. Intell.* **2020**, *91*, 103582. [CrossRef]
9. Nucamendi-Guillén, S.; Dávila, D.; Camacho-Vallejo, J.F.; González-Ramírez, R.G. A discrete bilevel brain storm algorithm for solving a sales territory design problem: A case study. *Memetic Comput.* **2018**, *10*, 441–458. [CrossRef]
10. García, J.; Yepes, V.; Martí, J.V. A hybrid k-means cuckoo search algorithm applied to the counterfort retaining walls problem. *Mathematics* **2020**, *8*, 555. [CrossRef]
11. Calvete, H.I.; Galé, C.; Iranzo, J.A.; Camacho-Vallejo, J.F.; Casas-Ramírez, M.S. A matheuristic for solving the bilevel approach of the facility location problem with cardinality constraints and preferences. *Comput. Oper. Res.* **2020**, *124*, 105066. [CrossRef]
12. Bir-Jmel, A.; Douiri, S.M.; Elbernoussi, S. Gene selection via BPSO and Backward generation for cancer classification. *RAIRO-Oper. Res.* **2019**, *53*, 269–288. [CrossRef]
13. Camacho-Vallejo, J.F.; Garcia-Reyes, C. Co-evolutionary algorithms to solve hierarchized Steiner tree problems in telecommunication networks. *Appl. Soft Comput.* **2019**, *84*, 105718. [CrossRef]
14. Zhao, T.F.; Chen, W.N.; Wu, X.K.; Yang, L.; Yang, Q. Discrete Resource Allocation in Epidemic Control with Heuristic Majority-Voting Particle Swarm Optimization. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 11–14 October 2020; pp. 2058–2065.
15. Mitchell, J.E. Branch-and-cut algorithms for combinatorial optimization problems. *Handb. Appl. Optim.* **2002**, *1*, 65–77.

16. De Farias, I.; Johnson, E.L.; Nemhauser, G.L. Branch-and-cut for combinatorial optimization problems without auxiliary binary variables. *Knowl. Eng. Rev.* **2001**, *16*, 25–39. [CrossRef]
17. Little, J.D. Branch and Bound Methods for Combinatorial Problems. In Proceedings of the TAPPI Operations Research Symposium, Philadelphia, PA, USA, 28–30 March 1966.
18. Brusco, M.J.; Stahl, S.; *Branch-and-Bound Applications in Combinatorial Data Analysis*; Springer: Berlin/Heidelberg, Germany, 2005; Volume 2.
19. Coniglio, S.; Furini, F.; San Segundo, P. A new combinatorial branch-and-bound algorithm for the knapsack problem with conflicts. *Eur. J. Oper. Res.* **2021**, *289*, 435–455. [CrossRef]
20. Ho, Y.C.; Pepyne, D.L. Simple explanation of the no-free-lunch theorem and its implications. *J. Optim. Theory Appl.* **2002**, *115*, 549–570. [CrossRef]
21. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. [CrossRef]
22. Dorigo, M.; Birattari, M.; Stutzle, T. Ant colony optimization. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–39. . MCI.2006.329691. [CrossRef]
23. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [CrossRef]
24. Yang, X.S. A New Metaheuristic Bat-Inspired Algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*; González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 65–74. ._6. [CrossRef]
25. Yang, X.S.; Deb, S. Cuckoo Search via Lévy flights. In Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009; pp. 210–214. [CrossRef]
26. Yang, X.S. Firefly Algorithms for Multimodal Optimization. In Proceedings of the Stochastic Algorithms: Foundations and Applications, Sapporo, Japan, 26–28 October 2009; Watanabe, O., Zeugmann, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 169–178. ._14. [CrossRef]
27. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. . 2013.12.007. [CrossRef]
28. Mirjalili, S. SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [CrossRef]
29. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. . 2016.01.008. [CrossRef]
30. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **2016**, *27*, 1053–1073. [CrossRef]
31. Crawford, B.; Soto, R.; Astorga, G.; García, J.; Castro, C.; Paredes, F. Putting continuous metaheuristics to work in binary search spaces. *Complexity* **2017**, *2017*, 8404231. [CrossRef]
32. Kitchenham, B. Procedures for performing systematic reviews. *Keele UK Keele Univ.* **2004**, *33*, 1–26.
33. Grivel, L.; Mutschke, P.; Polanco, X. Thematic mapping on bibliographic databases by cluster analysis: A description of the sdoc environment with solis. *J. Knowl. Organ.* **1995**, *22*, 70–77.
34. López-Fernández, M.C.; Serrano-Bedia, A.M.; Pérez-Pérez, M. Entrepreneurship and family firm research: A bibliometric analysis of an emerging field. *J. Small Bus. Manag.* **2016**, *54*, 622–639. [CrossRef]
35. Baykasoğlu, A.; Ozsoydan, F.B.; Senol, M.E. Weighted superposition attraction algorithm for binary optimization problems. *Oper. Res.* **2020**, *20*, 2555–2581. [CrossRef]
36. Chantar, H.; Thaher, T.; Turabieh, H.; Mafarja, M.; Sheta, A. BHHO-TVS: A binary harris hawks optimizer with time-varying scheme for solving data classification problems. *Appl. Sci.* **2021**, *11*, 6516. [CrossRef]
37. Nguyen, P.H.; Wang, D.; Truong, T.K. A novel binary social spider algorithm for 0–1 knapsack problem. *Int. J. Innov. Comput. Inf. Control* **2017**, *13*, 2039–2049.
38. Guo, W.; Liu, T.; Dai, F.; Xu, P. An improved whale optimization algorithm for feature selection. *Comput. Mater. Contin.* **2020**, *62*, 337–354. [CrossRef]
39. Beheshti, Z. UTF: Upgrade transfer function for binary meta-heuristic algorithms. *Appl. Soft Comput.* **2021**, *106*, 107346. [CrossRef]
40. Merikhi, B.; Soleymani, M. Automatic data clustering framework using nature-inspired binary optimization algorithms. *IEEE Access* **2021**, *9*, 93703–93722. [CrossRef]
41. Zhao, S.; Yang, Z.; Zhu, X.; Wang, Y. A Novel Binary Social Learning Particle Swarm Optimizer for Power System Unit Commitment. In Proceedings of the 2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2), Wuhan, China, 30 October–1 November 2020; pp. 1361–1365.
42. Barani, F.; Mirhosseini, M.; Nezamabadi-Pour, H. Application of binary quantum-inspired gravitational search algorithm in feature subset selection. *Appl. Intell.* **2017**, *47*, 304–318. [CrossRef]
43. Wang, W.; Li, C.; Liao, X.; Qin, H. Study on unit commitment problem considering pumped storage and renewable energy via a novel binary artificial sheep algorithm. *Appl. Energy* **2017**, *187*, 612–626. [CrossRef]
44. Lanza-Gutierrez, J.M.; Crawford, B.; Soto, R.; Berrios, N.; Gomez-Pulido, J.A.; Paredes, F. Analyzing the effects of binarization techniques when solving the set covering problem through swarm optimization. *Expert Syst. Appl.* **2017**, *70*, 67–82. [CrossRef]
45. Kumar, V.; Kumar, D.; Kaur, M.; Singh, D.; Idris, S.A.; Alshazly, H. A novel binary seagull optimizer and its application to feature selection problem. *IEEE Access* **2021**, *9*, 103481–103496. [CrossRef]

46. Pereira, L.A.; Papa, J.P.; Coelho, A.L.; Lima, C.A.; Pereira, D.R.; de Albuquerque, V.H.C. Automatic identification of epileptic EEG signals through binary magnetic optimization algorithms. *Neural Comput. Appl.* **2019**, *31*, 1317–1329. [CrossRef]
47. Jaafer, A.A.; Al-Bazoon, M.; Dawood, A.O. Structural topology design optimization using the binary bat algorithm. *Appl. Sci.* **2020**, *10*, 1481. [CrossRef]
48. García, J.; Astorga, G.; Yepes, V. An analysis of a KNN perturbation operator: An application to the binarization of continuous metaheuristics. *Mathematics* **2021**, *9*, 225. [CrossRef]
49. Pinto, H.; Peña, A.; Valenzuela, M.; Fernández, A. A binary sine-cosine algorithm applied to the knapsack problem. In Proceedings of the Computer Science On-line Conference, Zlin, Czech Republic, 24–27 April 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 128–138.
50. Dhiman, G.; Oliva, D.; Kaur, A.; Singh, K.K.; Vimal, S.; Sharma, A.; Cengiz, K. BEPO: A novel binary emperor penguin optimizer for automatic feature selection. *Knowl.-Based Syst.* **2021**, *211*, 106560. [CrossRef]
51. Atta, S.; Mahapatra, P.R.S.; Mukhopadhyay, A. Solving uncapacitated facility location problem using monkey algorithm. In *Intelligent Engineering Informatics*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 71–78.
52. Islam, M.J.; Li, X.; Mei, Y. A time-varying transfer function for balancing the exploration and exploitation ability of a binary PSO. *Appl. Soft Comput.* **2017**, *59*, 182–196. [CrossRef]
53. Valenzuela, M.; Peña, A.; Lopez, L.; Pinto, H. A binary multi-verse optimizer algorithm applied to the set covering problem. In Proceedings of the 2017 4th International Conference on Systems and Informatics (ICSAI), Hangzhou, China, 11–13 November 2017; pp. 513–518.
54. Hans, R.; Kaur, H. Binary Multi-Verse Optimization (BMVO) Approaches for Feature Selection. *Int. J. Interact. Multimed. Artif. Intell.* **2020**, *6*, 91. [CrossRef]
55. Soto, R.; Crawford, B.; Olivares, R.; Barraza, J.; Figueroa, I.; Johnson, F.; Paredes, F.; Olguín, E. Solving the non-unicost set covering problem by using cuckoo search and black hole optimization. *Nat. Comput.* **2017**, *16*, 213–229. [CrossRef]
56. Tapia, D.; Crawford, B.; Soto, R.; Cisternas-Caneo, F.; Lemus-Romani, J.; Castillo, M.; García, J.; Palma, W.; Paredes, F.; Misra, S. A q-learning hyperheuristic binarization framework to balance exploration and exploitation. In Proceedings of the International Conference on Applied Informatics, Eger, Hungary, 29–31 January 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 14–28.
57. Agrawal, P.; Ganesh, T.; Mohamed, A.W. A novel binary gaining–sharing knowledge-based optimization algorithm for feature selection. *Neural Comput. Appl.* **2021**, *33*, 5989–6008. [CrossRef]
58. Elmanakhly, D.A.; Saleh, M.; Rashed, E.A.; Abdel-Basset, M. BinHOA: Efficient Binary Horse Herd Optimization Method for Feature Selection: Analysis and Validations. *IEEE Access* **2022**, *10*, 26795–26816. [CrossRef]
59. Almonacid, B.; Aspée, F.; Soto, R.; Crawford, B.; Lama, J. Solving the manufacturing cell design problem using the modified binary firefly algorithm and the egyptian vulture optimisation algorithm. *IET Softw.* **2017**, *11*, 105–115. [CrossRef]
60. Jorquera, L.; Valenzuela, P.; Causa, L.; Moraga, P.; Rubio, J.M. A Percentile Whale Algorithm an Application to the Set Covering Problem. In Proceedings of the Computational Methods in Systems and Software, Online, Czech Republic, 1 October 2021; pp. 402–412.
61. Crawford, B.; Soto, R.; Olivares, R.; Embry, G.; Flores, D.; Palma, W.; Castro, C.; Paredes, F.; Rubio, J.M. A binary monkey search algorithm variation for solving the set covering problem. *Nat. Comput.* **2020**, *19*, 825–841. [CrossRef]
62. Pinto, H.; Peña, A.; Valenzuela, M.; Fernández, A. A binary grasshopper algorithm applied to the knapsack problem. In *Proceedings of the Computer Science On-line Conference*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 132–143.
63. Taghian, S.; Nadimi-Shahraki, M.H.; Zamani, H. Comparative analysis of transfer function-based binary Metaheuristic algorithms for feature selection. In Proceedings of the 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 28–30 September 2018; pp. 1–6.
64. García, J.; Lemus-Romani, J.; Altimiras, F.; Crawford, B.; Soto, R.; Becerra-Rozas, M.; Moraga, P.; Becerra, A.P.; Fritz, A.P.; Rubio, J.M.; et al. A binary machine learning cuckoo search algorithm improved by a local search operator for the set-union knapsack problem. *Mathematics* **2021**, *9*, 2611. [CrossRef]
65. Pinto, H.; Valenzuela, M.; Maureira, C.; Lopez, L.; Fernández, A. A Binary Firefly Algorithm Applied to Knapsack Problem. In Proceedings of the Computer Science On-line Conference, Online, 1 April 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 376–385.
66. Vásquez, C.; Lemus-Romani, J.; Crawford, B.; Soto, R.; Astorga, G.; Palma, W.; Misra, S.; Paredes, F. Solving the 0/1 knapsack problem using a galactic swarm optimization with data-driven binarization approaches. In Proceedings of the International Conference on Computational Science and Its Applications, Cagliari, Italy, 1–4 July 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 511–526.
67. Mohd Yusof, N.; Muda, A.K.; Pratama, S.F.; Abraham, A. A novel nonlinear time-varying sigmoid transfer function in binary whale optimization algorithm for descriptors selection in drug classification. *Mol. Divers.* **2022**, 1–10. [CrossRef]
68. Kalra, M.; Kumar, V.; Kaur, M.; Idris, S.A.; Öztürk, S.; Alshazly, H. A Novel Binary Emperor Penguin Optimizer for Feature Selection Tasks. *CMC-Comput. Mater. Contin.* **2022**, *70*, 6239–6255. [CrossRef]
69. Latha, M.; Kavitha, G. Combined Metaheuristic Algorithm and Radiomics Strategy for the Analysis of Neuroanatomical Structures in Schizophrenia and Schizoaffective Disorders. *IRBM* **2021**, *42*, 353–368. [CrossRef]
70. Sanchez, I.A.L.; Vargas, J.M.; Santos, C.A.; Mendoza, M.G.; Moctezuma, C.J.M. Solving binary cutting stock with matheuristics using particle swarm optimization and simulated annealing. *Soft Comput.* **2018**, *22*, 6111–6119. [CrossRef]

71. Phogat, M.; Kumar, D. Classification of complex diseases using an improved binary cuckoo search and conditional mutual information maximization. *Computación y Sist.* **2020**, *24*, 1121–1129. [CrossRef]
72. Pirozmand, P.; Ebrahimnejad, A.; Alrezaamiri, H.; Motameni, H. A novel approach for the next software release using a binary artificial algae algorithm. *J. Intell. Fuzzy Syst.* **2021**, *40*, 5027–5041. [CrossRef]
73. García, J.; Altimiras, F.; Peña, A.; Astorga, G.; Peredo, O. A binary cuckoo search big data algorithm applied to large-scale crew scheduling problems. *Complexity* **2018**, *2018*, 8395193. [CrossRef]
74. Siqueira, H.; Santana, C.; Macedo, M.; Figueiredo, E.; Gokhale, A.; Bastos-Filho, C. Simplified binary cat swarm optimization. *Integr. Comput.-Aided Eng.* **2021**, *28*, 35–50. [CrossRef]
75. Abdel-Basset, M.; Mohamed, R.; Sallam, K.M.; Chakrabortty, R.K.; Ryan, M.J. BSMA: A novel metaheuristic algorithm for multi-dimensional knapsack problems: Method and comprehensive analysis. *Comput. Ind. Eng.* **2021**, *159*, 107469. [CrossRef]
76. Too, J.; Abdullah, A.R.; Mohd Saad, N. A new quadratic binary harris hawk optimization for feature selection. *Electronics* **2019**, *8*, 1130. [CrossRef]
77. Jin, Z.; Li, N. Diagnosis of each main coronary artery stenosis based on whale optimization algorithm and stacking model. *Math. Biosci. Eng.* **2022**, *19*, 4568–4591. [CrossRef]
78. Ghosh, K.K.; Guha, R.; Bera, S.K.; Kumar, N.; Sarkar, R. S-shaped versus V-shaped transfer functions for binary Manta ray foraging optimization in feature selection problem. *Neural Comput. Appl.* **2021**, *33*, 11027–11041. [CrossRef]
79. Jorquera, L.; Valenzuela, P.; Causa, L.; Moraga, P.; Villavicencio, G. A Percentile Firefly Algorithm an Application to the Set Covering Problem. In Proceedings of the Computer Science On-line Conference, Online, 1 April 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 750–759.
80. Mandal, A.K.; Sen, R.; Goswami, S.; Chakrabarti, A.; Chakraborty, B. A New Approach for Feature Subset Selection using Quantum Inspired Owl Search Algorithm. In Proceedings of the 2020 10th International Conference on Information Science and Technology (ICIST), Kopaonik, Serbia, 8–11 March 2020; pp. 266–273.
81. Laabadi, S.; Naimi, M.; El Amri, H.; Achchab, B. A binary crow search algorithm for solving two-dimensional bin packing problem with fixed orientation. *Procedia Comput. Sci.* **2020**, *167*, 809–818. [CrossRef]
82. Agrawal, P.; Ganesh, T.; Oliva, D.; Mohamed, A.W. S-shaped and v-shaped gaining-sharing knowledge-based algorithm for feature selection. *Appl. Intell.* **2022**, *52*, 81–112. [CrossRef]
83. Crawford, B.; Soto, R.; Peña, A.; Astorga, G. A binary grasshopper optimisation algorithm applied to the set covering problem. In Proceedings of the Computer Science On-line Conference, Vsetin, Czech Republic, 25–28 April 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 1–12.
84. Xiong, G.; Shi, D.; Zhang, J.; Zhang, Y. A binary coded brain storm optimization for fault section diagnosis of power systems. *Electr. Power Syst. Res.* **2018**, *163*, 441–451. [CrossRef]
85. Becerra-Rozas, M.; Lemus-Romani, J.; Crawford, B.; Soto, R.; Cisternas-Caneo, F.; Embry, A.T.; Molina, M.A.; Tapia, D.; Castillo, M.; Misra, S.; et al. Reinforcement Learning Based Whale Optimizer. In Proceedings of the International Conference on Computational Science and Its Applications, Cagliari, Italy, 13–16 September 2021; pp. 205–219.
86. Thaher, T.; Chantar, H.; Too, J.; Mafarja, M.; Turabieh, H.; Houssein, E.H. Boolean Particle Swarm Optimization with various Evolutionary Population Dynamics approaches for feature selection problems. *Expert Syst. Appl.* **2022**, *195*, 116550. [CrossRef]
87. Abd Elaziz, M.; Yang, H.; Lu, S. A multi-leader Harris hawk optimization based on differential evolution for feature selection and prediction influenza viruses H1N1. *Artif. Intell. Rev.* **2022**, *55*, 2675–2732. [CrossRef]
88. Abdel-Basset, M.; Mohamed, R.; Elkomy, O.M.; Abouhawwash, M. Recent metaheuristic algorithms with genetic operators for high-dimensional knapsack instances: A comparative study. *Comput. Ind. Eng.* **2022**, *166*, 107974. [CrossRef]
89. Wang, X.; Sun, Y.; Gu, H. BMM: A binary metaheuristic mapping algorithm for mesh-based network-on-chip. *IEICE Trans. Inf. Syst.* **2019**, *102*, 628–631. [CrossRef]
90. Mohammadzadeh, H.; Gharehchopogh, F.S. A multi-agent system based for solving high-dimensional optimization problems: A case study on email spam detection. *Int. J. Commun. Syst.* **2021**, *34*, e4670. [CrossRef]
91. Pashaei, E.; Pashaei, E. Gene selection using hybrid dragonfly black hole algorithm: A case study on RNA-seq COVID-19 data. *Anal. Biochem.* **2021**, *627*, 114242. [CrossRef]
92. Tilahun, S.L.; Ngnotchouye, J.M.T. Firefly algorithm for discrete optimization problems: A survey. *KSCE J. Civ. Eng.* **2017**, *21*, 535–545. [CrossRef]
93. Kumar, V.; Kumar, D. Binary whale optimization algorithm and its application to unit commitment problem. *Neural Comput. Appl.* **2020**, *32*, 2095–2123. [CrossRef]
94. Bolaji, A.L.; Okwonu, F.Z.; Shola, P.B.; Balogun, B.S.; Adubisi, O.D. A modified binary pigeon-inspired algorithm for solving the multi-dimensional knapsack problem. *J. Intell. Syst.* **2021**, *30*, 90–103. [CrossRef]
95. Li, J.; Fong, S.; Wong, R.K.; Millham, R.; Wong, K.K. Elitist binary wolf search algorithm for heuristic feature selection in high-dimensional bioinformatics datasets. *Sci. Rep.* **2017**, *7*, 43544. [CrossRef]
96. ur Rehman, A.; Bermak, A. Recursive DBPSO for computationally efficient electronic nose system. *IEEE Sensors J.* **2017**, *18*, 320–327. [CrossRef]
97. Reddy K, S.; Panwar, L.; Panigrahi, B.; Kumar, R. Binary whale optimization algorithm: A new metaheuristic approach for profit-based unit commitment problems in competitive electricity markets. *Eng. Optim.* **2019**, *51*, 369–389. [CrossRef]

98. Cao, J.; Yin, B.; Lu, X.; Kang, Y.; Chen, X. A modified artificial bee colony approach for the 0-1 knapsack problem. *Appl. Intell.* **2018**, *48*, 1582–1595. [CrossRef]
99. Jorquera, L.; Villavicencio, G.; Causa, L.; Lopez, L.; Fernández, A. A Binary Bat Algorithm Applied to Knapsack Problem. In Proceedings of the Computer Science On-line Conference, Zlin, Czech Republic, 15 July 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 172–182.
100. El-Dakroury, H.E.D.M.; Gad, A.; Abdelaziz, A.Y. Load Restoration in Primary Distribution Networks Using the Binary Particle Swarm Optimization. In Proceedings of the 2019 IEEE Electrical Power and Energy Conference (EPEC), Ottawa, ON, Canada, 12–14 October 2016; pp. 1–6.
101. Han, C.; Zhou, G.; Zhou, Y. Binary symbiotic organism search algorithm for feature selection and analysis. *IEEE Access* **2019**, *7*, 166833–166859. [CrossRef]
102. Guendouzi, W.; Boukra, A. A Manhattan distance-based binary bat algorithm vs. integer ant colony optimisation for intrusion detection in the audit trails. *Int. J. Comput. Sci. Eng.* **2019**, *18*, 424–437. [CrossRef]
103. Qasim, O.S.; Algamal, Z.Y. Feature selection using different transfer functions for binary bat algorithm. *Int. J. Math. Eng. Manag. Sci.* **2020**, *5*, 697–706. [CrossRef]
104. Crawford, B.; Soto, R.; Lemus-Romani, J.; Becerra-Rozas, M.; Lanza-Gutiérrez, J.M.; Caballé, N.; Castillo, M.; Tapia, D.; Cisternas-Caneo, F.; García, J.; et al. Q-learnheuristics: Towards data-driven balanced metaheuristics. *Mathematics* **2021**, *9*, 1839. [CrossRef]
105. García, J.; Crawford, B.; Soto, R.; Astorga, G. A percentile transition ranking algorithm applied to binarization of continuous swarm intelligence metaheuristics. In Proceedings of the International Conference on Soft Computing and Data Mining, Johor, Malaysia, 6–8 February 2018; pp. 3–13.
106. Lin, G.; Guan, J. A hybrid binary particle swarm optimization for the obnoxious p-median problem. *Inf. Sci.* **2018**, *425*, 1–17. [CrossRef]
107. Xiong, G.; Yuan, X.; Mohamed, A.W.; Zhang, J. Fault section diagnosis of power systems with logical operation binary gaining-sharing knowledge-based algorithm. *Int. J. Intell. Syst.* **2022**, *37*, 1057–1080. [CrossRef]
108. Kumar, V.; Kaur, A. Binary spotted hyena optimizer and its application to feature selection. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 2625–2645. [CrossRef]
109. Qasim, O.S.; Algamal, Z.Y. A gray wolf algorithm for feature and parameter selection of support vector classification. *Int. J. Comput. Sci. Math.* **2021**, *13*, 93–102. [CrossRef]
110. Jain, R.; Joseph, T.; Saxena, A.; Gupta, D.; Khanna, A.; Sagar, K.; Ahlawat, A.K. Feature selection algorithm for usability engineering: A nature inspired approach. *Complex Intell. Syst.* **2021**, 1–11. [CrossRef]
111. Kahya, M.A.; Altamir, S.A.; Algamal, Z.Y. Improving firefly algorithm-based logistic regression for feature selection. *J. Interdiscip. Math.* **2019**, *22*, 1577–1581. [CrossRef]
112. García, J.; Crawford, B.; Soto, R.; Astorga, G. A percentile transition ranking algorithm applied to knapsack problem. In Proceedings of the Computational Methods in Systems and Software, Szczecin, Poland, 12–14 September 2017; pp. 126–138.
113. Mishra, K.; Majhi, S.K. A Dynamic Load Scheduling Using Binary Self-adaptive JAYA (BSAJAYA) Algorithm in Cloud-Based Computing. In *Advances in Intelligent Computing and Communication*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 111–121.
114. Hodashinsky, I.A.; Nemirovich-Danchenko, M.M.; Samsonov, S.S. Feature selection for fuzzy classifier using the spider monkey algorithm. *Бизнес-информатика* **2019**, *13*, 29–42. [CrossRef]
115. Hussien, A.G.; Hassanien, A.E.; Houssein, E.H.; Amin, M.; Azar, A.T. New binary whale optimization algorithm for discrete optimization problems. *Eng. Optim.* **2020**, *52*, 945–959. [CrossRef]
116. Manita, G.; Korbaa, O. Binary political optimizer for feature selection using gene expression data. *Comput. Intell. Neurosci.* **2020**, *2020*, 8896570. [CrossRef]
117. Zangari, M.; Pozo, A.; Santana, R.; Mendiburu, A. A decomposition-based binary ACO algorithm for the multiobjective UBQP. *Neurocomputing* **2017**, *246*, 58–68. [CrossRef]
118. Valenzuela, M.; Jorquera, L.; Villavicencio, G.; Moraga, P.; Pinto, H. A binary ant lion optimizer applied to knapsack problem. In Proceedings of the 2019 6th International Conference on Systems and Informatics (ICSAI), Shanghai, China, 2–4 November 2019; pp. 469–474.
119. Hans, R.; Kaur, H. Hybrid binary Sine Cosine Algorithm and Ant Lion Optimization (SCALO) approaches for feature selection problem. *Int. J. Comput. Mater. Sci. Eng.* **2020**, *9*, 1950021. [CrossRef]
120. Jordehi, A.R. Binary particle swarm optimisation with quadratic transfer function: A new binary optimisation algorithm for optimal scheduling of appliances in smart homes. *Appl. Soft Comput.* **2019**, *78*, 465–480. [CrossRef]
121. Qian, H.; Zhang, Q.; Lei, D.; Pan, Z. A cooperated fruit fly optimization algorithm for Knapsack problem. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017; pp. 591–595.
122. Prakash, J.; Singh, P.K. Gravitational search algorithm and K-means for simultaneous feature selection and data clustering: A multi-objective approach. *Soft Comput.* **2019**, *23*, 2083–2100. [CrossRef]
123. Gölcük, İ.; Ozsoydan, F.B. Evolutionary and adaptive inheritance enhanced Grey Wolf Optimization algorithm for binary domains. *Knowl.-Based Syst.* **2020**, *194*, 105586. [CrossRef]
124. Valenzuela, M.; Valenzuela, P.; Cáceres, C.; Jorquera, L.; Pinto, H. A percentile multi-verse optimizer algorithm applied to the knapsack problem. In Proceedings of the 2019 14th Iberian Conference on Information Systems and Technologies (CISTI), Coimbra, Portugal, 19–22 June 2019; pp. 1–8.

125. Ibrahim, A.M.; Tawhid, M.A.; Ward, R.K. A binary water wave optimization for feature selection. *Int. J. Approx. Reason.* **2020**, *120*, 74–91. [CrossRef]
126. Al-Wajih, R.; Abdulkadir, S.J.; Aziz, N.; Al-Tashi, Q.; Talpur, N. Hybrid binary grey wolf with Harris hawks optimizer for feature selection. *IEEE Access* **2021**, *9*, 31662–31677. [CrossRef]
127. Qu, C.; Zhang, L.; Li, J.; Deng, F.; Tang, Y.; Zeng, X.; Peng, X. Improving feature selection performance for classification of gene expression data using Harris Hawks optimizer with variable neighborhood learning. *Briefings Bioinform.* **2021**, *22*, bbab097. [CrossRef]
128. Rojas-Galeano, S. Binary optimisation with an urban pigeon-inspired swarm algorithm. In Proceedings of the Workshop on Engineering Applications, Santa Marta, Colombia, 16–18 October 2019; pp. 190–201.
129. Mourad, K.; Boudour, R. A modified binary firefly algorithm to solve hardware/software partitioning problem. *Informatica* **2021**, *45*, 1–12. [CrossRef]
130. Tiwari, S.; Kumar, A. Hybrid Taguchi-based Technique for Micro-phasor Measurement Units Placement in the Grid-connected Distribution System. *IETE J. Res.* **2021**, *2021*, 1–13. [CrossRef]
131. Mohammadzadeh, A.; Masdari, M.; Gharehchopogh, F.S.; Jafarian, A. Improved chaotic binary grey wolf optimization algorithm for workflow scheduling in green cloud computing. *Evol. Intell.* **2021**, *14*, 1997–2025. [CrossRef]
132. Awadallah, M.A.; Al-Betar, M.A.; Hammouri, A.I.; Alomari, O.A. Binary JAYA algorithm with adaptive mutation for feature selection. *Arab. J. Sci. Eng.* **2020**, *45*, 10875–10890. [CrossRef]
133. Jorquera, L.; Moraga, P.; Altimiras, F.; Valenzuela, P.; Rubio, J.M. A Machine Learning Whale Algorithm Applied to the Resource Allocation Problems. In Proceedings of the Computational Methods in Systems and Software, Online, Czech Republic, 1 October 2021; pp. 489–498.
134. Faris, H.; Habib, M.; Faris, M.; Alomari, M.; Alomari, A. Medical speciality classification system based on binary particle swarms and ensemble of one vs. rest support vector machines. *J. Biomed. Inform.* **2020**, *109*, 103525. [CrossRef] [PubMed]
135. Ghosh, K.K.; Ahmed, S.; Singh, P.K.; Geem, Z.W.; Sarkar, R. Improved binary sailfish optimizer based on adaptive β-hill climbing for feature selection. *IEEE Access* **2020**, *8*, 83548–83560. [CrossRef]
136. Mafarja, M.; Aljarah, I.; Faris, H.; Hammouri, A.I.; Ala'M, A.Z.; Mirjalili, S. Binary grasshopper optimisation algorithm approaches for feature selection problems. *Expert Syst. Appl.* **2019**, *117*, 267–286. [CrossRef]
137. Valenzuela, M.; Moraga, P.; Causa, L.; Pinto, H.; Rubio, J.M. A Machine Learning Whale Algorithm Applied to the Matrix Covering Problem. In Proceedings of the Computational Methods in Systems and Software, Online, Czech Republic, 1 October 2021; pp. 413–422.
138. Canayaz, M. MH-COVIDNet: Diagnosis of COVID-19 using deep neural networks and meta-heuristic-based feature selection on X-ray images. *Biomed. Signal Process. Control* **2021**, *64*, 102257. [CrossRef]
139. Abd El-Mageed, A.A.; Gad, A.G.; Sallam, K.M.; Munasinghe, K.; Abohany, A.A. Improved Binary Adaptive Wind Driven Optimization Algorithm-Based Dimensionality Reduction for Supervised Classification. *Comput. Ind. Eng.* **2022**, *167*, 107904. [CrossRef]
140. Nssibi, M.; Manita, G.; Korbaa, O. Binary Giza Pyramids Construction For Feature Selection. *Procedia Comput. Sci.* **2021**, *192*, 676–687. [CrossRef]
141. Pinto, H.; Peña, A.; Maureira, C.; Valenzuela, M.; Villavicencio, G. A Machine Learning Firefly Algorithm Applied to the Resource Allocation Problems. In Proceedings of the Computer Science On-line Conference, Online, 1 April 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 297–305.
142. Osuna-Enciso, V.; Pérez-Cisneros, M.; Zaldívar-Navarro, D. Micro-Grooved Pipe Design of Parabolic Trough by Metaheuristic Optimization: An Empirical Comparison. *Energies* **2020**, *13*, 449. [CrossRef]
143. Ledezma, L.F.F.; Alcaraz, G.G. Hybrid binary PSO for transmission expansion planning considering N-1 security criterion. *IEEE Lat. Am. Trans.* **2020**, *18*, 545–553. [CrossRef]
144. Crawford, B.; Soto, R.; de la Fuente Mella, H.; Elortegui, C.; Palma, W.; Torres-Rojas, C.; Vasconcellos-Gaete, C.; Becerra, M.; Peña, J.; Misra, S. Binary Fruit Fly Swarm Algorithms for the Set Covering Problem. *Comput. Mater. Contin.* **2022**, *71*, 4295–4318. [CrossRef]
145. Villavicencio, G.; Valenzuela, M.; Causa, L.; Moraga, P.; Pinto, H. A Machine Learning Firefly Algorithm Applied to the Matrix Covering Problem. In Proceedings of the Computer Science On-line Conference, Online, 1 April 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 316–325.
146. Reséndiz-Flores, E.O.; Navarro-Acosta, J.A.; Hernández-Martínez, A. Optimal feature selection in industrial foam injection processes using hybrid binary Particle Swarm Optimization and Gravitational Search Algorithm in the Mahalanobis–Taguchi System. *Soft Comput.* **2020**, *24*, 341–349. [CrossRef]
147. Li, G.; Le, C. Hybrid Binary Bat Algorithm with Cross-Entropy Method for Feature Selection. In Proceedings of the 2019 4th International Conference on Control and Robotics Engineering (ICCRE), Nanjing, China, 20–23 April 2019; pp. 165–169.
148. Pan, J.S.; Hu, P.; Chu, S.C. Binary fish migration optimization for solving unit commitment. *Energy* **2021**, *226*, 120329. [CrossRef]
149. Abdel-Basset, M.; El-Shahat, D.; Faris, H.; Mirjalili, S. A binary multi-verse optimizer for 0-1 multidimensional knapsack problems with application in interactive multimedia systems. *Comput. Ind. Eng.* **2019**, *132*, 187–206. [CrossRef]
150. Agrawal, P.; Ganesh, T.; Mohamed, A.W. Solving knapsack problems using a binary gaining sharing knowledge-based optimization algorithm. *Complex Intell. Syst.* **2022**, *8*, 43–63. [CrossRef]

151. Shahid, M.; Malik, T.N.; Said, A. Heuristic based binary grasshopper optimization algorithm to solve unitcommitment problem. *Turk. J. Electr. Eng. Comput. Sci.* **2021**, *29*, 944–961. [CrossRef]
152. de Souza, R.C.T.; de Macedo, C.A.; dos Santos Coelho, L.; Pierezan, J.; Mariani, V.C. Binary coyote optimization algorithm for feature selection. *Pattern Recognit.* **2020**, *107*, 107470. [CrossRef]
153. Villavicencio, G.; Valenzuela, M.; Altimiras, F.; Moraga, P.; Pinto, H. A K-Means Grasshopper Optimisation Algorithm Applied to the Set Covering Problem. In Proceedings of the Computer Science On-line Conference, Zlin, Czech Republic, 15 July 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 312–323.
154. Hamdy, A.; Mohamed, A. Greedy binary particle swarm optimization for multi-objective constrained next release problem. *Int. J. Mach. Learn. Comput.* **2019**, *9*, 561–568. [CrossRef]
155. Amic, S.; Soyjaudah, K.S.; Ramsawock, G. Binary cat swarm optimization for cryptanalysis. In Proceedings of the 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Bhubaneswar, India, 17–20 December 2017; pp. 1–6.
156. Pinto, H.; Peña, A.; Causa, L.; Valenzuela, M.; Villavicencio, G. A K-means Grasshopper Algorithm Applied to the Knapsack Problem. In Proceedings of the Computer Science On-line Conference, Zlin, Czech Republic, 15 July 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 234–244.
157. Gupta, D.; Rodrigues, J.J.; Sundaram, S.; Khanna, A.; Korotaev, V.; de Albuquerque, V.H.C. Usability feature extraction using modified crow search algorithm: A novel approach. *Neural Comput. Appl.* **2020**, *32*, 10915–10925. [CrossRef]
158. García, J.; Lalla-Ruiz, E.; Voß, S.; Droguett, E.L. Enhancing a machine learning binarization framework by perturbation operators: Analysis on the multidimensional knapsack problem. *Int. J. Mach. Learn. Cybern.* **2020**, *11*, 1951–1970. [CrossRef]
159. Aytimur, A.; Babayigit, B. Binary Artificial Bee Colony Algorithms for {0-1} Advertisement Problem. In Proceedings of the 2019 6th International Conference on Electrical and Electronics Engineering (ICEEE), Istanbul, Turkey, 16–17 April 2019; pp. 91–95.
160. García, J.; Crawford, B.; Soto, R.; Castro, C.; Paredes, F. A k-means binarization framework applied to multidimensional knapsack problem. *Appl. Intell.* **2018**, *48*, 357–380. [CrossRef]
161. Manohar, L.; Ganesan, K. Diagnosis of schizophrenia disorder in MR brain images using multi-objective BPSO based feature selection with fuzzy SVM. *J. Med. Biol. Eng.* **2018**, *38*, 917–932. [CrossRef]
162. Mafarja, M.; Jarrar, R.; Ahmad, S.; Abusnaina, A.A. Feature selection using binary particle swarm optimization with time varying inertia weight strategies. In Proceedings of the 2nd International Conference on Future Networks and Distributed Systems, Amman, Jordan, 26–27 June 2018; pp. 1–9.
163. Turkoglu, B.; Uymaz, S.A.; Kaya, E. Binary Artificial Algae Algorithm for feature selection. *Appl. Soft Comput.* **2022**, *120*, 108630. [CrossRef]
164. Pavez, L.; Altimiras, F.; Villavicencio, G. A K-means Bat Optimisation Algorithm Applied to the Set Covering Problem. In Proceedings of the Computational Methods in Systems and Software, Online, Czech Republic, 14–17 October 2020; pp. 622–632.
165. Bardamova, M.; Konev, A.; Hodashinsky, I.; Shelupanov, A. A fuzzy classifier with feature selection based on the gravitational search algorithm. *Symmetry* **2018**, *10*, 609. [CrossRef]
166. Papa, J.P.; Rosa, G.H.; de Souza, A.N.; Afonso, L.C. Feature selection through binary brain storm optimization. *Comput. Electr. Eng.* **2018**, *72*, 468–481. [CrossRef]
167. Devi, R.M.; Premkumar, M.; Jangir, P.; Kumar, B.S.; Alrowaili, D.; Nisar, K.S. BHGSO: Binary Hunger Games Search Optimization Algorithm for Feature Selection Problem. *CMC-Comput. Mater. Contin.* **2022**, *70*, 557–579. [CrossRef]
168. Jorquera, L.; Valenzuela, P.; Valenzuela, M.; Pinto, H. A binary ant lion optimisation algorithm applied to the set covering problem. In Proceedings of the Computer Science On-line Conference, Zlin, Czech Republic, 24–27 April 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 156–167.
169. Tütüncü, K.; Şahman, M.A.; Tuşat, E. A hybrid binary grey wolf optimizer for selection and reduction of reference points with extreme learning machine approach on local GNSS/leveling geoid determination. *Appl. Soft Comput.* **2021**, *108*, 107444. [CrossRef]
170. Tapia, D.; Crawford, B.; Soto, R.; Palma, W.; Lemus-Romani, J.; Cisternas-Caneo, F.; Castillo, M.; Becerra-Rozas, M.; Paredes, F.; Misra, S. Embedding Q-Learning in the selection of metaheuristic operators: The enhanced binary grey wolf optimizer case. In Proceedings of the 2021 IEEE International Conference on Automation/XXIV Congress of the Chilean Association of Automatic Control (ICA-ACCA), Online, 22–26 March 2021; pp. 1–6.
171. Nadimi-Shahraki, M.H.; Banaie-Dezfouli, M.; Zamani, H.; Taghian, S.; Mirjalili, S. B-MFO: A Binary Moth-Flame Optimization for Feature Selection from Medical Datasets. *Computers* **2021**, *10*, 136. [CrossRef]
172. Pavez, L.; Altimiras, F.; Villavicencio, G. A K-means Bat Algorithm Applied to the Knapsack Problem. In Proceedings of the Computational Methods in Systems and Software, Online, Czech Republic, 14–17 October 2020; pp. 612–621.
173. Zhong, C.; Chen, Y.; Peng, J. Feature selection based on a novel improved tree growth algorithm. *Int. J. Comput. Intell. Syst.* **2020**, *13*, 247. [CrossRef]
174. Al-Betar, M.A.; Hammouri, A.I.; Awadallah, M.A.; Abu Doush, I. Binary β-hill climbing optimizer with S-shape transfer function for feature selection. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 7637–7665. [CrossRef]
175. Islam, M.; Shareef, H.; Mohamed, A.; Wahyudie, A. A binary variant of lightning search algorithm: BLSA. *Soft Comput.* **2017**, *21*, 2971–2990. [CrossRef]

176. García, J.; Maureira, C. A KNN quantum cuckoo search algorithm applied to the multidimensional knapsack problem. *Appl. Soft Comput.* **2021**, *102*, 107077. [CrossRef]
177. Sarhani, M.; Afia, A.E.; Faizi, R. Facing the feature selection problem with a binary PSO-GSA approach. In *Recent Developments in Metaheuristics*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 447–462.
178. Zhao, J.; Liu, S.; Zhou, M.; Guo, X.; Qi, L. An improved binary cuckoo search algorithm for solving unit commitment problems: Methodological description. *IEEE Access* **2018**, *6*, 43535–43545. [CrossRef]
179. García, J.; Moraga, P.; Valenzuela, M.; Pinto, H. A db-scan hybrid algorithm: An application to the multidimensional knapsack problem. *Mathematics* **2020**, *8*, 507. [CrossRef]
180. Too, J.; Rahim Abdullah, A. Binary atom search optimisation approaches for feature selection. *Connect. Sci.* **2020**, *32*, 406–430. [CrossRef]
181. Alsharoa, A.; Celik, A.; Kamal, A.E. Energy Efficient 5G Networks Using Joint Energy Harvesting and Scheduling. *5G Netw. Fundam. Requir. Enabling Technol. Oper. Manag.* **2018**, *2018*, 427–452.
182. Lanza-Gutierrez, J.M.; Soto, R.; Crawford, B.; Gomez-Pulido, J.A.; Fernandez, N.; Castillo, C. Applying an Electromagnetism-like algorithm for solving the manufacturing cell design problem. In *Intelligent Systems: Concepts, Methodologies, Tools, and Applications*; IGI Global: Pennsylvania, PA, USA, 2018; pp. 1212–1231.
183. García, J.; Moraga, P.; Valenzuela, M.; Crawford, B.; Soto, R.; Pinto, H.; Peña, A.; Altimiras, F.; Astorga, G. A db-scan binarization algorithm applied to matrix covering problems. *Comput. Intell. Neurosci.* **2019**, *2019*, 3238574. [CrossRef]
184. Awadallah, M.A.; Hammouri, A.I.; Al-Betar, M.A.; Braik, M.S.; Abd Elaziz, M. Binary Horse herd optimization algorithm with crossover operators for feature selection. *Comput. Biol. Med.* **2022**, *141*, 105152. [CrossRef]
185. Mafarja, M.; Qasem, A.; Heidari, A.A.; Aljarah, I.; Faris, H.; Mirjalili, S. Efficient hybrid nature-inspired binary optimizers for feature selection. *Cogn. Comput.* **2020**, *12*, 150–175. [CrossRef]
186. Guendouzi, W.; Boukra, A. An enhanced bat echolocation approach for security audit trails analysis using manhattan distance. In *Recent Developments in Metaheuristics*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 477–493.
187. Chu, X.; Li, S.; Gao, D.; Zhao, W.; Cui, J.; Huang, L. A binary superior tracking artificial bee colony with dynamic Cauchy mutation for feature selection. *Complexity* **2020**, *2020*, 8864315. [CrossRef]
188. Yepes, V.; Martí, J.V.; García, J. Black hole algorithm for sustainable design of counterfort retaining walls. *Sustainability* **2020**, *12*, 2767. [CrossRef]
189. Thaher, T.; Arman, N. Efficient multi-swarm binary harris hawks optimization as a feature selection approach for software fault prediction. In Proceedings of the 2020 11th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 7–9 April 2020; pp. 249–254.
190. Jiang, Y.; Luo, Q.; Wei, Y.; Abualigah, L.; Zhou, Y. An efficient binary Gradient-based optimizer for feature selection. *Math. Biosci. Eng.* **2021**, *18*, 3813–3854. [CrossRef] [PubMed]
191. Hameed, S.S.; Hassan, W.H.; Latiff, L.A.; Muhammadsharif, F.F. A comparative study of nature-inspired metaheuristic algorithms using a three-phase hybrid approach for gene selection and classification in high-dimensional cancer datasets. *Soft Comput.* **2021**, *25*, 8683–8701. [CrossRef]
192. Agrawal, P.; Ganesh, T.; Mohamed, A.W. Chaotic gaining sharing knowledge-based optimization algorithm: An improved metaheuristic algorithm for feature selection. *Soft Comput.* **2021**, *25*, 9505–9528. [CrossRef]
193. Ouadfel, S.; Abd Elaziz, M. Efficient high-dimension feature selection based on enhanced equilibrium optimizer. *Expert Syst. Appl.* **2022**, *187*, 115882. [CrossRef]
194. Gao, Y.; Zhou, Y.; Luo, Q. An efficient binary equilibrium optimizer algorithm for feature selection. *IEEE Access* **2020**, *8*, 140936–140963. [CrossRef]
195. Cisternas-Caneo, F.; Crawford, B.; Soto, R.; de la Fuente-Mella, H.; Tapia, D.; Lemus-Romani, J.; Castillo, M.; Becerra-Rozas, M.; Paredes, F.; Misra, S. A data-driven dynamic discretization framework to solve combinatorial problems using continuous metaheuristics. In Proceedings of the International Conference on Innovations in Bio-Inspired Computing and Applications, Online, 16–18 December 2020; pp. 76–85.
196. Pandey, A.C.; Rajpoot, D.S.; Saraswat, M. Feature selection method based on hybrid data transformation and binary binomial cuckoo search. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 719–738. [CrossRef]
197. Ozsoydan, F.B. Effects of dominant wolves in grey wolf optimization algorithm. *Appl. Soft Comput.* **2019**, *83*, 105658. [CrossRef]
198. Pavez, L.; Altimiras, F.; Villavicencio, G. A Percentil Gravitational Search Algorithm an Aplication to the Set Covering Problem. In Proceedings of the Computational Methods in Systems and Software, Online, Czech Republic, 14–17 October 2020; pp. 663–673.
199. García, J.; Crawford, B.; Soto, R.; Astorga, G. A clustering algorithm applied to the binarization of swarm intelligence continuous metaheuristics. *Swarm Evol. Comput.* **2019**, *44*, 646–664. [CrossRef]
200. Jain, S.; Dharavath, R. Memetic salp swarm optimization algorithm based feature selection approach for crop disease detection system. *J. Ambient. Intell. Humaniz. Comput.* **2021**, 1–19. [CrossRef]
201. Abdi Alkareem Alyasseri, Z.; Alomari, O.A.; Al-Betar, M.A.; Awadallah, M.A.; Hameed Abdulkareem, K.; Abed Mohammed, M.; Kadry, S.; Rajinikanth, V.; Rho, S. EEG Channel Selection Using Multiobjective Cuckoo Search for Person Identification as Protection System in Healthcare Applications. *Comput. Intell. Neurosci.* **2022**, *2022*, 5974634. [CrossRef] [PubMed]
202. Pashaei, E.; Pashaei, E. An efficient binary chimp optimization algorithm for feature selection in biomedical data classification. *Neural Comput. Appl.* **2022**, *34*, 6427–6451. [CrossRef]

203. Moraga, P.; Valenzuela, P.; Altimiras, F.; Lopez, L.; Rubio, J.M. A Binary Whale Algorithm Applied to Knapsack Problem. In Proceedings of the Computational Methods in Systems and Software, Online, Czech Republic, 1 October 2021; pp. 465–474.
204. Ross, O.H.M. A review of quantum-inspired metaheuristics: Going from classical computers to real quantum computers. *IEEE Access* **2019**, *8*, 814–838. [CrossRef]
205. Baykasoğlu, A.; Ozsoydan, F.B. Dynamic optimization in binary search spaces via weighted superposition attraction algorithm. *Expert Syst. Appl.* **2018**, *96*, 157–174. [CrossRef]
206. Piri, J.; Mohapatra, P. An analytical study of modified multi-objective Harris Hawk Optimizer towards medical data feature selection. *Comput. Biol. Med.* **2021**, *135*, 104558. [CrossRef] [PubMed]
207. Yang, Z.; Li, K.; Guo, Y.; Feng, S.; Niu, Q.; Xue, Y.; Foley, A. A binary symmetric based hybrid meta-heuristic method for solving mixed integer unit commitment problem integrating with significant plug-in electric vehicles. *Energy* **2019**, *170*, 889–905. [CrossRef]
208. ÖZSOYDAN, F. A quantum based local search enhanced particle swarm optimization for binary spaces. *Pamukkale Univ. J. Eng. Sci.-Pamukkale Univ. Muhendis. Bilim. Derg.* **2018**, *24*, 675–681. [CrossRef]
209. Lai, X.; Hao, J.K.; Fu, Z.H.; Yue, D. Diversity-preserving quantum particle swarm optimization for the multidimensional knapsack problem. *Expert Syst. Appl.* **2020**, *149*, 113310. [CrossRef]
210. Durgut, R.; Aydin, M.E. Adaptive binary artificial bee colony algorithm. *Appl. Soft Comput.* **2021**, *101*, 107054. [CrossRef]
211. García, J.; Martí, J.V.; Yepes, V. The buttressed walls problem: An application of a hybrid clustering particle swarm optimization algorithm. *Mathematics* **2020**, *8*, 862. [CrossRef]
212. Quintero-Duran, M.J.; Candelo-Becerra, J.E.; Cabana-Jimenez, K. Distribution network reconfiguration with large number of switches solved by a modified binary bat algorithm and improved seed population. *Tehnički vjesnik* **2019**, *26*, 1284–1291.
213. De Souza, R.C.T.; dos Santos Coelho, L.; De Macedo, C.A.; Pierezan, J. A V-shaped binary crow search algorithm for feature selection. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
214. Dhal, P.; Azad, C. A multi-objective feature selection method using newton's law based pso with gwo. *Appl. Soft Comput.* **2021**, *107*, 107394. [CrossRef]
215. Jorquera, L.; Valenzuela, P.; Altimiras, F.; Moraga, P.; Villavicencio, G. A Percentil Bat Algorithm an Application to the Set Covering Problem. In Proceedings of the Computer Science On-line Conference, Zlin, Czech Republic, 15 July 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 223–233.
216. Zhao, S.; Wang, P.; Heidari, A.A.; Zhao, X.; Ma, C.; Chen, H. An enhanced Cauchy mutation grasshopper optimization with trigonometric substitution: Engineering design and feature selection. *Eng. Comput.* **2021**, *38*, 4583–4616. [CrossRef]
217. Mafarja, M.; Heidari, A.A.; Faris, H.; Mirjalili, S.; Aljarah, I. Dragonfly algorithm: Theory, literature review, and application in feature selection. *Nat.-Inspired Optim.* **2020**, *2020*, 47–67.
218. Crawford, B.; Soto, R.; Palma, W.; Aballay, F.; Lemus-Romani, J.; Misra, S.; Castro, C.; Paredes, F.; Rubio, J.M. A teaching-learning-based optimization algorithm for the weighted set-covering problem. *Tehnički Vjesnik* **2020**, *27*, 1678–1684.
219. Agarwal, T.; Kumar, V. A Systematic Review on Bat Algorithm: Theoretical Foundation, Variants, and Applications. *Arch. Comput. Methods Eng.* **2021**, *29*, 2707–2736. [CrossRef]
220. Valdivia, S.; Soto, R.; Crawford, B.; Caselli, N.; Paredes, F.; Castro, C.; Olivares, R. Clustering-based binarization methods applied to the crow search algorithm for 0/1 combinatorial problems. *Mathematics* **2020**, *8*, 1070. [CrossRef]
221. Ul Haq, S.S.; Ahmad, A.; Ikram, F.; Nawaz, T.; Majeed, A.; Saddique, M.S.; Sattar, M.K. A novel binary variant model of swarm inspired polar bear optimization algorithm employed for scalable unit commitment. *Int. Trans. Electr. Energy Syst.* **2021**, *31*, e12711. [CrossRef]
222. Mafarja, M.; Aljarah, I.; Heidari, A.A.; Faris, H.; Fournier-Viger, P.; Li, X.; Mirjalili, S. Binary dragonfly optimization for feature selection using time-varying transfer functions. *Knowl.-Based Syst.* **2018**, *161*, 185–204. [CrossRef]
223. Ghosh, K.K.; Singh, P.K.; Hong, J.; Geem, Z.W.; Sarkar, R. Binary social mimic optimization algorithm with x-shaped transfer function for feature selection. *IEEE Access* **2020**, *8*, 97890–97906. [CrossRef]
224. Sharma, S.R.; Singh, B.; Kaur, M. Improving the classification of phishing websites using a hybrid algorithm. *Comput. Intell.* **2021**, *38*, 667–689. [CrossRef]
225. Kahya, M.A.; Altamir, S.A.; Algamal, Z.Y. Improving whale optimization algorithm for feature selection with a time-varying transfer function. *Numer. Algebra Control Optim.* **2021**, *11*, 87. [CrossRef]
226. Crawford, B.; Soto, R.; Cisternas-Caneo, F.; Tapia, D.; de la Fuente-Mella, H.; Palma, W.; Lemus-Romani, J.; Castillo, M.; Becerra-Rozas, M. A comparison of Learnheuristics using different Reward Functions to solve the Set Covering Problem. In Proceedings of the International Conference on Optimization and Learning, Grasmere, UK, 4–8 October 2021; pp. 74–85.
227. Lemus-Romani, J.; Becerra-Rozas, M.; Crawford, B.; Soto, R.; Cisternas-Caneo, F.; Vega, E.; Castillo, M.; Tapia, D.; Astorga, G.; Palma, W.; et al. A novel learning-based binarization scheme selector for swarm algorithms solving combinatorial problems. *Mathematics* **2021**, *9*, 2887. [CrossRef]
228. Huang, X.; Li, P.; Pu, Y. Amplitude angle modulated bat algorithm with application to zero-one knapsack problem. *IEEE Access* **2019**, *7*, 27957–27969. [CrossRef]

229. Wang, L.; Shi, R.; Li, W.; Yuan, X.; Dong, J. An Improved Dragonfly Algorithm Based on Angle Modulation Mechanism for Solving 0–1 Knapsack Problems. In Proceedings of the International Conference on Swarm Intelligence, Qingdao, China, 17–21 July 2021; pp. 83–93.
230. Dong, J.; Wang, Z.; Mo, J. A phase angle-modulated bat algorithm with application to antenna topology optimization. *Appl. Sci.* **2021**, *11*, 2243. [CrossRef]
231. Wang, L.; Shi, R.; Dong, J. A Hybridization of Dragonfly Algorithm Optimization and Angle Modulation Mechanism for 0-1 Knapsack Problems. *Entropy* **2021**, *23*, 598. [CrossRef]
232. Sun, W.Z.; Zhang, M.; Wang, J.S.; Guo, S.S.; Wang, M.; Hao, W.K. Binary Particle Swarm Optimization Algorithm Based on Z-shaped Probability Transfer Function to Solve 0-1 Knapsack Problem. *IAENG Int. J. Comput. Sci.* **2021**, *48*, 294–303.
233. Rizk-Allah, R.M.; Hassanien, A.E.; Elhoseny, M.; Gunasekaran, M. A new binary salp swarm algorithm: Development and application for optimization tasks. *Neural Comput. Appl.* **2019**, *31*, 1641–1663. [CrossRef]
234. Guo, S.s.; Wang, J.s.; Guo, M.w. Z-shaped transfer functions for binary particle swarm optimization algorithm. *Comput. Intell. Neurosci.* **2020**, *2020*, 6502807. [CrossRef] [PubMed]
235. Mirjalili, S.; Zhang, H.; Mirjalili, S.; Chalup, S.; Noman, N. A novel U-shaped transfer function for binary particle swarm optimisation. In *Soft Computing for Problem Solving 2019*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 241–259.
236. Beheshti, Z. A novel x-shaped binary particle swarm optimization. *Soft Comput.* **2021**, *25*, 3013–3042. [CrossRef]
237. Feng, Y.; An, H.; Gao, X. The importance of transfer function in solving set-union knapsack problem based on discrete moth search algorithm. *Mathematics* **2018**, *7*, 17. [CrossRef]
238. Gölcük, İ.; Ozsoydan, F.B.; Durmaz, E.D. Analysis of Different Binarization Techniques within Whale Optimization Algorithm. In Proceedings of the 2019 Innovations in Intelligent Systems and Applications Conference (ASYU), Izmir, Turkey, 31 October–2 November 2019; pp. 1–5.
239. Kennedy, J.; Eberhart, R.C. A discrete binary version of the particle swarm algorithm. In Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997; Volume 5, pp. 4104–4108.
240. Crawford, B.; Soto, R.; Peña, C.; Palma, W.; Johnson, F.; Paredes, F. Solving the set covering problem with a shuffled frog leaping algorithm. In Proceedings of the Asian Conference on Intelligent Information and Database Systems, Bali, Indonesia, 23–25 March 2015; pp. 41–50.
241. Crawford, B.; Soto, R.; Astudillo, G.; Olguín, E.; Misra, S. Solving set covering problem with fireworks explosion. In Proceedings of the International Conference on Computational Science and Its Applications, Beijing, China, 4–7 July 2016; pp. 273–283.
242. Proakis, J.G.; Salehi, M.; Zhou, N.; Li, X. *Communication Systems Engineering*; Prentice Hall: Hoboken, NJ, USA, 1994; Volume 2.
243. Afshinmanesh, F.; Marandi, A.; Rahimi-Kian, A. A novel binary particle swarm optimization method using artificial immune system. In Proceedings of the EUROCON 2005-The International Conference on "Computer as a Tool", Belgrade, Serbia and Montenegro, 21–24 November 2005; Volume 1, pp. 217–220.
244. Kotsiantis, S.B.; Zaharakis, I.; Pintelas, P. Supervised machine learning: A review of classification techniques. *Emerg. Artif. Intell. Appl. Comput. Eng.* **2007**, *160*, 3–24.
245. Celebi, M.E.; Aydin, K. *Unsupervised Learning Algorithms*; Springer: Berlin/Heidelberg, Germany, 2016; Volume 9.
246. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
247. Becerra-Rozas, M.; Lemus-Romani, J.; Crawford, B.; Soto, R.; Cisternas-Caneo, F.; Embry, A.T.; Molina, M.A.; Tapia, D.; Castillo, M.; Rubio, J.M. A New Learnheuristic: Binary SARSA-Sine Cosine Algorithm (BS-SCA). In Proceedings of the International Conference on Metaheuristics and Nature Inspired Computing, Marrakech, Morocco, 27–30 October 2021; pp. 127–136.
248. Talbi, E.G. *Metaheuristics: From Design to Implementation*; John Wiley & Sons: Hoboken, NJ, USA, 2009.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.