



HAL
open science

Metaheuristic Biclustering Algorithms: From State-of-the-Art to Future Opportunities

Adán José-García, Julie Jacques, Vincent Sobanski, Clarisse Dhaenens

► **To cite this version:**

Adán José-García, Julie Jacques, Vincent Sobanski, Clarisse Dhaenens. Metaheuristic Biclustering Algorithms: From State-of-the-Art to Future Opportunities. *ACM Computing Surveys*, 2023, 53 (3), pp.1-38. 10.1145/3617590 . hal-04205406v2

HAL Id: hal-04205406

<https://hal.science/hal-04205406v2>

Submitted on 13 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Metaheuristic Biclustering Algorithms: From State-of-the-art to Future Opportunities

ADÁN JOSÉ-GARCÍA and JULIE JACQUES, Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRISTAL, Lille, France

VINCENT SOBANSKI, Univ. Lille, Inserm, CHU Lille, U1286 - INFINITE - Institute for Translational Research in Inflammation, Lille, France and Institut Universitaire de France (IUF), France

CLARISSE DHAENENS, Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRISTAL, Lille, France

Biclustering is an unsupervised machine-learning technique that simultaneously clusters rows and columns in a data matrix. Over the past two decades, the field of biclustering has emerged and grown significantly, and currently plays an essential role in various applications such as bioinformatics, text mining, and pattern recognition. However, finding significant biclusters in large-scale datasets is an NP-hard problem that can be formulated as an optimization problem. Therefore, metaheuristics have been applied to address biclustering problems due to their (i) ability to efficiently explore search spaces of complex optimization problems, (ii) capability to find solutions in reasonable computation time, and (iii) facility to adapt to different problem formulations, as they are considered general-purpose heuristic algorithms. Although several studies on biclustering approaches have been proposed, a comprehensive study using metaheuristics for bicluster analysis is missing. This work presents a survey of metaheuristic approaches to address the biclustering problem in various scientific applications. The review focuses on the underlying optimization methods and their main search components: representation, objective function, and variation operators. A specific discussion on single versus multi-objective approaches is presented. Finally, some emerging research directions are presented.

CCS Concepts: • **Computing methodologies** → **Cluster analysis**; **Bio-inspired approaches**; **Discrete space search**; **Continuous space search**;

Additional Key Words and Phrases: Metaheuristics, biclustering, co-clustering, subspace clustering

ACM Reference format:

Adán José-García, Julie Jacques, Vincent Sobanski, and Clarisse Dhaenens. 2023. Metaheuristic Biclustering Algorithms: From State-of-the-art to Future Opportunities. *ACM Comput. Surv.* 56, 3, Article 69 (October 2023), 38 pages.

<https://doi.org/10.1145/3617590>

Authors' addresses: A. José-García, J. Jacques, and C. Dhaenens, Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRISTAL, F-59000 Lille, Lille, France; e-mails: adan.josegarcia@univ-lille.fr, julie.jacques@univ-lille.fr, clarisse.dhaenens@univ-lille.fr; V. Sobanski, Univ. Lille, Inserm, CHU Lille, U1286 - INFINITE - Institute for Translational Research in Inflammation, F-59000 Lille, Lille, France and Institut Universitaire de France (IUF), Paris, France; e-mail: vincent.sobanski@univ-lille.fr.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

0360-0300/2023/10-ART69 \$15.00

<https://doi.org/10.1145/3617590>

1 INTRODUCTION

Biclustering¹ is an unsupervised machine-learning technique aiming to simultaneously cluster rows and columns of a data matrix. Biclustering algorithms aim to discover homogeneous submatrices (biclusters) providing unique advantages [72, 120]: (i) they can unravel local patterns, i.e., observations meaningfully correlated on a subset of attributes; (ii) they provide flexibility in the definition of different types of biclusters; (iii) they allow overlapping biclusters, such that one observation can belong to more than one bicluster; and (iv) they help alleviate the “*curse of dimensionality*” problem [87]. Therefore, biclustering is a helpful and active research field that has gradually become a widely used technique in different applications such as bioinformatics [47, 80], text mining [10, 19, 85], dimensionality reduction [87], recommendation systems [18, 109, 117], energy [29] and water [105] consumption, time-series analysis [11, 42], disease identification [115, 120], marked segmentation [62], and many more [90, 120]. To show the volume of studies currently being published in the field of biclustering, Figure 1 presents a year-count of research publications about biclustering in the past 22 years. This plot shows an increasing trend in the number of publications since the first works in 2000.

The biclustering task has been proven to be an NP-complete² problem [12], leading to the proposal of optimization methods as heuristics to solve this combinatorial problem. Thus, the development of effective heuristic methods and appropriate cost functions are critical factors for discovering meaningful biclusters from complex datasets in various real-world applications. In this regard, Pontes et al. [90] analyzed a large number of approaches to biclustering and classified them depending on whether they use evaluation metrics within the search heuristic method. In particular, nature-inspired metaheuristics have gradually been applied successfully to biclustering problems because of their excellent exploratory capability [80, 90, 102]. These approaches model the behavior of natural phenomena, which exhibit an ability to learn or adapt to new situations to solve problems in complex and changing environments [49].

Several reviews on bicluster analysis have been conducted, emphasizing different aspects and perspectives of the biclustering problem [72, 90, 120]. However, only Pontes et al. [90] surveyed some methods of biclustering based on metaheuristics as part of a more extensive survey focused on gene expression data. José-García et al. [52] have recently published a preliminary paper that briefly summarizes some metaheuristic methods for biclustering problems. Therefore, we present an up-to-date overview of metaheuristic-based biclustering approaches that have been reported in the past 22 years. This survey contributes to the following main aspects:

- A review of important aspects of the biclustering task when addressed as an optimization problem, such as encoding schemes, variation operators, and bicluster quality metrics.
- An in-depth review of single-objective and multi-objective metaheuristics applied to bicluster analysis.
- An overview of available datasets, software tools, and external bicluster metrics for biclustering problems.
- A discussion of research opportunities in biclustering, a guide to the different components of metaheuristics, and future trends in the field of biclustering.

The outline of this review is as follows: Section 2 describes the basic terms and concepts related to biclustering analysis. Section 3 outlines the main components involved in metaheuristic-based

¹Biclustering is also referred to as co-clustering, subspace clustering, bi-dimensional, or two-way clustering in the specialized literature.

²Nondeterministic Polynomial-time Complete: Any of a class of computational problems for which no efficient solution algorithm has been found.

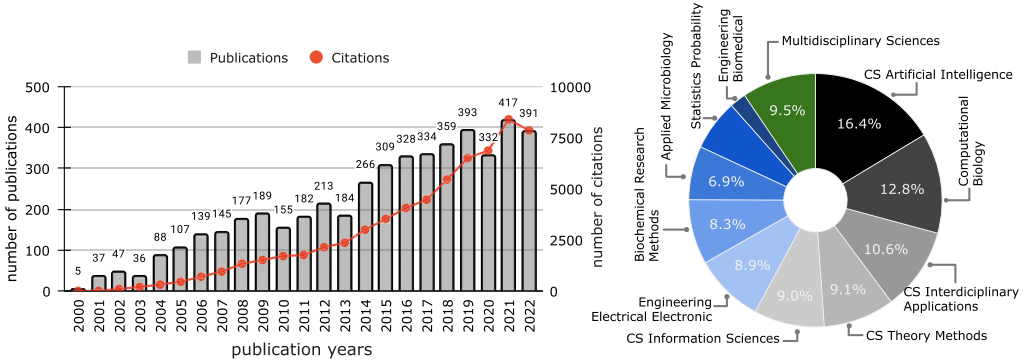


Fig. 1. Number of publications and citations on *biclustering* (left) and publication percentages in the top 10 research fields (right) throughout the past 22 years. This bibliometric data and research fields were extracted from the Web of Science (WoS) database using the keywords *biclustering*, *co-clustering*, and *subspace clustering*. “CS” is an abbreviation for *Computer Science*.

biclustering algorithms. Section 4 reviews single-objective biclustering algorithms for optimizing a unique objective function. Section 5 presents multi-objective biclustering metaheuristics, which optimize multiple cost functions simultaneously. In Section 6, we provide an overview of datasets, software tools, and validation metrics for biclustering problems. Section 7 presents perspectives and future tendencies in biclustering. Finally, the conclusions are given in Section 8.

2 BASIC PRELIMINARIES

In this section, we define the problem of biclustering, present classical approaches, and expose how this problem has been modeled as a combinatorial optimization one.

2.1 Biclustering Definition

Given a data matrix $X \in \mathbb{R}^{N \times M}$ where N denotes the number of patterns (rows), and M denotes the number of attributes (columns). Let us define a set of patterns as R and the set of attributes as C ; therefore, the matrix $X_{R,C} = (R, C)$ denotes the full dataset X . Thus, a bicluster is a subset of rows exhibiting similar behavior across a subset of columns, denoted as $B_{I,J} = (I, J)$ such that $I \subseteq R$ and $J \subseteq C$.

Depending on the application and nature of data, several *types of biclusters* have been described in the literature [72, 90]. In the following definitions, let $B_{I,J} = (I, J)$ be a bicluster in which b_{ij} refers to the value of the i th pattern under the j th attribute. π represents any constant value for B , α_i ($1 \leq i \leq |I|$) and α_j ($1 \leq j \leq |J|$) refers to the constant values used in the additive models for each pattern and attribute; and β_i ($1 \leq i \leq |I|$) and β_j ($1 \leq j \leq |J|$) corresponds to the constant values used in the multiplicative models. The different types of biclusters are described in detail below and illustrated in Figure 2 based on this notation:

- *Biclusters with constant values.* A bicluster with constant values reveals subsets of rows and subsets of columns with similar behaviors. This type of bicluster is expressed by: $b_{ij} = \pi$.
- *Biclusters with constant values on rows or columns.* A bicluster with constant values in the rows or columns identifies a subset of rows/columns with similar behavior in a subset of rows/columns. Therefore, the values can vary from row to row or column to column. It can be expressed in the additive or multiplicative form:
 - Constant rows: $b_{ij} = \pi + \alpha_i$ or $b_{ij} = \pi \times \beta_i$
 - Constant columns: $b_{ij} = \pi + \alpha_j$ or $b_{ij} = \pi \times \beta_j$

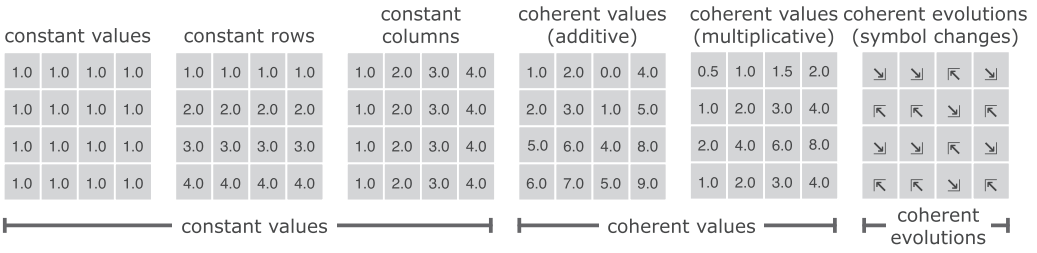


Fig. 2. Examples of different bicluster types with constant values, coherent values, and coherent evolutions.

- *Biclusters with coherent values on both rows and columns* [2]. A bicluster with coherent values in the rows or columns identifies a subset of rows/columns with coherent behavior, regardless of the exact numerical values over a subset of rows/columns. It can also be expressed as either additive or multiplicative:
 - Additive: $b_{ij} = \pi + \alpha_i + \alpha_j$
 - Multiplicative: $b_{ij} = \pi \times \beta_i \times \beta_j$
- *Bicluster with coherent evolutions*: A subset of patterns (rows) is up-regulated or down-regulated coherently across subsets of attributes (columns) irrespective of their actual values; that is, in the same directions but with varying magnitude. In these scenarios, coherent evolutions are difficult to model using a mathematical formulation.

Based on the previous definitions of bicluster types, it is possible to formally describe three kinds of patterns [2]:

- *Perfect shifting pattern*. A bicluster follows this pattern if its values can be obtained by adding a constant-condition number α_j to a typical value for each row (π_i), where α_j denotes the *shifting coefficient* for column j . The bicluster values fulfill the following equation: $b_{ij} = \pi_i + \alpha_j$.
- *Perfect scaling pattern*. A bicluster follows this pattern if its values can be obtained by multiplying a constant-condition number β_j to a typical value for each row (π_i), where β_j is called the *scaling coefficient* for column j . Although the rows present the same behavior, changes are more abrupt for some rows than others. The bicluster values are mathematically defined as follows: $b_{ij} = \pi_i \times \beta_j$.
- *Perfect shifting and scaling pattern*. A bicluster follows this pattern if it exhibits coherent values on both rows and columns, i.e., for the additive and multiplicative model at the same time. The bicluster can be represented by: $b_{ij} = \pi_i \times \beta_j + \alpha_j$.

In most real-world problems, cluster analysis involves the extraction of several biclusters, where the relations between the biclusters are defined by two criteria, *exclusivity* and *exhaustivity* [72, 102]. The exclusivity criterion indicates that an element must belong to a single bicluster, whereas the exhaustivity criterion specifies that every element must be part of one or more biclusters. Exclusivity commonly refers to covering the input matrix, while exhaustivity is associated with overlapping among biclusters.

Considering these two criteria, several *bicluster structures* can be obtained [72]. Figure 3 illustrates some examples of bicluster structures, for instance, biclusters with exclusive rows or exclusive columns (no overlapping and partial covering), exclusive rows and exhaustive columns (columns overlapping and partial covering), exhaustive rows and exhaustive columns (full overlapping and full covering). Consequently, the selection of different bicluster types and structures to be discovered depends on both the problem to be solved and the type of data involved. Therefore, the biclustering algorithm should be able to identify the desired biclusters.

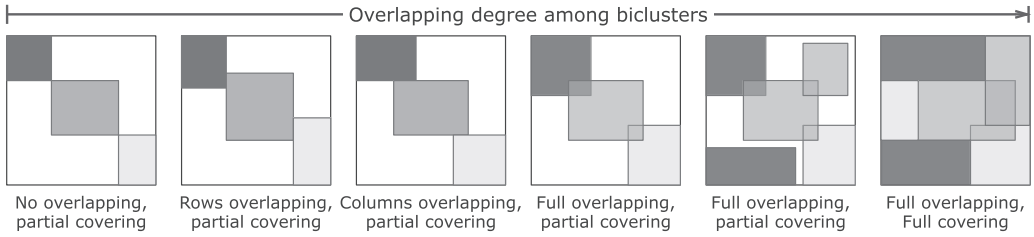


Fig. 3. Examples of different bicluster structures based on exclusivity (matrix covering) and exhaustivity criteria (bicluster overlapping).

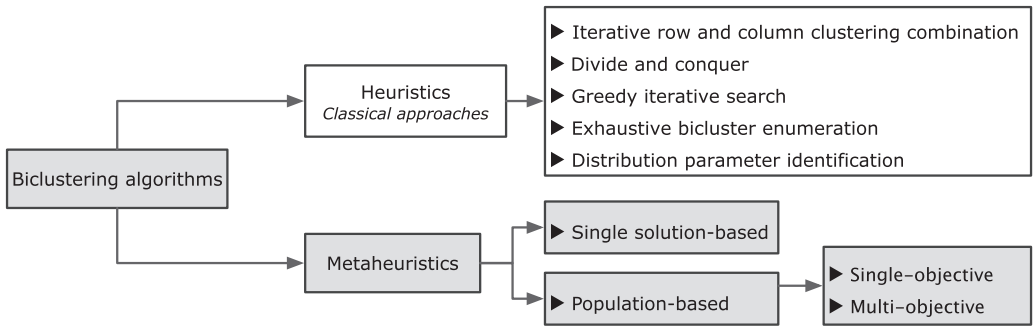


Fig. 4. Biclustering algorithms taxonomy. In this survey, metaheuristic biclustering algorithms highlighted in gray are reviewed.

2.2 Classical Biclustering Approaches

Biclustering has gained much interest since the seminal work of Cheng and Church on biclustering to analyze gene expression data [12]. There are several biclustering algorithms in the literature, in particular, to deal with biological data, including detection of gene expression [12, 90], protein interaction [25], and microarray data analysis [56].

To study such enriched literature, we categorized the biclustering algorithms into two main classes: *heuristic-based* and *metaheuristic-based* approaches. This categorization builds upon previous taxonomies proposed by Madeira and Oliveira [72] and by Seridi et al. [102]. Figure 4 illustrates the employed taxonomy of biclustering algorithms, where the metaheuristic-based approaches are the primary focus of this review work and is detailed in the following sections. Regarding the heuristic-based algorithms, Madeira and Oliveira proposed to divide these approaches into five classes [72], their main advantages and shortcomings are listed in Table 1:

Iterative row and column clustering combination. To identify biclusters, classical clustering methods are applied to the rows and columns of the data matrix separately; then, the resulting clusters are combined using an iterative procedure to obtain biclusters of good quality. Examples of such algorithms are the *Coupled Two-Way Clustering* (CTWC) [36], the *Interrelated Two-Way Clustering* (ITWC) [13], and the *Double Conjugated Clustering* (DCC) algorithm [72].

Divide-and-conquer. These approaches split the problem into several smaller sub-problems of the same type, solving each one recursively. Then, the solutions of the sub-problems are combined to obtain a single solution to the original problem. Divide-and-conquer biclustering algorithms start with the entire data matrix as the initial bicluster. Then, this bicluster is split into several

Table 1. Main Benefits and Shortcomings of Different Classes of Classical Biclustering Approaches

Class Name	Approaches	Benefits	Shortcomings
Iterative row and column clustering combination	CTWC [36], ITWC [13], DCC [72]	<i>The simplest way to identify biclusters with traditional clustering algorithms</i>	<i>Performance depends on meeting certain criteria such as stability and bicluster size</i>
Divide-and-conquer	DCA [43], Bi-max [92]	<i>Are usually fast as recursively break the biclustering problem into more subproblems</i>	<i>Are likely to miss good biclusters that might be split before their identification</i>
Greedy iterative search	CCA [12], OPSM [6], QUBIC [61], LAS [104]	<i>Are usually fast, as creates biclusters by iteratively adding/removing rows/columns</i>	<i>Performance depends on bicluster metrics and may find suboptimal solutions</i>
Exhaustive bicluster enumeration	SAMBA [112], BiBit [98], DeBi [103]	<i>Find the “best” biclusters in the data matrix</i>	<i>Tend to be very complex and can only be run under bicluster size constraints</i>
Distribution parameter identification	BS [56], BBC [40], FABIA [44]	<i>Allows learning joint distributions and dependencies in the input data</i>	<i>Estimating each parameter can be difficult and time-consuming</i>

biclusters iteratively until satisfying a certain termination criterion. These approaches are quite fast, but good biclusters may be split before they can be identified. The main algorithms in this class are the *Direct Clustering Algorithm* (DCA) [43] and the *Binary Inclusion-Maximal Biclustering Algorithm* (Bimax) [92].

Greedy iterative search. These methods create biclusters by adding or removing rows and columns using a quality criterion that maximizes the local gain. Therefore, although these approaches may make wrong decisions and miss good biclusters, they usually tend to be very fast. The most representative work in this category is *Cheng and Church’s Algorithm* (CCA) [12]. Other approaches are the *Order-Preserving Submatrix* (OPSM) [6], the *QUalitative BIClustering* (QUBIC) [61], and the *Large Average Submatrices* (LAS) algorithm [104].

Exhaustive bicluster enumeration. These algorithms consider that the best submatrices can only be identified by generating all the possible row and column combinations of the data matrix. Therefore, an exhaustive enumeration of all possible biclusters in the matrix is performed. These approaches find the best biclusters (if they exist) but are suitable only for very small datasets. The main drawback is their high complexity, requiring restrictions on the size of the biclusters when performing the exhaustive enumeration. The main algorithms in this category are the *Statistical-Algorithmic Method for Bicluster Analysis* (SAMBA) [112], *Bit-Pattern Biclustering Algorithm* (BiBit) [98], and the *Differentially Expressed Biclusters* (DeBi) [103].

Distribution parameter identification. These approaches assume a given statistical model and try to identify the distribution parameters used to generate the data by optimizing a particular quality criterion. Some of the algorithms adopting this approach are *Spectral Biclustering* (SB) [56], the *Bayesian BiClustering* (BBC) [40], and the *Factor Analysis for Bicluster Acquisition* (FABIA) algorithm [44].

2.3 Biclustering as an Optimization Problem

As mentioned previously, a bicluster $B(I, J)$ associated with a data matrix $X(R, C)$ is a submatrix such that $I \subseteq R$ and $J \subseteq C$, where R is a set of patterns (rows), and C is a set of attributes (columns). The biclustering problem aims to extract biclusters of a maximal size that satisfy a coherence constraint. This task of extracting bicluster from a data matrix can be seen as a combinatorial optimization problem [22, 23].

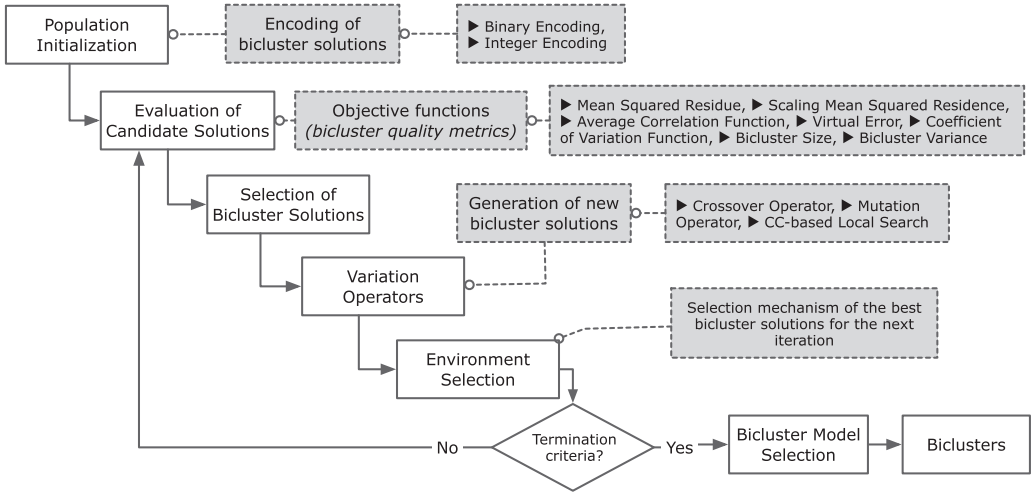


Fig. 5. Main components of biclustering algorithms based on metaheuristics.

Designing a bicluster is equivalent to jointly selecting a subset of rows and a subset of columns from an input data matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$, where N denotes the number of rows and M denotes the number of columns. Let us assume that there is no restriction on the number of rows and columns and no constraints about the nature of the biclusters (i.e., bicluster type and structure). Then, by nature, the biclustering task is a combinatorial problem with a search space size of $O(2^N \times 2^M) = O(2^{N+M})$.

However, identifying interesting biclusters is a complex task that requires defining some quality criteria to be optimized. Such criteria can measure the similarity within a bicluster, coherence, and dissimilarity (when a set of biclusters is searched). These biclustering quality criteria can be used as an objective function in a combinatorial optimization context, either alone or multiple, into a multi-objective perspective. In the second scenario, several complementary quality criteria are optimized simultaneously.

In the general case, Cheng and Church showed that the problem of finding significant biclusters is NP-hard [12], giving rise to numerous heuristic and metaheuristic approaches. These approaches do not guarantee the optimality of their solutions; however, their exploratory capabilities allow them to find suitable solutions in a reasonable computational time. The following section is dedicated to the review of metaheuristics designed for biclustering problems.

3 MAIN COMPONENTS OF METAHEURISTIC BICLUSTERING ALGORITHMS

This section presents the main components involved when designing a metaheuristic to address the biclustering problem. First, a metaheuristic requires a suitable representation of biclusters, which is directly related to the objective function (i.e., bicluster quality measures) to be optimized. The main steps in population-based metaheuristics are initializing the population of candidate solutions, evaluating solutions, applying variation operators, and selecting the best biclustering solutions. This iterative process is repeated until a particular termination criterion is satisfied. Figure 5 illustrates these main components, where the main types of bicluster encodings, the different bicluster quality measures, and the variation operators to generate new bicluster solutions are described below in this section. Other specific components, including the initialization procedures and selection strategies, are presented in detail when describing the single- or multi-objective biclustering algorithms in Sections 4 and 5.

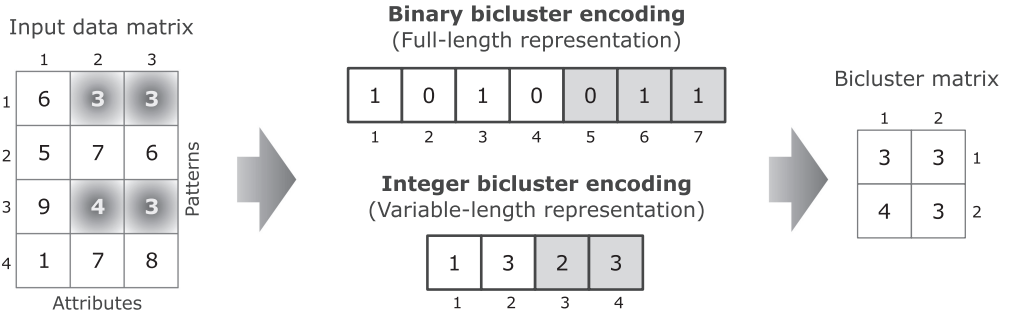


Fig. 6. Exemplification of two types of bicluster encodings in metaheuristic-based biclustering algorithms. The bicluster elements are shaded in the input data matrix, while the pattern and attribute indices are represented by white and gray boxes, respectively.

3.1 Bicluster Encoding

Metaheuristics require a representation or an encoding of potential solutions to the optimization problem. The encoding scheme is directly related to the objective function to be optimized and the recombination operators used to generate new solutions. Therefore, encoding schemes play a relevant role in the efficiency and effectiveness of any metaheuristic and constitute an essential step in its design [111].

In the bicluster analysis literature, different encoding schemes are commonly used in metaheuristics to represent bicluster solutions: *binary encoding* and *integer encoding*. These bicluster encodings are illustrated in Figure 6 by considering a 4×3 didactic data matrix and are described separately below.

Binary bicluster encoding (BBE) [8]. A bicluster $B(I, J)$ is represented as a binary vector of fixed-length size $(N + M)$: $x = \{r_1, \dots, r_N, c_1, \dots, c_M\}$, where the first N positions are related to the number of patterns (rows) and the remaining M positions to the number of attributes (columns) from the data matrix X . If the i th pattern or the j th attribute in X belongs to the bicluster $B(I, J)$, then $r_i = 1$ or $c_j = 1$ for $1 \leq i \leq N$ and $1 \leq j \leq M$; otherwise, $r_i = 0$ or $c_j = 0$. Figure 6 illustrates the binary encoding for $N = 4$ and $M = 3$ when $x = \{1, 0, 1, 0, 0, 1, 1\}$.

Integer bicluster encoding (IBE) [19]. A bicluster $B(I, J)$ is represented as an integer vector of variable-length size $N_r + M_c$: $x = \{r_1, \dots, r_{N_p}, c_1, \dots, c_{M_s}\}$, where the first N_r positions are ordered pattern indices, whereas the last M_c positions correspond to ordered attribute indices from the data matrix X . Each i th pattern position takes an integer value in the set $\{1, \dots, N\}$, and each j th attribute position takes a value in $\{1, \dots, M\}$, where N and M are the numbers of patterns and attributes in the data matrix, respectively. Figure 6 illustrates the integer encoding for $N_r = 2$ and $M_c = 2$ when $x = \{1, 3, 2, 3\}$.

Overall, the BBE encoding is a practical representation, but it requires exploring all patterns and attributes of each bicluster. On the contrary, the IBE encoding requires less computation time and memory space, as it depends on the number of ordered patterns and attributes for a particular bicluster [3, 19, 101]. Therefore, the integer bicluster encoding is more efficient in terms of time and space on large-scale datasets; however, it is more impractical when dealing with variable-length solutions in population-based metaheuristics.

3.2 Objective Functions

The development of both an effective biclustering algorithm and an appropriate bicluster quality measure (bicluster criteria) is essential to guide the search process. In particular, the quality

Table 2. Main Characteristics of Bicluster Metrics Used in Metaheuristic Bicluster Algorithms According to their Suitability for Finding Different Types of Biclusters Defined in Section 2.1

Metric Name	Metric ID	Range	Benefits	Shortcomings
Bicluster Size	bSIZE \uparrow	[1, + inf]	+ Find larger size biclusters	- Should be used with other metrics
Bicluster Variance [43]	VAR \uparrow	[0, + inf]	+ Discover constant biclusters	- Bicluster overlap is not allowed
Bicluster Row Variance [89]	rVAR \downarrow	[0, + inf]	+ Find larger row biclusters where row values vary across columns	- Should be used with other metrics
Mean Square Residue [12]	MSR \downarrow	[0, + inf]	+ Capture shifting patterns	- Unable for scaling patterns and coherent evolutions
Scaling Mean Square Residue [76]	SMSR \downarrow	[0, + inf]	+ Recognize scaling patterns	- Unable for shifting patterns
Average Correlation Function [78]	ACF \uparrow	[0, +1]	+ Find shifting & scaling patterns	- Assume linear dependency in rows
Average Correlation Value [113]	ACV \uparrow	[0, +1]	+ Capture shifting & scaling patterns	- Assume linear dependency in rows
Virtual Error [28]	VE \downarrow	[0, + inf]	+ Capture shifting OR scaling patterns	- Sensitive to outliers
Coefficient of Variation Function [68]	CVF \downarrow	[0, + inf]	+ Find constant value biclusters	- Should be used with other metrics

measure aims to recognize different types of patterns in the input data matrix by quantifying the quality of a bicluster. Several bicluster metrics have been proposed in the literature to discover different types of biclusters. This section presents some of the most well-known evaluation measures that have been used as objective functions in different metaheuristics.

For convenience, consider the following common notation for the interpretation of the different objective functions. Given a bicluster $B(I, J)$ with $|I|$ patterns (rows), $|J|$ attributes (columns), the average b_{iJ} of the i th row, the average b_{Ij} of j th column, and the average value b_{IJ} of the bicluster $B(I, J)$ are represented, respectively, as follows:

$$b_{iJ} = \frac{1}{|J|} \sum_{j \in J} b_{ij}, \quad b_{Ij} = \frac{1}{|I|} \sum_{i \in I} b_{ij}, \quad b_{IJ} = \frac{1}{|I| \times |J|} \sum_{i \in I, j \in J} b_{ij}. \quad (1)$$

Additionally, the following notation is used, an abbreviation of the bicluster evaluation measure followed by an arrow to denote if the function is maximized (\uparrow) or minimized (\downarrow), indicating the bicluster solution's quality. The main characteristics of the biclustering metrics considered here are summarized in Table 2.

Bicluster Size (bSIZE). This function represents the size of a bicluster, where α is a constant representing a preference towards the maximization of the number of rows or columns:

$$\text{bSIZE}(B)\uparrow = \alpha \frac{|I|}{|R|} + (1 - \alpha) \frac{|J|}{|C|}. \quad (2)$$

Sometimes this measure is referred to as the **Bicluster volume (VOL)**, i.e., as the number of elements b_{ij} in B , $|I| \times |J|$.

Bicluster Variance (VAR). Hartigan [43] proposed the variance measure to identify biclusters with constant values:

$$\text{VAR}(B)\downarrow = \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} (b_{ij} - b_{IJ})^2. \quad (3)$$

Existing biclustering approaches based on metaheuristics use modifications of VAR to deal with the identification of more complex types of biclusters. In practice, the **row variance (rVAR)** is

used to avoid trivial or constant-value biclusters (i.e., biclusters with significant rows variances). The rVAR measure is defined as [12, 26]:

$$\text{rVAR}(B)_{\uparrow} = \frac{1}{|I| \times |J|} \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} (b_{ij} - b_{iJ})^2. \quad (4)$$

Mean Squared Residue (MSR). Cheng and Chung [12] proposed this quality metric for evaluating the coherence of the columns and rows of a bicluster. MSR is defined as follows:

$$\text{MSR}(B)_{\downarrow} = \frac{1}{|I| \times |J|} \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} (b_{ij} - b_{iJ} - b_{iJ} + b_{IJ})^2. \quad (5)$$

The lower the MSR, the stronger the coherence exhibited by the bicluster and the better the quality. If a bicluster has an MSR value lower than a given threshold δ , then it is called a δ -bicluster.

Scaling Mean Squared Residue (SMSR). Mukhopadhyay et al. [76] developed this measure to recognize scaling patterns in biclusters. The SMSR measure is defined as:

$$\text{SMSR}(B)_{\downarrow} = \frac{1}{|I| \times |J|} \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} \frac{(b_{ij} - b_{iJ} - b_{iJ} + b_{IJ})^2}{b_{iJ}^2 \times b_{iJ}^2}. \quad (6)$$

Average Correlation Function (ACF). Nepomuceno et al. [78] proposed the ACF to evaluate the correlation between patterns in a bicluster. It is defined as:

$$\text{ACF}(B)_{\uparrow} = \frac{2}{|I| (|I| - 1)} \sum_{i=1}^{|I|} \sum_{j=i+1}^{|I|} \left| \frac{\text{cov}(p_i, p_j)}{\sigma_{p_i} \sigma_{p_j}} \right|, \quad (7)$$

where $\text{cov}(p_i, p_j) = 1/|J| \sum_{k=1}^{|J|} (b_{ik} - b_{iJ})(b_{jk} - b_{jJ})$ represents the covariance of the rows corresponding to patterns p_i and p_j , and σ_{p_i} (respectively, σ_{p_j}) represents the standard deviations of the rows corresponding to patterns p_i and p_j . The ACF measure generates values in the $[-1, 1]$ range, where values close to the unity represent that the patterns in B are highly correlated.

Average Correlation Value (ACV). Teng and Chan [113] proposed this measure to evaluate the correlation homogeneity of a bicluster. The ACV is defined as:

$$\text{ACV}(B)_{\uparrow} = \max \left\{ \frac{1}{|I| (|I| - 1)} \sum_{i,j \in I, i \neq j} r_{ij}, \frac{1}{|J| (|J| - 1)} \sum_{i,j \in J, i \neq j} r'_{ij} \right\}, \quad (8)$$

where r_{ij} or r'_{ij} is a Pearson correlation coefficient between the i th row and j th column. ACV generates values in the interval $[0, 1]$, where a value closer to 1 indicates that the rows or columns in the bicluster are highly co-expressed, whereas a low ACV value means the opposite.

Virtual Error (VE). Divina et al. [28] proposed the VE function to identify shifting or scaling patterns in biclusters. It is defined as follows:

$$\text{VE}(B)_{\downarrow} = \frac{1}{|I| \times |J|} \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} |\hat{b}_{ij} - \hat{\rho}_i|, \quad (9)$$

where $\rho_i = 1/|J| \sum_{j=1}^{|J|} b_{ij}$ represents a *virtual pattern* from B , and \hat{b}_{ij} (respectively, $\hat{\rho}_i$) is the *standardized value* of the element b_{ij} . The standardized values of a vector $V = \{v_1, \dots, v_n\}$, denoted as \hat{V} , is the set $\hat{V} = \{\hat{v}_1, \dots, \hat{v}_n\}$ with $\hat{v}_k = (v_k - \mu_V) / \sigma_V$ for $1 \leq k \leq n$, where μ_V and σ_V are

the mean and standard deviation of V . A small value of VE represents a high similarity among the patterns in the biclusters.

Coefficient of Variation Function (CVF). Maatouk et al. [68] presented the CVF to characterize the variability of a bicluster. The function is defined as

$$CVF(B)_\downarrow = \frac{\sigma_B}{b_{IJ}}, \quad (10)$$

where σ_B represents the standard deviation of the bicluster, and b_{IJ} denotes the average of all the values in the bicluster B . A high value of CVF indicates that the bicluster presents a high level of dispersion.

3.3 Variation Operators

In metaheuristic algorithms, variation operators play an important role in providing a good compromise between exploration and diversification to find high-quality solutions. Given the definitions of the different bicluster encoding types (binary and integer), we describe some crossover and mutation operators commonly used in metaheuristic-based biclustering algorithms, such as *genetic algorithms*, *differential evolution*, and *particle swarm optimization*.

Crossover operators. This operator is inspired by biological systems, where crossover is a process that occurs between pairs of chromosomes that align, split into specific parts, and then exchange fragments with each other. In biclustering algorithms, crossover is simulated by swapping segments of positions between two candidate bicluster solutions. Thus, this operator tries to combine pairs of bicluster solutions (e.g., *vectors*, *individuals*, *agents*, *particles*) to produce new offspring that inherit the characteristics of the parent solutions. Let us consider two bicluster solutions $P_1 = \{r_1, \dots, r_n, c_1, \dots, c_m\}$ and $P_2 = \{r'_1, \dots, r'_l, c'_1, \dots, c'_k\}$, where $r_n \leq r'_l$:

- *Single-point crossover* [101, 102]. The crossover is performed in each bicluster rows and columns fragment. For the rows fragment, the crossover point λ_1 in P_1 is generated randomly in the $r_1 < \lambda_1 \leq r_n$ range. Similarly, the random point λ_2 in P_2 , $\lambda_2 = r'_j$, where $r'_j \geq \lambda_1$ and $r'_{j-1} \leq \lambda_1$. The crossover in the columns fragment is performed similarly to the rows fragment.
- *Two-point crossover* [26]. It works the same way as the one-point crossover, but it selects two points inside for each bicluster fragment, two points for the row fragment, and another two points for the column part.
- *Uniform crossover* [26]. It combines the bicluster positions (bits) that are uniformly sampled from each part of the bicluster solution.
- *Bicluster crossover operator* [69]. This operator is specifically designed for bicluster approaches based on the following four steps: First, the creation of the *merge bicluster* step consists of merging the row sets I_1 and I_2 (the column sets J_1 and J_2) of the two parent biclusters B_1 and B_2 into a single set I (respectively, J) to form a new bicluster $B_{\text{merge}}(I, J)$. Second, a discretized bicluster B_{discrete} is created by clustering the columns of the $B_{\text{merge}}(I, J)$ bicluster based on the standard deviation of their rows. The third step consists in building a variation matrix from the B_{discrete} bicluster, which presents the column clusters for each row in B_{merge} . Finally, to extract biologically relevant biclusters, this step aims to extract child biclusters whose rows follow the same pattern under a set of columns. These new biclusters are determined by browsing the variation matrix and selecting the rows with the same value under the same row pairs (see further details on this crossover operator in Reference [69]).

Mutation operators. It is considered a secondary operator in several metaheuristics and is used less frequently than the crossover. In practice, it is recommended to use a low mutation percentage or to start with a high percentage at the beginning of the search and then decrease it to favor the performance of the metaheuristic. Hence, mutation operators aim to modify a bicluster solution randomly or according to a specific heuristic strategy. Some examples are:

- *Random mutation operator* [69]. Let us consider an individual in the population $B = \{r_1, \dots, r_n, c_1, \dots, c_m\}$, two mutation points r_i and c_j are generated corresponding to the rows and columns parts, respectively, such that $r_1 < r_i \leq r_n$ and $c_1 < c_j \leq c_m$. Then, a random row r'_i and column c'_j values are generated to replace the chosen positions r_i and c_j , respectively.
- *Multiple-node deletion.* This operator is based on the CC algorithm [12] and refers to the step of removing one or more elements from the candidate bicluster. First, compute b_{ij} , b_{Ij} , b_{IJ} , and $\text{MSR}(I, J)$. Next, if $1/|I| \sum_{j \in J} (b_{ij} - b_{Ij} - b_{IJ} + b_{JJ})^2 > \gamma \times \text{MSR}(I, J)$, then remove the rows $i \in I$. Second, recompute b_{ij} , b_{Ij} , b_{IJ} and $\text{MSR}(I, J)$. Then, if $1/|I| \sum_{i \in I} (b_{ij} - b_{iJ} - b_{IJ} + b_{JJ})^2 > \gamma \times \text{MSR}(I, J)$, then remove the columns $j \in J$.
- *Single-node deletion.* This operator is based on the CC algorithm [12] and refers to the step of removing one or more rows or columns from the candidate bicluster. While $\text{MSR}(I, J) > \delta$, recompute b_{ij} , b_{Ij} , b_{IJ} , and $\text{MSR}(I, J)$. Next, find the node p (row or column) with the largest squared residue; then, delete the node p . Finally, return the updated bicluster when the while condition is satisfied.
- *Multiple-node addition.* This operator is based on the CC algorithm [12] and refers to the step of adding one or more elements to the candidate bicluster. First, compute b_{ij} , b_{Ij} , b_{IJ} , and $\text{MSR}(I, J)$. Then, if $1/|I| \sum_{i \in I} (b_{ij} - b_{iJ} - b_{IJ} + b_{JJ})^2 \leq \text{MSR}(I, J)$, then add the columns $j \notin J$. Second, recompute b_{ij} , b_{Ij} , b_{IJ} , and $\text{MSR}(I, J)$. Next, if $1/|I| \sum_{j \in J} (b_{ij} - b_{iJ} - b_{IJ} + b_{JJ})^2 \leq \text{MSR}(I, J)$, then add the rows $i \notin I$.
- *CC-based mutation operator* [101, 102]. This mutation strategy is based on the CC algorithm [12], which aims to generate β biclusters. Given a bicluster solution, its rows or columns with MSR values above (or below) a certain threshold are eliminated (or added) applying the following conditions, where a “node” refers to a row or a column: (i) multiple node deletion, (ii) single node deletion, and (iii) multiple node addition.
- *Correlation-based bicluster mutation* [69]. This mutation method tries to improve the coherence of each row in a bicluster using an average correlation function. Thus, based on a correlation matrix, the number of rows coherent with each row in each bicluster is calculated. If the less coherent row is coherent with a number of rows less than a correlation threshold, then the mutation operator replaces it with the most coherent row in the bicluster. Otherwise, it adds the most coherent row to the biclusters.

4 SINGLE-OBJECTIVE BICLUSTERING APPROACHES

This section presents a review of single-objective metaheuristics for the biclustering problem. These approaches are mostly population-based metaheuristics that iteratively attempt to improve a population of biclustering solutions. First, the population is usually initialized randomly. Then, a new population of potential solutions is generated, which could be integrated into the current one by using some selection criteria. Finally, the search process stops when a given condition is satisfied (see Figure 5).

Table 3 summarizes relevant details of 25 single-objective biclustering algorithms based on metaheuristics. These algorithms, including hybrid biclustering metaheuristic approaches, are described in more detail below in this section.

Table 3. List of Single-objective Biclustering Algorithms Based on Metaheuristics

Year	Algorithm	Metaheuristic	Criterion*	Encoding	Parameters†	Application
2004	HEA [8]	GA	MSR	BBE	$N_g, N_p, R_m, R_c, \delta, \alpha$	Gene expression (2)
2006	SAB [9]	SA	MSR	Other	$T, R_{cool}, N_{mr}, N_{mc}, \delta$	Gene expression (3)
2006	SEBI [26]	GA	bSIZE, MSR, rVAR	BBE	$N_g, N_p, R_m, R_c, R_e, w_r, w_c, \delta$	Gene expression (2)
2007	BIC-aiNET [19]	AIS	MSR	IBE	$N_g, N_p, w_r, w_c, \epsilon, \delta$	Text mining (1)
2009	BiHEA [34]	GA	MSR	BBE	$N_g, N_p, R_m, \theta, \delta, \alpha$	Gene expression (1)
2009	SS&GA [79]	GA + SS (H)	MSR	BBE	$N_g, N_p, N_e, R_c, w_r, w_c$	Gene expression (2)
2010	HEAB [78]	GA + SS (H)	ACF	BBE	$N_g, N_p, N_e, R_c, w_r, w_c, \beta$	Gene expression (1)
2011	SSB [80]	SS	ACF	BBE	$N_g, N_p, N_e, R_c, w_r, w_c, \beta$	Gene expression (3)
2011	BPSO [95]	PSO	ACV	BBE	$N_g, N_p, w, c_1, c_2, \delta, \beta$	Text mining (1)
2012	CBEB [94]	GA	MSR	BBE	$N_g, N_p, N_s, D_t, \delta, \beta$	Gene expression (3)
2012	EvoBic [3]	GA	bSIZE, MSR, ACF	IBE	$N_g, N_p, R_m, R_c, \delta, \beta$	Gene expression (2)
2012	PSO-SA-BIC [114]	PSO + SA (H)	ACV	BBE	$N_g, N_p, w, c_1, c_2, T, R_{cool}, \delta, \beta$	Gene expression (3)
2013	Evo-Bexpa [89]	GA	VE, VOL, rVAR, <i>Overlap</i>	BBE	$N_g, N_p, R_m, R_c, \beta, w_r, w_c, w_o, w_v$	Gene expression (4)
2014	EBACross [68]	GA	bSIZE, MSR, ACF, CVF	BBE	$N_g, N_p, R_m, R_c, \delta, \beta, w_r, w_c, th$	Gene expression (2)
2014	TriGen [42]	GA	MSR _{time} , LSL _{time}	IBE	$N_g, N_p, R_m, R_c, w_r, w_c, w_t, \delta, \beta$	Gene expression (1)
2015	COCSB [67]	CS	MSR	BBE	$N_g, N_p, N_{ls}, N_{cs}, \delta, \beta$	Gene expression (6)
2015	SSB-Bio [82]	SS	bSIZE, ACF, ACF _{bio}	BBE	$N_g, N_p, N_e, w_1, w_2, w_{rc}, \beta$	Gene expression (2)
2015	BISS [81]	SS	ACF	BBE	$N_g, N_p, N_e, R_c, w_r, w_c, \rho, \beta$	Gene expression (3)
2018	BISS-go [83]	SS	ACF _{go}	BBE	$F_g, N_g, N_p, N_e, R_c, w_r, w_c, \rho, \beta$	Gene expression (3)
2018	GACSB [125]	GA + CS (H)	MRS, VE, ACV	BBE	$N_g, N_p, R_m, R_c, R_e, w, \delta$	Gene expression (3)
2018	EBA [69]	GA	bSIZE, MSR, ACF, CVF	BBE	$N_g, N_p, R_m, R_c, N_{mr}, N_{mc}, \delta, \alpha$	Gene expression (3)
2019	BP-EBA [47]	GA	bSIZE, MSR, SMSR	BBE	$N_g, N_p, R_m, R_c, R_e, \delta$	Gene expression (3)
2019	EBIC [86]	GA + <i>Tabu list</i>	<i>logarithm-based criterion</i>	IBE	$N_g, N_p, R_m, R_c, R_e, v, w_1$	Gene expression (8)
2021	HPSO-Tric [77]	PSO + SA (H)	ACF _{time}	BBE	$N_g, N_p, w, c_1, c_2, T, R_{cool}, \delta, \beta$	Gene expression (1)
2021	ELSA [70]	GA	ACF _{stat} , ACF _{bio}	BBE	$N_g, N_p, R_m, R_c, R_e, \delta, \alpha$	Gene expression (2)
2021	TriCS [110]	CS	MSR _{time}	BBE	$N_g, N_p, R_c, N_{nest}, \lambda, \delta, \alpha$	Gene expression (1)

“Metaheuristic” indicates the type of metaheuristic algorithm, which can be Simulated Annealing (SA), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Scatter Search (SS), Cuckoo Search (CS), Artificial Immune System (AIS). “Criterion” refers to the bicluster evaluation metric as defined in Section 3.2; “Encoding” indicates the type of bicluster encoding binary (BBE) or integer (IBE) as described in Section 3.1; “Parameters” refers to the input parameters required by the metaheuristic, which are defined at the bottom of the table; and “Application” indicates a real-world application used by the corresponding algorithm, the number of datasets is given in parentheses.

*: Biclustering criteria are combined into a single objective function. †: Number of iterations (N_g); population size of individuals, cells, or particles (N_p); external population’s size (N_e); CS nest size (N_{nest}); mutation rate (R_m); crossover rate (R_c); elitism rate (R_e); number of biclusters (β); minimum rows (N_{mr}); minimum columns (N_{mc}); rows weight (w_r); columns weight (w_c); rows-columns weight (w_{rc}); objective function weights (w_i); mean squared residue score threshold (δ); multiple node deletion frequency (α); overlap degree (θ); minimum correlation value (ρ); AIS suppression threshold (ϵ); CBEB number of subspaces (N_s); SA temperature (T); SA cooling rate (R_{cool}); PSO inertia weight (w); PSO parameters (c_1, c_2); distance threshold (D_t); Gene annotation file (F_g).

4.1 Simulated Annealing

Simulated annealing (SA) is a probabilistic method proposed by Kirkpatrick et al. [55] to find the global minimum of a cost function. SA emulates the physical process whereby a melted solid material (initial state) is gradually cooled until the minimum energy state is reached, when the material structure is “frozen.” SA is a single-solution-based metaheuristic that improves a single-point solution, evaluated by a single criterion function, and could be viewed as search trajectories through the search space [49].

Simulated annealing has been used to address the biclustering problem. Brayan et al. [9] proposed an SA-based biclustering approach called **SAB**. In SAB, each solution’s fitness function is computed using the MSR criterion, and the algorithm is run k times to obtain k biclusters. To avoid overlap among biclusters, in the SAB algorithm, the discovered biclusters are masked in the original data. This strategy is similar to the Cheng and Church (CC) [12], where the original values are replaced with random ones to prevent them from being part of any further bicluster.

4.2 Genetic Algorithms

Genetic algorithm (GA), developed by Holland in the early 1970s [45], emulates the principle of evolution by natural selection stated by Charles Darwin. Several biclustering approaches have

been proposed based on GAs, which sometimes are referred to as evolutionary algorithms (EAs). Next, we summarize these approaches.

Bleuler et al. [8] proposed the first evolutionary biclustering algorithm, **HEA**, in 2004. This algorithm uses a fixed-length binary encoding, and the MSR criterion is used as the fitness function. In addition, a bit mutation and uniform crossover are used as variation operators to generate new biclustering solutions during the evolutionary process. In HEA, a diversity maintenance strategy is considered, which decreases the overlapping level among the bicluster, and the CC algorithm [12] is also applied as a local search method to increase the size of the biclusters. In the end, the entire population of individuals is returned as the set of resulting biclusters. Gallo et al. [34] proposed the **BiHEA** algorithm, similar to Bleuler's approach, as both perform a local search based on the CC algorithm and use the same objective function. However, the approaches differ in the crossover operators (BiHEA uses a two-point crossover). In addition, BiHEA considers an external archive to keep the best generated biclusters throughout the evolutionary process.

Divina and Aguilar-Ruiz [26] presented a sequential evolutionary biclustering (SEBI) algorithm. The term sequential refers to that the evolutionary algorithm generates only one bicluster per run; thus, to generate several biclusters, SEBI needs to be invoked iteratively. Furthermore, a general matrix of weights is considered to control the overlapping among biclusters. In SEBI, three crossover and mutation strategies are used with equal probability of reproduction: one-point, two-point, and uniform crossovers, and mutations that add a row or a column to the bicluster or the standard mutation. The evaluation of individuals is carried out by an objective function that involves three different criteria: MSR, bicluster size, and row variance.

Huang et al. [94] proposed a biclustering approach based on genetic algorithms and hierarchical clustering called **CBEb**. First, the rows of the data matrix (conditions) are separated into a number of condition subsets (subspaces). Next, the genetic algorithm is applied to each subspace in parallel. Then, an expanding-merging strategy is employed to combine the subspace results into output biclusters. In CBEb, the MSR metric is used as an objective function, whereas a simple crossover and a binary mutation are used to reproduce new solutions. Although this approach outperforms several traditional biclustering algorithms, it requires a longer computation time than the other methods. This disadvantage of CBEb is mainly due to its utilization process and separation method for creating and evaluating several subspaces.

An evolutionary biclustering (EvoBic) algorithm with a variable-length representation was proposed by Ayadi et al. in Reference [3]. This integer encoding represents the individuals as a string composed of ordered genes and conditions indices, reducing time and memory space [19]. EvoBic algorithm considers three different biclustering metrics (bSIZE, MSR, ACF), a single-point crossover, and a standard mutation to generate new biclusters.

Another evolutionary biclustering algorithm, called **Evo-Bexpa**, was presented by Pontes et al. [89]. This algorithm allows identifying types of biclusters in terms of different objectives. These objectives have been put together by considering a single aggregative objective function. Evo-Bexpa bases the bicluster evaluation on the use of expression patterns, recognizing both shifting and scaling patterns by considering the VE, Vol, Overlap, and Var quality metrics.

Maatouk et al. [68] proposed an evolutionary biclustering algorithm named **EBACross**. First, the initialization of the initial population is based on the CC algorithm [12]. Then, the evaluation and selection of the individuals are based on four complementary biclusters metrics: bicluster size (bSIZE), MSR metric, average correlation (ACF), and the coefficient of variation function (CVF). In EBACross, a binary encoding of fixed length, a crossover method based on the standard deviation of the biclusters, and a mutation strategy based on the biclusters' coherence are considered. Later, the same authors proposed a generic evolutionary biclustering algorithm (EBA) [69]. The authors analyzed the EBA's performance by varying its genetic components in this work. In

the study, they considered three different biclustering metrics (bSIZE, MSR, and ACF), two selection operators (parallel and aggregation methods), two crossover methods (random-order and biclustering methods), and two mutation operators (random and biclustering strategies). Hence, several versions of the EBA algorithm were introduced. In terms of statistical and biological significance, the clustering performance showed that the EBA configuration based on a selection with aggregation and biclustering crossover and mutation operators performed better for several microarray data. Recently, also Maatouk et al. [70] proposed the **ELSA** algorithm, an evolutionary algorithm based on a local search method that integrates biological information in the search process. The authors stated that statistical criteria are reflected by the size of the biclusters and the correlation between their genes, while the biological criterion is based on their biological relevance and functional enrichment degree. Thus, the ELSA algorithm evaluates the statistical and biological quality of the biclusters separately by using two objective functions based on the average correlation metric (ECF). Furthermore, to preserve the best biclusters over different generations, an archiving strategy is used in ELSA.

Huang et al. [47] proposed a bi-phase evolutionary biclustering algorithm (BP-EBA). The first phase is dedicated to the evolution of rows and columns, and the other is for the identification of biclusters. The interaction of the two phases guides the algorithm toward feasible search directions and accelerates its convergence. BP-EBA uses a binary encoding, while the population is initialized using a hierarchical clustering strategy to discover bicluster seeds. The following biclustering metrics were employed to evaluate the individuals in the population: MSR, SMSR, and bSIZE. Finally, the performance of this approach was compared with other biclustering algorithms using microarray datasets.

Gutierrez-Aviles [42] proposed an evolutionary algorithm, **TriGen**, to find biclusters in temporal gene expression data (known as *triclusters*). Thus, the aim is to find triclusters of gene expression that simultaneously take into account the experimental conditions and time points. Here, an individual is composed of three structures: a sequence of genes, a sequence of conditions, and a sequence of time points. Furthermore, the authors proposed specific genetic operators to generate new triclusters. Two different metrics were taken into account to evaluate the individual: MSR_{time} (modification of the MSR metric) and LSL_{time} (least-squares approximation for the points in a 3D space representing a tricluster). As a result, TriGen could extract groups of genes with similar patterns in subsets of conditions and times, and these groups have shown to be related in terms of their functional annotations extracted from the Gene Ontology.

4.3 Scatter Search

Scatter search (SS) is a population-based evolutionary metaheuristic that emphasizes systematic processes against random procedures [58]. The optimization process consists of evolving a set called *reference*, which iteratively is updated by using *combination* and *improvement* methods that exploit context knowledge. In contrast to other evolutionary approaches, SS is founded on the premise that systematic designs and methods for creating new solutions afford significant benefits beyond those derived randomly.

Nepomuceno and his collaborators have developed a series of SS-based biclustering algorithms for gene expression data [80–83]. In Reference [80], the authors presented the **SSB** algorithm to find shifting and scaling patterns biclusters, which are interesting and relevant patterns from a biological point of view. For this purpose, the average correlation function (ACF) was modified and used as the objective function. The SSB algorithm uses a binary encoding to represent biclustering solutions and includes a local search method to improve biclusters with positively correlated genes. The same authors proposed another SS-based biclustering algorithm that integrates prior biological knowledge [82]. This algorithm (herein referred to as **SSB-bio**) requires as input, in

addition to the gene expression data matrix, an annotation file that relates each gene to a set of terms from the repository *Gene Ontology* (GO). Thus, two biological measures, *FracGO* and *SimNTO*, were proposed and integrated as part of the objective function to be optimized. This fitness function is a weighted-sum function composed of three factors: the bicluster size (bSIZE), the bicluster correlation (ACF metric), and the bicluster biological relevance based on *FracGO* or *SimNTO* measures (ACF_{bio} metric).

Nepomuceno et al. [81] proposed the **BISS** algorithm to find biclusters with both shifting and scaling patterns and negatively correlated patterns. This algorithm, similar to SSB-Bio, is based on *a priori* biological information from the GO repository, particularly the categories: biological process, cellular components, and molecular function. In BISS, a fitness function involving the AFC and bSIZE metrics was used to evaluate the quality of the biclusters. Furthermore, in another recent study, Nepomuceno et al. [83] used the BISS algorithm for the biclustering of high-dimensional expression data. This algorithm, referred to as **BISS-go**, also considers the biological knowledge available in the GO repository to find biclusters composed of groups of genes that are functionally coherent. This task is accomplished by defining two GO semantic similarity measures that are integrated into the fitness function. The reported results show that the inclusion of biological information improves the performance of the biclustering process.

4.4 Cuckoo Search

Cuckoo search (CS), developed by Yang and Deb [124], is a nature-inspired metaheuristic based on the brood parasitism of some cuckoo species. In this algorithm, the exploration of the search space is enhanced using Lévy flights (Lévy distribution) instead of using simple isotropic random walks.

Lu et al. [67] introduced a CS-based biclustering algorithm for gene expression data named **COCSB**. The authors incorporated different strategies in COCSB to improve its diversity performance and convergence rate, such as the searching- and abandoning-nest operations. In CBEB, the MSR metric is used as the objective function, whereas a binary bicluster representation of the solutions is considered. This approach was compared to several classical biclustering algorithms, including CC and SEBI, obtaining a good biological significance and time computation performance.

4.5 Particle Swarm Optimization

Particle swarm optimization (PSO), introduced by Kennedy and Eberhart [54], is a population-based search method in which the individuals (referred to as particles) are grouped into a swarm. The particles explore the search space by adjusting their trajectories iteratively according to self-experience and neighboring particles [58].

Rathipriya et al. [95] proposed a PSO-based biclustering algorithm called **BPSO**. BPSO was applied on web data to find biclustering that contains relationships between web users and webpages, useful for e-commerce applications such as web advertising and marketing. The individuals were encoded using the traditional binary bicluster representation, and the average correlation value (ACV) metric was used as the fitness function. This PSO-based algorithm outperformed two traditional biclustering algorithms based on greedy search. Furthermore, the identified biclusters by BPSO covered a more considerable percentage of users and webpages, capturing the global browsing patterns from web usage data.

4.6 Hybrid Metaheuristic Approaches

It is well-known that nature-inspired metaheuristics are effective strategies for solving optimization problems. However, sometimes it is difficult to choose a metaheuristic for a particular instance

problem. In these scenarios, hybrid approaches provide flexible tools that can help to cope with this problem. In line with this, different hybrid nature-inspired metaheuristics for the biclustering problem are described below.

Nepomuceno et al. [78, 79] proposed a couple of hybrid metaheuristics for biclustering based on *scatter search* (SS) and *genetic algorithms* (GAs). In Reference [79], the authors proposed the **SS&GA** biclustering algorithm, in which the general scheme is based on SS but incorporates some of GA's features, such as the mutation and crossover operators, to generate new biclustering solutions. This algorithm uses a binary encoding to represent solutions and considers the MSR metric to evaluate the quality of the biclusters. Later, the authors proposed a similar hybrid evolutionary algorithm (herein referred to as **HEAB**) for biclustering gene expression data [78]. In HEAB, the ACF metric is used as the fitness function based on a correlation measure. First, HEAB searches for biclusters with groups of highly correlated genes; then, new biclusters with shifting and scaling patterns are created by analyzing the correlation matrix. The experimental results using the *Lymphoma dataset* indicated that the correlation-based metric outperformed the well-known MSR metric.

Recently, Yin et al. [125] proposed a hybrid biclustering approach based on *cuckoo search* (CS) and *genetic algorithms* (GAs), **GACSB**. This approach considers the CS algorithm as the main framework and uses the tournament strategy and the elite-retention strategy based on the GA to generate the next generation of solutions. In addition, GACSB uses as objective functions different metrics, namely, ACV, MSR, and VE. The experimental results obtained by GACSB were compared with several classic biclustering algorithms, such as the CC algorithm and SEBI, where GACSB outperformed these algorithms when considering various gene expression datasets.

Furthermore, hybrid biclustering approaches that combine *particle swarm optimization* (PSO) features and *simulated annealing* (SA) have been proposed in the literature [77, 114]. First, Thangavel et al. [114] proposed a PSO-SA biclustering algorithm (**PSO-SA-BIC**) to extract biclusters of gene expression data. In this approach, SA is used as a local search procedure to improve the low-performing particles' position. A modified version of the ACV metric is used to identify biclusters with shifting and scaling patterns. The experimental results showed that the PSO-SA-BIC algorithm outperformed some classical algorithms by providing statistically significant biclusters. Recently, Narmadha and Rathipriya [77] developed a hybrid approach combining PSO and SA to extract triclusters from a 3D-gene expression dataset (*Yeast Cell Cycle* data). This algorithm, named **HPSO-TriC**, uses a fitness function based on the ACF metric, which aims to identify tricluster with a high correlation degree among genes over samples and time points (this function is referred to as ACF_{time}). The HPSO-TriC algorithm was compared with a PSO-based biclustering algorithm, performing better, as the extracted tricluster was more biologically significant.

4.7 Summary of Single-objective Biclustering Approaches

We described 25 single-objective biclustering methods based on metaheuristics, including simulated annealing (SA), genetic algorithm (GA), particle swarm optimization (PSO), scatter search (SS), and cuckoo search (CS). These algorithms are summarized in Table 3, where GA-based biclustering algorithms represent 61% of the surveyed methods. Moreover, some hybrid approaches (a combination of different metaheuristics) have been proposed due to the complexity of the biclustering problem; indeed, several of the reviewed approaches often incorporate a local search strategy to better exploit the search space. Regarding the objective function, many biclustering algorithms (43%) combine information from multiple bicluster metrics in such a way that these optimization functions usually consider the homogeneity (e.g., the MSR metric) and the bicluster size (bSIZE). For measuring the bicluster homogeneity, the mean squared residence (MSR) and the average correlation function (ACF) are commonly used with percentages of 52% and 43%,

respectively. Finally, we noticed that Binary Bicluster Encoding (BBE) is the most used among the biclustering algorithms (87%), even though it is less efficient than Integer Encoding (IBE) in terms of computation time and memory space.

5 MULTI-OBJECTIVE BICLUSTERING APPROACHES

Several real-world optimization problems naturally involve multiple objectives. As the name suggests, a multi-objective optimization problem (MOOP) has a number of objective functions to be minimized or maximized. Therefore, the optimal solution for MOOPs is not a single solution but a set of solutions denoted as Pareto-optimal solutions. In this sense, a solution is Pareto-optimal if it is impossible to improve a given objective without deteriorating another. Generally, such a set of solutions represents the compromise solutions between different conflicting objectives [21].

The biclustering problem can be formulated as a combinatorial optimization problem, such that multiple bicluster quality criteria can be optimized simultaneously [72, 102]. Indeed, in gene expression data analysis, the quality of a bicluster can be defined by its size and its intra-cluster variance (coherence). However, these criteria are independent and notably in conflict, as the bicluster's coherence can constantly be improved by removing a row or a column, i.e., by reducing the bicluster's size.

This section introduces several multi-objective metaheuristics that have been proposed for the biclustering problem. Table 4 summarizes some relevant details about these types of algorithms that are introduced in this section.

5.1 Multi-objective Evolutionary Algorithms Based on NSGA-II

One of the most representative multi-objective algorithms is the non-dominated sorting genetic algorithm (NSGA-II) [21]. This Pareto-dominance algorithm is characterized by incorporating an explicit diversity-preservation mechanism. As NSGA-II has widely been used to address the biclustering problem, we will outline how the algorithm operates. In NSGA-II, the offspring population Q_t is first created by using the parent population P_t ; then, the two populations are combined to form the population R_t . Next, a non-dominated sorting method is used to classify the entire population R_t . Then, the new population P_{t+1} is filled by solutions of different non-dominated fronts, starting with the best non-dominated front, followed by the second front, and so on. Finally, when the last allowed front is being considered, there may exist more solutions than the remaining slots in the new population; in this case, a niching strategy based on the crowding distance is used to choose the members of the last front. For a more detailed description and understanding of NSGA-II, the reader is referred to Reference [21]. Next, we describe different biclustering algorithms that use NSGA-II as the underlying optimization method.

Several multi-objective biclustering approaches have been proposed based on the well-known NSGA-II algorithm [17, 27, 28, 57, 73–75, 101]. Research in multi-objective biclustering became popular after the work by Mitra and Banka [75] entitled “Multi-objective evolutionary biclustering of gene expression data,” which was published in 2006. This algorithm (referred to as **MOEAB**) uses the CC method [12] during the population's initialization and after applying the variation operators. MOEAB uses a binary encoding representation (BBE), a uniform single-point crossover, and a single-bit mutation. Concerning the objective functions, the size of the bicluster (bSIZE) and the MSR metric were considered.

Divina et al. [27, 28] have proposed some multi-objective biclustering approaches based on NSGA-II for microarray data. First, Divina and Aguilar-Ruiz [27] presented the sequential multi-objective biclustering (SMOB) algorithm for finding biclusters of high quality with large variation. SMOB adopts a sequential strategy such that the algorithm is invoked several times, each time

Table 4. List of Multi-objective Biclustering Algorithms Based on Metaheuristics

Year	Algorithm	Metaheuristic	Criteria	Encoding*	Parameters†	Application
2006	MOEAB [75]	NSGA-II	bSIZE, MSR	BBE	$N_g, N_p, R_m, R_c, R_e, \delta, \alpha$	Gene expression (2)
2007	SMOB [27]	NSGA-II	bSIZE, MSR, rVAR	BBE	$N_g, N_p, R_m, R_c, R_e, \delta, \alpha$	Gene expression (3)
2008	MOFBA [74]	NSGA-II	bSIZE, MSR, rVAR	IBE	$N_g, N_p, R_m, R_c, R_e, \alpha_r, \alpha_c, \alpha, \delta$	Gene expression (2)
2008	MOPSOB [65]	MOPSO	bSIZE, MSR, rVAR	BBE	$N_g, N_p, w, c_1, c_2, \delta, \alpha$	Gene expression (2)
2008	CMOPSOB [63]	MOPSO	bSIZE, MSR, rVAR	BBE	$N_g, N_p, w, c_1, c_2, \delta, \epsilon$	Gene expression (2)
2009	AMOPSOB [60]	MOPSO	bSIZE, MSR, rVAR	BBE	$N_g, N_p, w, c_1, c_2, \delta, \alpha$	Gene expression (2)
2009	MOACOB [64]	MOACO	bSIZE, MSR	BBE	$N_g, N_p, \lambda, \pi, \delta, \alpha$	Gene expression (2)
2009	MOM-aiNet [14]	MOAIS	bSIZE, MSR	IBE	$N_g, N_p, w_r, w_c, \epsilon, \delta, \alpha$	Gene expression (3)
2009	MOGAB [73]	NSGA-II	MSR, rVAR	IBE	$N_g, N_p, R_m, R_c, R_e, \delta, \alpha$	Gene expression (4)
2009	SPEA2B [35]	SPEA2	bSIZE, MSR, rVAR	BBE	$N_g, N_p, N_e, R_m, R_c, R_e, \delta, \alpha$	Gene expression (2)
2011	MOBI [101]	NSGA-II	bSIZE, MSR, rVAR	IBE	$N_g, N_p, R_m, R_c, R_e, \delta, \alpha$	Gene expression (2)
2012	SMOB-VE [28]	NSGA-II	bSIZE, VE, rVAR	BBE	$N_g, N_p, R_m, R_c, R_e, T_s, \delta, \alpha$	Gene expression (9)
2015	HMOBI [102]	IBEA	MSR, rVAR	IBE	$N_g, N_p, R_m, R_c, R_e, \delta, \alpha$	Gene expression (4)
2015	SPEA2B- δ [38]	SPEA2	bSIZE, MSR	BBE	$N_g, N_p, N_e, R_m, R_c, \delta, \alpha$	Gene expression (1)
2015	EMOA- δ -TRIMAX [7]	NSGA-II	MSR, bSIZE, ρ	BBE	$N_g, N_p, R_m, R_c, \delta, \alpha$	Gene expression (3)*
2016	AMOSAB [99]	AMOSAB	bSIZE, MSR	IBE	$N_g, N_p, T, R_{cool}, N_{mr}, N_{mc}, \delta$	Gene expression (2)
2017	PBD-SPEA2 [39]	SPEA2	bSIZE, MSR, rVAR	IBE*	$N_g, N_p, R_m, R_c, R_e, \delta, \alpha$	Gene expression (2)
2019	BP-NSGA2 [57]	NSGA-II	bSIZE, MSR	BBE	$N_g, N_p, R_m, R_c, R_e, \delta, \alpha$	Gene expression (1)
2019	AMOSAB-PS [1]	AMOSAB	bSIZE, MSR	IBE	$N_g, N_p, T, R_{cool}, N_{mr}, N_{mc}, \delta$	Gene expression (3)
2019	β -SMOB [29]	NSGA-II	bSIZE, MSR, rVAR	BBE	$N_g, N_p, R_m, R_c, R_e, \delta, \alpha$	Energy consumption (1)
2020	MMCo-Clus [17]	NSGA-II	MSR, rVAR, AI	BBE*	$N_g, N_p, R_m, R_c, R_e, \delta, \alpha$	Gene expression (3)
2022	BOBEA [71]	NSGA-II	bSIZE, rVAR	IBE	$N_g, N_p, R_m, R_c, R_e, \delta, \alpha$	None
2022	TSTP [108]	NSGA-II	bSIZE, ACV	BBE*	$N_g, N_p, R_m, R_c, R_e, N_{mr}, \delta, \alpha$	Gene expression (2)

“Metaheuristic” indicates the multi-objective algorithm used as the underlying optimization strategy (NSGA-II, MOPSO, MOAIS, SPEA2, IBEA, AMOSA). “Criteria” refers to the bicluster evaluation metrics used as objective functions as defined in Section 3.2; “Encoding” indicates the type of bicluster encoding, binary (BBE) or integer (IBE) as described in Section 3.1; “Parameters” refers to the input parameters required by the metaheuristic, which are defined at the bottom of the table; and “Application” indicates a real-world application used by the corresponding algorithm, the number of datasets is given in parentheses.

* Modification to the original encoding described in Section 3.1. †: Number of iterations (N_g); population size of individuals, cells, or particles (N_p); external population’s size (N_e); mutation rate (R_m); crossover rate (R_c); elitism rate (R_e); number of biclusters (β); minimum rows (N_{mr}); minimum columns (N_{mc}); rows weight (w_r); columns weight (w_c); rows-columns weight (w_{rc}); mean squared residue score threshold (δ); multiple node deletion frequency (α); minimum correlation value (ρ); AIS suppression threshold (ϵ); SA temperature (T); SA cooling rate (R_{cool}); PSO inertia weight (w); PSO parameters (c_1, c_2); Gene annotation file (F_g).

returning a bicluster stored in a temporal list. This algorithm considered a binary encoding, three different crossover operators (one-point, two-point, and uniform), and tree mutation strategies (single-bit, add-row, and add-column). The objectives considered in SMOB were: the MSR metric, the bicluster size (bSIZE), and the row variance (rVAR). Later, Divina et al. [28] presented the virtual error (VE) metric, which measures how well the genes in a bicluster follow the general tendency. Then, the VE metric was used as an additional objective function in SNOB. This modified algorithm, referred to as **SMOB-VE**, aims to find biclusters with shifting and scaling patterns using VE instead of the MSR metric.

Maulik et al. [73, 74] also presented biclustering algorithms based on the NSGA-II algorithm. In Reference [74], the authors presented a multi-objective fuzzy biclustering (MOFBA) algorithm for discovering overlapping biclusters. MOFBA simultaneously optimizes fuzzy versions of the metrics MSR, bSIZE, and rVAR. Furthermore, MOFBA uses an integer encoding of variable string length, a single-point crossover, and a uniform mutation strategy. Subsequently, the authors proposed the **MOGAB** algorithm [73], which optimizes two objective functions, MSR and rVAR. Similar to MOFBA, the MOGAB algorithm uses a variable string length encoding, a single-point crossover, and a uniform mutation strategy. Additionally, the authors presented the bicluster index (BI) to validate the obtained biclusters from microarray data.

Seridi et al. [101] proposed a multi-objective biclustering algorithm that simultaneously optimizes three conflicting objectives, the bicluster size (bSIZE), the MSR metric, and the row variance (rVAR). The presented evolutionary framework **MOBI** can integrate any evolutionary algorithm, such as NSGA-II in this case. MOBI uses an integer bicluster encoding, a single-point crossover, and the CC local-search heuristic [12] replaces the mutation operator.

Kong et al. [57] presented an interesting biclustering algorithm incorporating a bi-phase evolutionary architecture and the NSGA-II algorithm. In this algorithm, referred to as **BP-NSGA2**, the first phase consists of evolving the population of rows and columns and then the population of biclusters. The two populations are initialized using a hierarchical clustering method, and then they are evolved independently. Next, during the evolutionary process, the NSGA-II algorithm simultaneously optimizes the MSR metric and bicluster size. This multi-objective approach outperformed two traditional biclustering approaches. The same authors revisited this idea of incorporating a bi-phase evolutionary strategy for the proposal of the BP-EBA algorithm [47].

Recently, Cui et al. [17] proposed a multi-objective optimization-based multi-view co-clustering (MMCo-Clus) algorithm for feature selection of gene expression data. First, two data views are constructed using information from two different biological data sources. Next, the MMCo-Clus algorithm identifies biclusters (co-clustering solutions) considering the constructed views. Finally, a small number of non-redundant features are selected from the original feature space using consensus clustering. MMCo-Clus uses two well-known bicluster measures, the MSR metric and the bicluster size (bSIZE), and the agreement index. Although this approach focuses on feature selection, it applies an intrinsic biclustering strategy to select relevant features.

5.2 Multi-objective Evolutionary Algorithms Based on SPEA2 and IBEA

Similar to the NSGA-II algorithm, the strength Pareto evolutionary algorithm (SPEA2) [129] is a well-known multi-objective evolutionary algorithm (MOEA) in the specialized literature. This algorithm is characterized by maintaining an external population, which is used to introduce elitism. This population stores a fixed number of non-dominated solutions found during the entire evolutionary process. Additionally, SPEA2 uses these elite solutions to participate in the genetic operations along with the current population to improve the convergence and diversity of the algorithm. Below, we describe some biclustering approaches that use SPEA2 as the primary multi-objective optimization method.

Gallo et al. [35] addressed the microarray biclustering problem using different MOEAs, where the SPEA2 algorithm performed better. This approach (**SPEA2B**) considered a binary encoding, a probabilistic-based mutation, and a two-point crossover operator. Four different objectives were considered in SPEA2B: the number of genes, the number of conditions, row variance (rVAR), and the MSR metric. Additionally, a greedy method was implemented based on the CC algorithm to maintain large size and low homogeneity biclusters in the population.

Golchin et al. [38, 39] have proposed biclustering approaches based on the SPEA2 algorithm. First, Golchin et al. [38] presented a multi-objective biclustering algorithm (herein referred to as **SPEA2B- δ**), which optimizes the MSR metric and the size of the bicluster simultaneously. This algorithm used a binary bicluster encoding, a single-point crossover, and a single-bit mutation operator. SPEA2B- δ also incorporated a search heuristic strategy similar to the CC algorithm to remove unwanted genes and conditions. As SPEA2B- δ generates a set of biclustering solutions (Pareto front approximation), a fitness selection function based on the coherence and size of the biclusters is considered to choose the best solutions. Later on, Golchin and Liew [39] proposed a SPEA2-based biclustering algorithm for gene expression data named **PBD-SPEA2**. This algorithm considers three objective functions: MSR, bSIZE, and rVAR. An interesting aspect of PBD-SPEA2 is that each individual in the population represents multiple biclusters, instead of only one bicluster

as in other approaches. Thus, given a user-defined number of biclusters, k , an integer-based encoding of fixed length but extended to k biclusters is considered. Regarding the variation operators for generating new solutions, PBD-SPEA2 uses the CC heuristic as a mutation operator and a similarity-based crossover. Finally, a sequential selection technique is used to choose the final solution from the Pareto front approximations obtained by PBD-SPEA2.

Similar to NSGA-II and SPEA2 algorithms, the indicator-based multi-objective algorithm (IBEA) [128] is another representative MOEA in the evolutionary computation literature. In this regard, Seridi et al. [102] proposed a biclustering approach based on the IBEA algorithm named **HMOBI** [102]. This algorithm used an integer-based representation, a single-point crossover, and a mutation operator based on the CC algorithm. In HMOBI, three biclustering metrics are considered as objective functions: the MSR metric and the row variance, and the bicluster size. The obtained results were compared in terms of the bicluster quality and their biological relevance.

5.3 Multi-objective Particle Swarm Optimization

The particle swarm optimization (PSO) algorithm has been extended to solve multi-objective optimization problems (MOPs). Most of the existing multi-objective PSO (MOPSO) algorithms involve developments from the evolutionary computation field to address MOPs. For a review of different MOPSO algorithms, the reader is referred to the survey by Reyes-Sierra [97]. Below, we present different MOPSO algorithms for the biclustering problem.

Two multi-objective approaches to microarray biclustering based on MOPSO algorithms were presented by Junwan Liu et al. [63, 65]. Both approaches use a binary bicluster encoding and simultaneously optimize three objective functions: the bicluster size (bSIZE), the MSR metric, and the row variance (rVAR). The first algorithm, named **MOPSOB** [65], considers a relaxed form of the Pareto dominance (ϵ -dominance), whereas the second approach (**CMOPSOB** [63]) uses a Pareto-based dominance as in the NSGA-II algorithm. Additionally, in the CMOPSOB algorithm, the information of nearest neighbors between particles is considered when updating the particles' velocity, aiming to accelerate the algorithm's convergence. In the comparative analysis, the biological relevance of the biclusters obtained by CMOPSOB was analyzed considering the GO repository, showing that this approach could find biologically meaningful clusters.

Another gene expression biclustering algorithm based on a MOPSO was proposed by Lashkargir et al. [60]. The authors proposed a hybrid MOPSO for the biclustering problem (**AMOPSOB**), which uses binary encoding and optimizes four objective functions: bicluster size, row variance, and the MSR metric. The AMOPSOB algorithm includes a local search method based on the CC algorithm and, in addition to the PSO steps, three mutation operators: standard, add-row, and add-column. Additionally, this approach can find biclusters with a low level of overlap among biclusters by considering an external archive.

5.4 Other Multi-objective Approaches

This section describes a number of multi-objective approaches to biclustering that do not fit into the previous classification. The approaches described below are based on ant colony optimization (ACO), artificial immune systems (AISs), and simulated annealing (SA).

The ACO algorithm [30] is a probabilistic technique inspired by the behavior of ants in finding paths from their colony to a food source. In this regard, a multi-objective ACO algorithm for microarray biclustering (MOACOB) was introduced by Liu et al. [64]. MOACOB algorithm uses ACO concepts for biclustering microarray data, where the bicluster size and the MSR metric are optimized simultaneously. Furthermore, this algorithm uses a relaxed form of the Pareto dominance (ϵ -dominance) and considers a binary encoding to represent biclusters. In MOACOB, a number of ants probabilistically construct solutions using a given pheromone model; then, a local

search procedure is applied to the constructed solutions. In general, this multi-objective approach based on ACO outperformed another three biclustering algorithms in terms of the size of the obtained biclusters.

AISs are inspired by the principles of immunology and the observed immune process of vertebrates [111]. Additionally, AIS is highly robust, adaptive, inherently parallel, and self-organized. In line with these, Coelho et al. [14] proposed a multi-objective biclustering algorithm based on AIS to analyze texts, named **BIC-aiNet**. In the text mining problem, the input data matrix is composed of rows (texts) and columns (attributes of the corresponding texts), and the aim is to find the bipartitions of the whole dataset. The BIC-aiNet algorithm uses an integer bicluster representation, a mutation strategy to insert or remove rows and columns, and a suppression procedure to eliminate entire biclusters if a particular condition is satisfied. This approach was compared with the k-means algorithm, showing that BIC-aiNet discovered more meaningful text biclusters.

The simulated annealing-based multi-objective optimization (AMOS) algorithm [4] has been used as the underlying optimization strategy for finding bicluster in gene expression data [1, 99]. First, Sahoo et al. [99] presented an AMOSA-based biclustering algorithm (AMOSAB) that optimized the MSR metric and the row variance (rVAR) simultaneously. AMOSAB used a real-based encoding of biclusters and a decodification method based on the Euclidean distance to obtain the final biclusters. Then, the same authors, Acharya et al. [1], proposed modifications to the AMOSAB algorithm where the decodification method considered three different distance functions: Euclidean, Point Symmetry (PS), and Line Symmetry (LS). The results showed that the AMOSA algorithm using the PS and LS distance performed better than the Euclidean version.

5.5 Summary of Multi-objective Biclustering Approaches

We analyzed 23 biclustering approaches that use different nature-inspired multi-objective meta-heuristics, such as NSGA-II, SPEA2, MOPSO, and AMOSA. The main characteristics of these methods are summarized in Table 4, where multi-objective evolutionary algorithms are the most widely used with 63%, the NSGA-II algorithm being the most common of this group with 42%. Regarding the objective functions used to guide the search of the multi-objective algorithms, we noticed that the bicluster size (bSIZE), the bicluster coherence (MSR metric), and the row variance (rVAR) are commonly optimized simultaneously, as these criteria are independent and usually in conflict. Finally, regarding the types of bicluster representations, it was noted that both techniques are used almost equally, the binary representation (BBE) with 58%, while the integer representation (IBE) with 42%. However, in multi-objective algorithms, there is a tendency in recent years to use the IBE representation, which is more efficient than the binary representation for biclustering problems.

6 DATASETS, SOFTWARE TOOLS, AND VALIDATION METRICS

When aiming to develop a new biclustering approach or to use an existing one, it is important to be aware of the various resources available that the scientific community has proposed. These resources help not only to facilitate the development of a new approach but also to verify its validity and scope. Therefore, this section presents the various available datasets, software tools, and performance metrics to address biclustering problems.

6.1 Synthetic and Real-life Datasets

Benchmarking plays an essential role in studying biclustering methods and, more generally, in optimization-based algorithms [5]. The selection of suitable datasets helps to analyze the strengths and weaknesses of different techniques; thus, this knowledge can be used to develop more efficient optimization-based biclustering approaches. Table 5 summarizes some datasets used in different

Table 5. Summary of Biclustering Case Studies in Different Real-world Applications

Data	Application	Description	Software Tool	Ref.
Microarray (6,200 ORFs and 515 conditions)	Gene classification/ Yeast	Biclustering for gene classification	SAMBA	[112]
Microarray (21,031 genes and 351 conditions)	Gene classification/ Plant genomes	Biclustering on known genes and construct co-expression network	QUBIC	[118]
Microarray data with 12,533 probes from 72 patients	Disease subtype/ Leukemia	Biclustering by qualitative biclustering algorithm	QUBIC	[61]
Microarray data for 80 paired gastric cancer and reference tissues	Disease subtype/ Gastric cancer	Biclustering on gene expression data and pathways associated with cancer development	QUBIC	[16]
Microarray data with 7,756 genes and clinical data from 437 patients	Disease subtype/ Brest cancer	Biclustering to identify molecular-based tumor subgroup	cMonkey	[119]
6,100 gene expression profiles of human cancer cell	Gene-drug association	Biclustering drug-induced gene expression profiles	ISA	[48]
7,056 genome expression profiles of five human cells	Gene-drug association	Identify drug-gene modules by biclustering method	FABIA	[121]
Matrix of normalized miRNA-seq expression profiles	Biomarker and gene signatures detection	Biclustering to evaluate miRNA deregulation	ISA	[32]
Coefficients matrix of 1,109 unique SNPs associated with 23 studied traits	Biomarker and gene signatures detection	Gene annotation and identification of enriched canonical pathway and gene network inference	Bayesian Bic	[41]
Microarray (Mycobacterium tuberculosis under 2,325 measures)	Modularity analysis	Biclustering for tuberculosis data	cMonkey2	[96]
Microarray (4,117 orthologs in 46 tissue groups)	Modularity analysis	Biclustering to predict co-regulated modules	ISA	[122]

Table 6. Summary of Available Software Tools for Biclustering

Year	Toolbox Name	PL	Publication	Source Code	Metaheuristic
2010	BAT	Java/Web	BAT: A new biclustering analysis toolbox	BAT Link	✓
2013	MTBA	Matlab	MTBA: A MATLAB toolbox for Biclustering analysis	MTBA Link	×
2014	BiBench	Python	Application of biclustering algorithms to biological data	BiBench Link	×
2014	-	Matlab	Similarity measures for comparing biclusterings	Link	×
2016	BicNET	Java/Web	BicNET: Flexible module discovery in large-scale biological networks using biclustering	BicNET Link	×
2017	Biclustlib	Python	A systematic comparative evaluation of biclustering techniques	Biclustlib Link	×
2018	BMOEA	Python	Biclustering multi-objective evolutionary algorithms	BMOEA Link	✓
2019	CoClust	Python	CoClust: A Python package for co-clustering	CoClust Link	×
2020	RecBic	C++	RecBic: A fast and accurate algorithm recognizing trend-preserving biclusters	RecBic Link	×
2021	BiClust	R	BiClust: A toolbox for bicluster analysis in R	BiClust Link	×
2021	BIDEAL	Matlab	BIDEAL: A toolbox for bicluster analysis—generation, visualization, and validation	BIDEAL Link	×

The “Metaheuristic” column indicates if the package contains algorithms based on metaheuristics. “PL” indicates the type of programming language.

biological applications where the main objective is to find biclusters from microarray and gene expression data.

6.2 Availability of Software Tools

As can be seen, a large number of algorithms have been proposed to address various bicluster problems. The scientific community has created several software tools comprising these algorithms to facilitate their applicability. A number of such software tools implemented in various programming languages are listed in Table 6.

6.3 Bicluster Validity Metrics

Another important aspect of unsupervised learning is the numerical validation of biclustering algorithms. In biclustering (as in cluster analysis), no existing algorithm guarantees an optimal solution. Therefore, evaluation of the results produced by a biclustering algorithm is essential, and there are several bicluster validation techniques in the literature. In Section 3.2, we introduced internal bicluster criteria, which are used as objective functions in metaheuristics. Nevertheless, this section focuses on external validity metrics, which imply that the results of a biclustering algorithm are evaluated based on a pre-specified structure. In other words, external criteria rely on the *a priori* knowledge or ground truth about the input dataset.

Let us consider the following notation and definitions for the presentation of the different similarity measures for biclustering: Let $\mathbf{X} \in \mathbb{R}^{N \times M}$ be a data matrix with N patterns (rows) and M attributes (columns). Then, let us define the set of patterns as $R = \{1, 2, \dots, N\}$ and the set of attributes as $C = \{1, 2, \dots, M\}$; therefore, the matrix $\mathbf{X}_{R,C} = (R, C)$ denotes the full dataset \mathbf{X} . In bicluster analysis, a *bicluster* $B = (I, J)$ is a tuple of two nonempty sets such that $I \subseteq R$ and $J \subseteq C$. Thus, a *biclustering solution* is a collection $\mathbb{B} = \{B_1, B_2, \dots, B_k\}$ of k biclusters of the data represented by \mathbf{X} .

In the following definitions, we assume that $\mathbb{B} = \{B_i\}_{i=1}^k$ and $\mathbb{B}^* = \{B_i^*\}_{i=1}^q$ are the found solution by the algorithm and the reference biclustering solution, respectively. The most common similarity measures (dissimilarity measures were transformed into similarities) are detailed below.

– Prelić et al. [92] proposed the measures of relevance S_{PreRel} and recovery S_{PreRec} :

$$S_{\text{PreRel}}(\mathbb{B}, \mathbb{B}^*) = \sqrt{S_{\text{R}}(\mathbb{B}, \mathbb{B}^*) \times S_{\text{C}}(\mathbb{B}, \mathbb{B}^*)}, \quad (11)$$

$$S_{\text{PreRec}}(\mathbb{B}, \mathbb{B}^*) = S_{\text{PreRel}}(\mathbb{B}^*, \mathbb{B}), \quad (12)$$

where $S_{\text{R}}(\mathbb{B}, \mathbb{B}^*)$ and $S_{\text{C}}(\mathbb{B}, \mathbb{B}^*)$ are defined as follows:

$$S_{\text{R}}(\mathbb{B}, \mathbb{B}^*) = \frac{1}{k} \sum_{B_i \in \mathbb{B}} \max_{B_j^* \in \mathbb{B}^*} \left\{ \frac{|I_i \cap I_j^*|}{|I_i \cup I_j^*|} \right\} \quad \text{and} \quad S_{\text{C}}(\mathbb{B}, \mathbb{B}^*) = \frac{1}{k} \sum_{B_i \in \mathbb{B}} \max_{B_j^* \in \mathbb{B}^*} \left\{ \frac{|J_i \cap J_j^*|}{|J_i \cup J_j^*|} \right\}. \quad (13)$$

– Liu and Wang [66] proposed the measure $S_{\text{L\&W}}$:

$$S_{\text{L\&W}}(\mathbb{B}, \mathbb{B}^*) = \frac{1}{k} \sum_{B_i \in \mathbb{B}} \max_{B_j^* \in \mathbb{B}^*} \left\{ \frac{|I_i \cap I_j^*| + |J_i \cap J_j^*|}{|I_i \cup I_j^*| + |J_i \cup J_j^*|} \right\}. \quad (14)$$

– Dice [24, 100] proposed the measure S_{Dice} as follows:

$$S_{\text{Dice}}(\mathbb{B}, \mathbb{B}^*) = \frac{1}{k} \sum_{B_i \in \mathbb{B}} \max_{B_j^* \in \mathbb{B}^*} \left\{ 2 \times \frac{|I_i \times J_i \cap I_j^* \times J_j^*|}{|I_i \times J_i| + |I_j^* \times J_j^*|} \right\}. \quad (15)$$

– Ayadi et al. [3] proposed the measure:

$$S_{\text{Ayadi}}(\mathbb{B}, \mathbb{B}^*) = \frac{1}{k} \sum_{B_i \in \mathbb{B}} \max_{B_j^* \in \mathbb{B}^*} \left\{ \frac{|I_i \cap I_j^*| \times |J_i \cap J_j^*|}{|I_i \cup I_j^*| \times |J_i \cup J_j^*|} \right\}. \quad (16)$$

– Eren et al. [31] presented the measures:

$$S_{\text{ERel}}(\mathbb{B}, \mathbb{B}^*) = \frac{1}{k} \sum_{B_i \in \mathbb{B}} \max_{B_j^* \in \mathbb{B}^*} \left\{ \text{Jaccard}(B_i, B_j^*) \right\}, \quad (17)$$

$$S_{\text{ERec}}(\mathbb{B}, \mathbb{B}^*) = S_{\text{ERel}}(\mathbb{B}^*, \mathbb{B}), \quad (18)$$

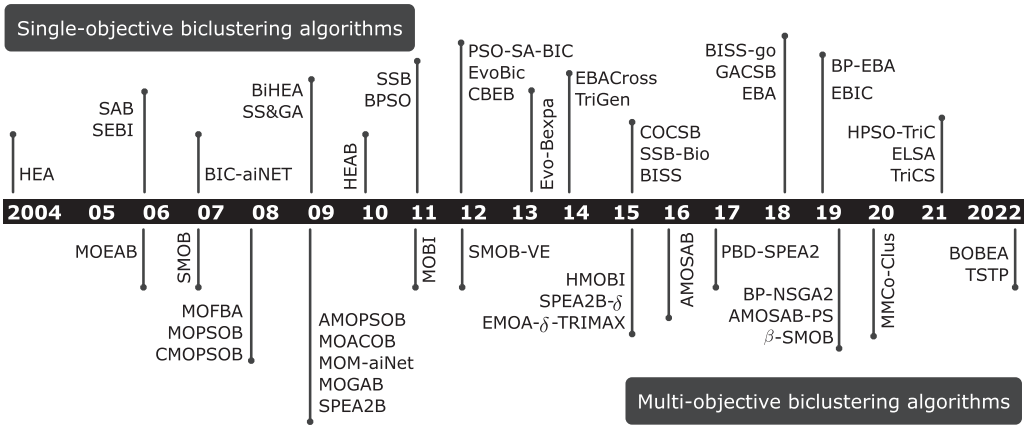


Fig. 7. Timeline of metaheuristic biclustering algorithms proposed in the past 22 years. Above the timeline, the 25 single-objective approaches are presented, whereas below, the 23 multi-objective algorithms are displayed.

where $Jaccard(B_i, B_j^*)$ denotes the Jaccard index applied to the biclusters B_i and B_j^* and is defined as

$$Jaccard(B_i, B_j^*) = \frac{|I_i \times J_i \cap I_j^* \times J_j^*|}{|I_i \times J_i \cup I_j^* \times J_j^*|}. \quad (19)$$

For a more detailed overview of validation metrics for comparing biclustering solutions, the reader is referred to the work of Horta and Campello [46], where 14 external validity metrics from the literature are described and analyzed, and some recommendations for their use are provided.

6.4 Summary of Available Bicluster Analysis Tools

Since the first biclustering method was created in 2000 by Cheng and Church [12], the research community has developed and made several resources available to address biclustering problems. In this regard, this section has summarized some of these essential resources to help the reader to have a better overview when approaching the bicluster analysis problem. First, the section presents some synthetic data sources and data from real applications. Then, a list of biclustering algorithms software packages is presented (several of these algorithms are reported in Sections 4 and 5). Finally, this section ends with the presentation of several performance metrics for comparing pairs of clustering solutions. The availability and knowledge of all these resources are essential for the development of new and improved biclustering methods.

7 PERSPECTIVES AND FUTURE RESEARCH DIRECTIONS

7.1 Influence of Diverse Metaheuristics on Bicluster Analysis

Metaheuristic-based biclustering algorithms have gained much relevance in recent decades, mainly because biclustering remains an important problem in practice and, computationally, it is a highly combinatorial problem. In this sense, Figure 7 illustrates the 48 algorithms surveyed in this manuscript that have been proposed within the past 22 years. Similarly, these approaches are listed in Table 7 according to their underlying metaheuristic algorithm.

From Table 7, metaheuristics based on evolutionary computation have been widely recognized as a major approach to addressing the biclustering problem. In particular, *genetic algorithms*

Table 7. Metaheuristic Algorithms and Corresponding Biclustering Algorithms

	Metaheuristic	Times	Metaheuristic Biclustering Algorithms
Single-objective	GA	13	HEA [8], SEBI [26], BiHEA [34], SS&GA [79], HEAB [78] CBEB [94], EvoBic [3], Evo-Bexpa [89], EBACross [68], TriGen [42], EBA [69], BP-EBA [47], ELSA [70]
	SS	6	SS&GA [79], HEAB [78], SSB [80], SSB-Bio [82], BISS [81], BISS-go [83]
	PSO	3	BPSO [95], PSO-SA-BIC [114], HPSO-TriC [77]
	CS	3	COCSB [67], GACSB [125], TriCS [110]
	SA	1	SAB [9]
	AIS	1	BIC-aiNET [19]
	HBA	5	SS&GA [79], HEAB [78], PSO-SA-BIC [114], GACSB [125], HPSO-TriC [77]
Multi-objective	NSGA-II	11	MOEAB [75], SMOB [27], SMOB-VE [28], EMOA- δ -TRIMAX [7], BP-NSGA2 [57], MMCo-Clus [17], TSTP [108], MOFBA [74], MOGAB [73], MOBI [101], BOBEA [71]
	MOPSO	3	MOPSOB [65], CMOPSOB [63], AMOPSOB [60]
	SPEA2	3	SPEA2B [35], SPEA2B- δ [38], PBD-SPEA2 [39]
	AMOSAB	2	AMOSAB [99], AMOSAB-PS [1]
	MOACO	1	MOACOB [64]
	MOAIS	1	MOM-aiNet [14]
	IBEA	1	HMOBI [102]

“HBA” indicates hybrid biclustering algorithms. “Times” indicates the number of approaches using that particular metaheuristic.

(GA) and their extensions to multi-objective optimization, such as NSGA-II and SPEA2, represent about 70% of studies. Other computational intelligence paradigms based on metaheuristics are less present; for example, *particle swarm optimization* (PSO and MOPSO) represent 15%, while approaches based on *scatter search*, *simulated annealing*, and *artificial immune systems* have been used in smaller proportion. Unsurprisingly, GAs are the most widely used in bicluster analysis, since these algorithms are among the most studied and applied metaheuristics [59, 106]. As a reminder, GAs solve optimization problems based on natural selection (biological evolution), which repeatedly modifies a population of individual solutions over successive generations until it “evolves” toward an optimal solution. Thus, GAs can solve different optimization problems, including those where the objective function is discontinuous, nondifferentiable, stochastic, or highly nonlinear [45].

However, from Table 7, it is possible to identify other well-known metaheuristics that still need to be fully explored to address the problem of bicluster analysis. Particularly, **differential evolution (DE)**, proposed by Storn and Price [107], is an evolutionary algorithm not used in biclustering. Similarly, DE’s extension to multi-objective problems could be adapted to biclustering; a comprehensive review of DE for multiobjective optimization is given in Reference [15]. *Artificial immune systems* [127] is another example of metaheuristics that can be further explored for bicluster analysis.

Finally, in this study, we have classified the metaheuristic biclustering approaches into two main categories, depending on their underlying search strategy, namely, single-objective and multi-objective biclustering approaches. Indeed, single-objective metaheuristics represent 52% of the surveyed algorithms, whereas multi-objective algorithms represent 48%. Contrary to single-objective approaches, multi-objective biclustering algorithms optimize multiple biclustering metrics simultaneously, having the advantage of discovering several types of biclusters (see Section 2.1).

Table 8. Bicluster Encoding Types and Their Preferences in Different Single- and Multi-objective Biclustering Algorithms

Encoding	Times	Single-objective Biclustering	Multi-objective Biclustering
BBE [8]	25 (75%)	HEA [8], SEBI [26], BIC-aiNET [19], BiHEA [34], SS&GA [79], HEAB [78], SSB [80], BPSO [95], CBEB [94], PSO-SA-BIC [114], Evo-Bexpa [89], EBACross [68], COCSB [67], SSB-Bio [82], BISS [81], BISS-go [83], GACSB [125], EBA [69], BP-EBA [47], HPSO-TriC [77], ELSA [70], TriCS [110]	MOEAB [75], SMOB [27], MOPSOB [65], CMOPSOB [63], AMOPSOB [60], MOACOB [64], SPEA2B [35], SMOB-VE [28], SPEA2B- δ [38], EMOA- δ -TRIMAX [7], BP-NSGA2 [57], MMCo-Clus [17], TSTP [108]
			MOFBA [74], MOGAB [73], MOM-aiNet [14], MOBI [101], HMOBI [102], AMOSAB [99], AMOSAB-PS [1], PBD-SPEA2 [39], BOBEA [71]
IBE [19]	12 (25%)	BIC-aiNET [19], EvoBic [3], TriGen [42]	

“BBE” refers to binary bicluster encoding and “IBE” indicates binary encoding. “Times” indicates the number of approaches using the encoding type, with the percentage in parentheses.

However, these approaches generate a set of solutions (Pareto set approximation) requiring an additional mechanism to filter and select the best biclustering solution. Therefore, the optimization technique’s selection depends on the complexity of the biclustering problem in terms of the type of biclusters and the bicluster structures to be discovered (overlapping among bicluster and matrix covering matrix).

7.2 Impact of Different Bicluster Encodings in Metaheuristic Algorithms

Based on the revised works summarized in Table 8 and illustrated in Figure 8(a), it is notable that a large number of metaheuristic biclustering approaches (75%) consider a binary bicluster encoding (BBE). However, 25% of the surveyed approaches utilize an integer bicluster encoding. The definition of these two bicluster representations, including advantages and disadvantages, are given in Section 3.1.

Regarding the use of both types of representations in Figure 8(a), it is observed that the integer representation is mostly used in multi-objective approaches, while the binary encoding is used interchangeably between single-objective and multi-objective approaches. Therefore, with the exception of genetic algorithms, all other single-objective metaheuristics use a binary representation, including PSO, AIS, and simulated annealing. In contrast, there is clear evidence for using an integer representation to represent bicluster solutions in multi-objective algorithms such as NSGA-II, SPEA2, AMOSA, and MOACO.

7.3 Understanding the Role of Bicluster Quality Metrics in Metaheuristic Algorithms

Several bicluster quality metrics have been proposed in the bicluster analysis literature, and most of them have been used as optimization criteria in metaheuristic biclustering algorithms. These metrics were described in detail in Section 3.2, and their importance for identifying different types of biclusters. Table 9 summarizes bicluster metrics used in metaheuristic blustering algorithms, whereas Figure 8(b) illustrates the relationship between these bicluster quality metrics with different types of metaheuristic-based biclustering algorithms.

First, the MSR metric was the most-used metric among all metrics mentioned in this survey, with 34%. Moreover, MSR was used indifferently between the types of metaheuristics and between single- and multi-objective approaches. Then, bSIZE is the most-used metric after MSR; bSIZE has been used among different metaheuristics and mostly in multi-objective algorithms. Next, ACF has only been used with genetic algorithms and scatter search, so it has only been used in single-objective approaches. Next, the rVAR metric has been widely used in multi-objective approaches such as NSGA-II, MOPSO, and SPEA2. Finally, the ACV, SF, VE, and SMRS metrics have been used

Table 9. Bicluster Metrics and Their Appearances in Different Single- and Multi-objective Biclustering Algorithms Reviewed in Sections 4 and 5, Respectively

Metric	Times	Single-objective Biclustering	Multi-objective Biclustering
MSR [12]	33 (34%)	HEA [8], SAB [9], SEBI [26], BIC-aiNET [19], SS&GA [79], CBEB [94], EvoBic [3], EBACross [68], TriGen [42], COCSB [67], GACSB [125], EBA [69], BP-EBA [47], TriCS [110]	MOEAB [75], SMOB [27], MOFBA [74], MOPSOB [65], CMOPSOB [63], AMOPSOB [60], MOACOB [64], MOM-aiNet [14], MOGAB [73], SPEA2B [35], MOBI [101], HMOBI [102], SPEA2B- δ [38], EMOA- δ -TRIMAX [7], AMOSAB [99], PBD-SPEA2 [39], BP-NSGA2 [57], AMOSAB-PS [1], MMCo-Clus [17]
			MOEAB [75], SMOB [27], MOFBA [74], MOPSOB [65], CMOPSOB [63], AMOPSOB [60], MOACOB [64], MOM-aiNet [14], SPEA2B [35], MOBI [101], SMOB-VE [28], SPEA2B- δ [38], EMOA- δ -TRIMAX [7], AMOSAB [99], PBD-SPEA2 [39], BP-NSGA2 [57], AMOSAB-PS [1], BOBEA [71], TSTP [108]
bSIZE [12]	26 (27%)	SEBI [26], EvoBic [3], Evo-Bexpa [89], EBACross [68], SSB-Bio [82], EBA [69], BP-EBA [47]	SMOB [27], MOFBA [74], MOPSOB [65], CMOPSOB [63], AMOPSOB [60], MOGAB [73], SPEA2B [35], MOBI [101], SMOB-VE [28], EMOA- δ -TRIMAX [7], AMOSAB [99], PBD-SPEA2 [39], BP-NSGA2 [57], AMOSAB-PS [1], BOBEA [71], TSTP [108]
			SMOB [27], MOFBA [74], MOPSOB [65], CMOPSOB [63], AMOPSOB [60], MOGAB [73], SPEA2B [35], MOBI [101], SMOB-VE [28], HMOBI [102], PBD-SPEA2 [39], MMCo-Clus [17], BOBEA [71]
rVAR [12, 26]	15 (16%)	SEBI [26], Evo-Bexpa [89]	
ACF [78]	10 (12%)	HEAB [78], SSB [80], EvoBic [3], EBACross [68], SSB-Bio [82], BISS [81], BISS-go [83], EBA [69], HPSO-TriC [77], ELSA [70]	
ACV [113]	4 (5%)	BPSO [95], PSO-SA-BIC [114], GACSB [125]	TSTP [108]
VE [28]	3 (3%)	Evo-Bexpa [89], GACSB [125]	SMOB-VE [28]
CVF [68]	2 (2%)	EBACross [68], EBA [69]	
SMSR [76]	1 (1%)	BP-EBA [47]	

“Times” indicates the number of approaches using the type of bicluster metric, with the percentage in parentheses.

to a lower extent, with percentages below 5%; therefore, it is difficult to conclude on the preferred use of these bicluster metrics.

In the specialized literature of bicluster analysis, many other bicluster metrics have not been used in the context of metaheuristic biclustering algorithms, such as *relevance index* (RI) [126], *sub-matrix correlation score* (SCS) [123], *Spearman’s biclustering measure* (SBM) [33], and *maximal standard area* (MSA) [37]. Therefore, it would be convenient to use and study the performance of these metrics as objective functions in metaheuristic biclustering algorithms.

7.4 Exploring the Applications of Metaheuristic Biclustering Algorithms

Most of the described metaheuristic biclustering algorithms presented in Sections 4 and 5 are applied to synthetic test problems. The advantage of investigating performance of these biclustering algorithms on test problems is that the problem complexity, such as the type and number of biclusters, is known *a priori*. Thus, applying these biclustering algorithms to test problems becomes useful and informative. However, testing the performance of metaheuristic biclustering algorithms on application problems is also necessary to demonstrate the use of the algorithm in practice.

In Tables 3 and 4, presented in previous sections, we have identified three main applications of metaheuristic biclustering algorithms: *gene expression data*, *text mining*, and *energy consumption analysis*. Furthermore, Figure 9 illustrates these three applications of biclustering algorithms and their interactions with their main components, including the type of optimization, type of metaheuristics, and type of coding of the biclustering solutions. First, from Figure 9, it

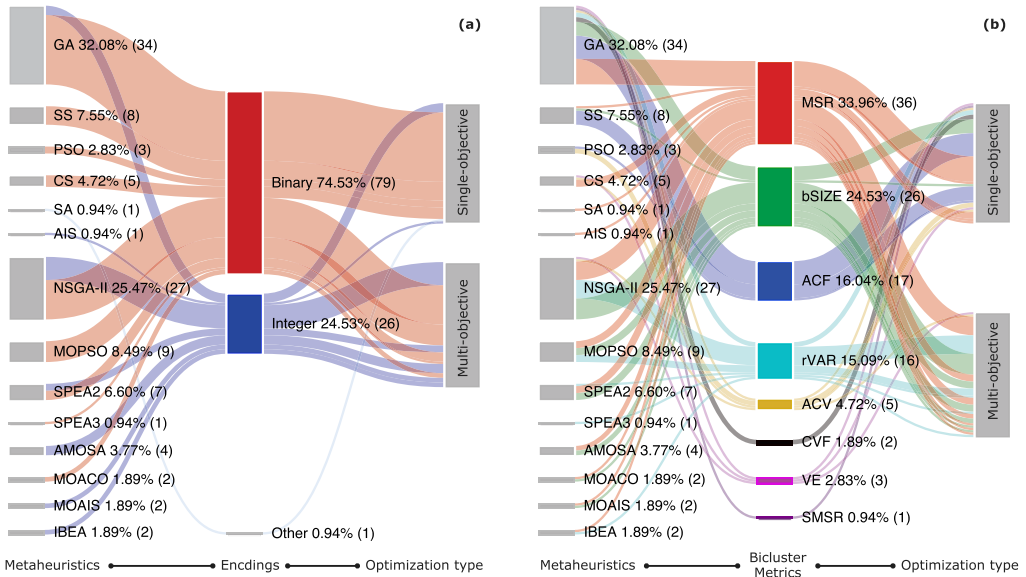


Fig. 8. Visualization of two main components in metaheuristic biclustering algorithms: (a) encoding types and (b) bicluster metrics. The plots illustrate the relationship between these components with different metaheuristics and optimization methods.

can be clearly observed that most of the case studies have focused on problems related to gene expression data analysis (94%), followed by text mining (4%), and finally, data analysis of energy consumption (2%). This trend of metaheuristic algorithms is related to the successful application of the biclustering concept applied in bioinformatics, particularly to cluster gene expression datasets in two dimensions: genes and conditions [12]. Regarding bicluster analysis for text mining data, biclustering algorithms have been used on web data to identify relationships between users and web pages; this information is useful in e-commerce applications such as web advertising and marketing [95]. Finally, metaheuristic biclustering algorithms have been applied to identify subgroups of buildings with similar energy consumption behavior over time; these insights can help policymakers detect irregular situations and improve the efficiency of buildings [29]. In this direction, the applicability of metaheuristic-based biclustering algorithms in bioinformatics is remarkable. However, their potential for other biclustering domains is still very limited; for example, text mining [10, 19, 85], dimensionality reduction [87], recommendation systems [18, 109, 117], energy [29] and water [105] consumption, time-series analysis [11, 42], disease identification [115, 120], and marked segmentation [62].

Since most metaheuristic biclustering focus on the analysis of gene expression data, Figure 9 allows us to easily identify the interaction of specific components of these algorithms to solve the problem in gene expression data analysis. First, about 60% of algorithms are single-objective, while about 40% of approaches address the gene expression problem as a multi-objective optimization. The optimization type depends on the number of bicluster metrics that simultaneously guide the solution search during the optimization process. Regarding the type of metaheuristics, genetic algorithm-based approaches (GA and NSGA-II) are the most used, representing 45% studies. Surprisingly, the Scatter Search (SS) metaheuristic is widely used, with 11%; SS constructs new test solutions by combining reference solutions and using strategic designs exploiting contextual knowledge [58]. Regarding the encoding type, 76% use binary representation, while 22%

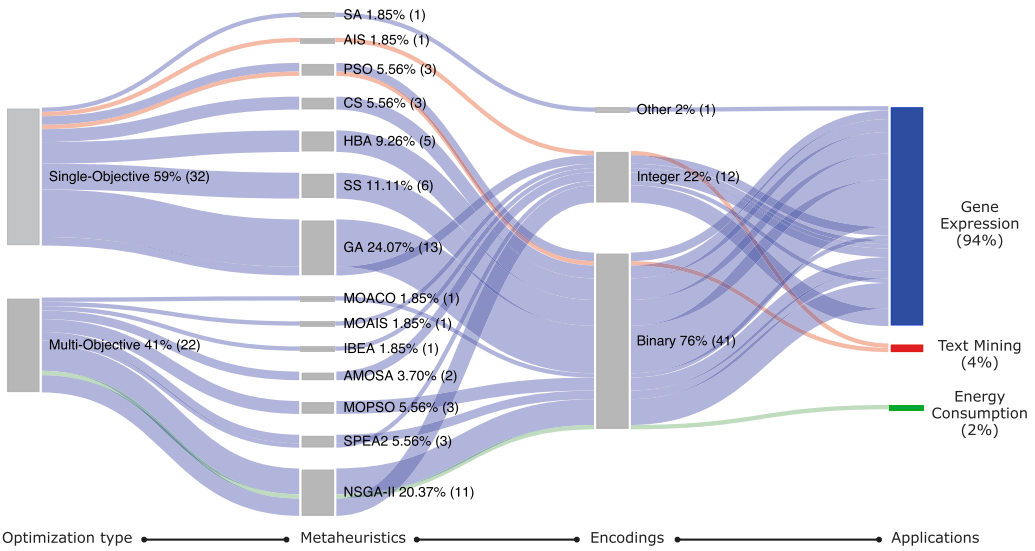


Fig. 9. Application problems of metaheuristic biclustering algorithms and their interactions with the algorithm components: optimization type, metaheuristic type, and bicluster encoding. “Metaheuristics” indicates the type of metaheuristic algorithm: *Simulated Annealing* (SA), *Genetic Algorithm* (GA), *Particle Swarm Optimization* (PSO), *Scatter Search* (SS), *Cuckoo Search* (CS), *Artificial Immune System* (AIS). “Binary Encoding” and “Integer Encoding” refer to the type of bicluster encoding as described in Section 3.1.

use integer-based representation. Most single-objective metaheuristics use a binary representation, including PSO, AIS, and SA, except for genetic algorithms, whose approaches use both representations. However, most multi-objective metaheuristics use an integer representation of bicluster solutions. These trends in the metaheuristic components can be seen in Figure 9.

7.5 Future Research Directions

Bicluster analysis is an open scientific field with several research directions, opportunities, and challenges that commonly involve the following issues:

- **Emerging metaheuristics.** Novel nature-inspired metaheuristics are continuously proposed in the literature as potential approaches to solving the biclustering problem. Mainly, many-objective optimization evolutionary algorithms such as MOEA/D [93], NSGA-III [20] can be used to cope with multiple bicluster quality metrics (more than three objective functions). It is essential to mention that selecting the best biclustering solution is an added challenge when using these multi-objective biclustering approaches.
- **Bicluster encoding.** To solve complex biclustering problems and discover different bicluster types and structures, studying and analyzing the current objective function and bicluster representations is necessary to help select the appropriate optimization scheme and components according to the biclustering scenario. For instance, an integer based-representation is preferable over a binary representation [1, 101]. Most of these approaches using an integer representation encode a single bicluster; however, it is possible to codify multiple biclusters in a single solution, as demonstrated recently by Golchin [39].
- **Biclustering applications.** Most of the proposed metaheuristic-based biclustering algorithms have been designed to work on biological data (mainly gene expression data). However, the applicability of biclustering algorithms in other domains and with other data types

- (e.g., heterogeneous) is very limited, as it poses additional challenges. In Reference [115], the authors proposed a greedy procedure to extract biclusters from heterogeneous, temporal, and large-scale data. This procedure has been applied successfully to Electronic Health Records (EHR), thanks to the data sparsity in this scenario, optimizing the enumeration. It will be interesting to study the potential of metaheuristics to discover heterogeneous-like bicluster in EHR applications.
- **Bicluster benchmark.** Although many synthetic and real-life datasets have been used systematically in the literature, there is no recognized benchmark the research community could use to evaluate and compare biclustering approaches. Such a benchmark should include different types of bicluster, diverse bicluster structures, noise, overlapping, and so on. Furthermore, when comparing the performance of biclustering approaches, it is crucial to consider their statistical and biological significance (i.e., the available biological information [70]).
 - **Comparative studies of metaheuristic biclustering algorithms and components.** There is a need for comparative studies of different metaheuristic biclustering algorithms and their components. There are several comparative studies of traditional biclustering algorithms to determine their performance against different types of biclusters [84, 88, 91]. However, in the context of biclustering algorithms based on metaheuristics, these studies are needed to determine the specific scenarios where these algorithms provide a better advantage over traditional algorithms. In addition, it is important to study the discriminative ability of bicluster metrics when used as objective functions in metaheuristic algorithms to find biclusters with different characteristics.
 - **Triclustering.** Recently, the biclustering problem is referred to as triclustering when the time dimension is considered in addition to rows and columns information. Finding tricluster when considering temporal data brings up new research challenges, as it will require the adaptation of current algorithms, bicluster metrics, evaluation measures, and so on. Indeed, the triclustering problem has been addressed recently using single-objective metaheuristics [42, 77]; however, there are opportunities to address this problem as multi-objective optimization.
 - **Multi-view biclustering.** There are many application domains where multiple pieces of information are available for each individual subject. For instance, multiple data matrices might be available for the same set of genes and conditions in biological datasets. In this regard, multi-view data clustering algorithms can integrate these information pieces to find consistent clusters across different data views [50, 51]. This same multi-view clustering concept can be extended to biclustering, where the aim is to discover biclusters across multiple data matrices (i.e., data views).
 - **Large-scale biclustering.** The decentralized nature of high-dimensional biclustering problems makes it very interesting to use high-performance computing to distribute the underlying computations across increasingly available computing facilities. Although challenging due to the heterogeneous and complex nature of modern large-scale computing platforms, the integration and enablement of parallelism must be closely coupled to the biclustering problem components, not separated. A scalable method for biclustering big data called EBIC was recently proposed for multi-GPU environments [86]. This approach is one of the first steps in this direction, but further efforts are needed to bridge the gap between current biclustering algorithms and new developments of scalable methods, including metaheuristic-based approaches.
 - **Visualization of biclusters.** The validation and interpretability of obtained biclusters are essential steps in real-world applications with high-dimensional feature spaces. In

this regard, bicluster-data visualization will help better understand and communicate the obtained solutions [116, 120]. However, this aspect has yet to be studied in bicluster analysis. Biclusters are usually represented in the form of line plots [1, 17] (the set of horizontal lines represents highly correlated patterns across a certain number of features) and heatmap plots [39, 83] (rows and columns of the input matrix are reordered to display the found biclusters properly). Therefore, there is a need to develop more suitable bicluster visualization approaches that facilitate the interpretation of the results, especially in applications with high dimensional data and heterogeneous features [53].

8 CONCLUSION

Biclustering has emerged as an important approach and is essential in various applications ranging from bioinformatics to text mining. However, this is an NP-hard optimization problem from the computational point of view. In this regard, many nature-inspired metaheuristics have been applied to address the bicluster analysis problem.

This survey presented a detailed study of metaheuristics approaches to address different biclustering challenges. The review focused on the underlying optimization methods and their main search components: biclustering encoding, variation operators, and bicluster objective functions. This review focused on single versus multi-objective approaches. Additionally, we presented a discussion and emerging research directions.

Applications of bicluster analysis continue to emerge, and we expect this to continue moving forward in the coming years. While biclustering has been used heavily in bioinformatic tasks, applications continue to appear in digital health, data mining, and dimensionality reduction. Ideally, this survey will help drive further theoretical and algorithm development based on metaheuristics for bicluster analysis.

ACKNOWLEDGMENTS

The authors are grateful to the University of Lille, CHU Lille, and INSERM, founded by the MEL through the I-Site cluster humAIn@Lille.

REFERENCES

- [1] Sudipta Acharya, Sriparna Saha, and Pracheta Sahoo. 2019. Bi-clustering of microarray data using a symmetry-based multi-objective optimization framework. *Soft Comput.* 23, 14 (2019), 5693–5714. DOI : <https://doi.org/10.1007/s00500-018-3227-5>
- [2] Jesús S. Aguilar-Ruiz. 2005. Shifting and scaling patterns from gene expression data. *Bioinformatics* 21, 20 (2005), 3840–3845.
- [3] Wassim Ayadi, Ons Maatouk, and Hend Bouziri. 2012. Evolutionary biclustering algorithm of gene expression data. In *International Workshop on Database and Expert Systems Applications*. IEEE, 206–210. DOI : <https://doi.org/10.1109/dexa.2012.46>
- [4] Sanghamitra Bandyopadhyay, Sriparna Saha, Ujjwal Maulik, and Kalyanmoy Deb. 2008. A simulated annealing-based multiobjective optimization algorithm: AMOSA. *IEEE Trans. Evolut. Comput.* 12, 3 (2008), 269–283. DOI : <https://doi.org/10.1109/tevc.2007.900837>
- [5] Vahid Beiranvand, Warren Hare, and Yves Lucet. 2017. Best practices for comparing optimization algorithms. *Optim. Eng.* 18, 4 (2017), 815–848. DOI : <https://doi.org/10.1007/s11081-017-9366-1>
- [6] Amir Ben-Dor, Benny Chor, Richard Karp, and Zohar Yakhini. 2003. Discovering local structure in gene expression data: The order-preserving submatrix problem. *J. Comput. Biol.* 10, 3-4 (2003), 373–384. DOI : <https://doi.org/10.1089/10665270360688075>
- [7] Anirban Bhar, Martin Haubrock, Anirban Mukhopadhyay, and Edgar Wingender. 2015. Multiobjective triclustering of time-series transcriptome data reveals key genes of biological processes. *BMC Bioinform.* 16,1 (2015), 200. DOI : <https://doi.org/10.1186/s12859-015-0635-8>
- [8] S. Bleuler, A. Prelic, and E. Zitzler. 2004. An EA framework for biclustering of gene expression data. In *Congress on Evolutionary Computation (CEC'2004)*. IEEE, 166–173. DOI : <https://doi.org/10.1109/cec.2004.1330853>

- [9] K. Bryan, P. Cunningham, and N. Bolshakova. 2006. Application of simulated annealing to the biclustering of gene expression data. *IEEE Trans. Inf. Technol. Biomed.* 10, 3 (2006), 519–525. DOI: <https://doi.org/10.1109/titb.2006.872073>
- [10] Stanislav Busygin, Oleg Prokopyev, and Panos M. Pardalos. 2008. Biclustering in data mining. *Comput. Oper. Res.* 35, 9 (2008), 2964–2987. DOI: <https://doi.org/10.1016/j.cor.2007.01.005>
- [11] Eduardo Castanho, Helena Aidos, and Sara Madeira. 2022. Biclustering fMRI time series: A comparative study. *BMC Bioinform.* 23 (2022). DOI: <https://doi.org/10.1186/s12859-022-04733-8>
- [12] Yizong Cheng and George M. Church. 2000. Biclustering of expression data. In *International Conference on Intelligent Systems for Molecular Biology*. ISCB, 93–103.
- [13] Chun Tang, Li Zhang, Aidong Zhang, and M. Ramanathan. 2001. Interrelated two-way clustering: An unsupervised approach for gene expression data analysis. In *IEEE International Symposium on Bioinformatics and Bioengineering*. IEEE, 41–48. DOI: <https://doi.org/10.1109/bibe.2001.974410>
- [14] Guilherme Palermo Coelho, Fabrício Olivetti de França, and Fernando J. Von Zuben. 2009. Multi-objective biclustering: When non-dominated solutions are not enough. *J. Math. Model. Algor.* 8, 2 (2009), 175–202. DOI: <https://doi.org/10.1007/s10852-009-9102-8>
- [15] Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler. 2005. DEMO: Differential evolution for multiobjective optimization. In *Evolutionary Multi-criterion Optimization*. Springer Berlin, 520–533. DOI: https://doi.org/10.1007/978-3-540-31880-4_36
- [16] Juan Cui, Yunbo Chen, Wen-Chi Chou, Liankun Sun, Li Chen, Jian Suo, Zhaohui Ni, Ming Zhang, Xiaoxia Kong, Lisabeth L. Hoffman, Jinsong Kang, Yingying Su, Victor Olman, Darryl Johnson, Daniel W. Tench, I. Jonathan Amster, Ron Orlando, David Puett, Fan Li, and Ying Xu. 2011. An integrated transcriptomic and computational analysis for biomarker identification in gastric cancer. *Nucleic Acids Res.* 39, 4 (2011), 1197–1207. DOI: <https://doi.org/10.1093/nar/gkq960>
- [17] Laizhong Cui, Sudipta Acharya, Sumit Mishra, Yi Pan, and Joshua Zhexue Huang. 2020. MMCO-Clus—An evolutionary co-clustering algorithm for gene selection. *IEEE Trans. Knowl. Data Eng.* 34 (2020), 1–1. DOI: <https://doi.org/10.1109/tkde.2020.3035695>
- [18] Pablo A. D. de Castro, Fabrício O. de Franca, Hamilton M. Ferreira, and Fernando J. Von Zuben. 2007. Applying biclustering to perform collaborative filtering. In *7th International Conference on Intelligent Systems Design and Applications (ISDA'07)*. 421–426. DOI: <https://doi.org/10.1109/ISDA.2007.91>
- [19] Pablo A. D. de Castro, Fabrício O. de França, Hamilton M. Ferreira, and Fernando J. Von Zuben. 2007. Applying biclustering to text mining: An immune-inspired approach. In *International Conference on Artificial Immune Systems*. Springer, 83–94. DOI: https://doi.org/10.1007/978-3-540-73922-7_8
- [20] Kalyanmoy Deb and Himanshu Jain. 2014. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. *IEEE Trans. Evolut. Comput.* 18, 4 (2014), 577–601. DOI: <https://doi.org/10.1109/tevc.2013.2281535>
- [21] Kalyanmoy Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolut. Comput.* 6, 2 (2002), 182–197. DOI: <https://doi.org/10.1109/4235.996017>
- [22] Clarisse Dhaenens and Laetitia Jourdan. 2016. *Metaheuristics for Big Data*. John Wiley & Sons.
- [23] Clarisse Dhaenens and Laetitia Jourdan. 2019. Metaheuristics for data mining. *4OR* 17, 2 (2019), 115–139.
- [24] Lee R. Dice. 1945. Measures of the amount of ecologic association between species. *Ecology* 26, 3 (1945), 297–302. DOI: <https://doi.org/10.2307/1932409>
- [25] Chris Ding, Ya Zhang, Tao Li, and Stephen R. Holbrook. 2006. Biclustering protein complex interactions with a biclique finding algorithm. In *6th International Conference on Data Mining (ICDM'06)*. IEEE, 178–187.
- [26] Federico Divina and Jesús S. Aguilar-Ruiz. 2006. Biclustering of expression data with evolutionary computation. *IEEE Trans. Knowl. Data Eng.* 18, 5 (2006), 590–602. DOI: <https://doi.org/10.1109/tkde.2006.74>
- [27] Federico Divina and Jesús S. Aguilar-Ruiz. 2007. A multi-objective approach to discover biclusters in microarray data. In *Genetic and Evolutionary Computation Conference (GECCO'07)*. ACM Press, London, England, 385. DOI: <https://doi.org/10.1145/1276958.1277038>
- [28] Federico Divina, Beatriz Pontes, Raúl Giráldez, and Jesús S. Aguilar-Ruiz. 2012. An effective measure for assessing the quality of biclusters. *Comput. Biol. Med.* 42, 2 (2012), 245–256. DOI: <https://doi.org/10.1016/j.compbiomed.2011.11.015>
- [29] Federico Divina, Francisco Luis Gutiérrez Vela, and Miguel García Torres. 2019. Biclustering of smart building electric energy consumption data. *Appli. Sci.* 9, 2 (2019). DOI: <https://doi.org/10.3390/app9020222>
- [30] Marco Dorigo and Christian Blum. 2005. Ant colony optimization theory: A survey. *Theor. Comput. Sci.* 344, 2-3 (2005), 243–278. DOI: <https://doi.org/10.1016/j.tcs.2005.05.020>
- [31] K. Eren, M. Deveci, O. Kucuktunc, and U. V. Catalyurek. 2013. A comparative analysis of biclustering algorithms for gene expression data. *Brief. Bioinform.* 14, 3 (2013), 279–292. DOI: <https://doi.org/10.1093/bib/bbs032>
- [32] Antonino Fiannaca, Massimo La Rosa, Laura La Paglia, Riccardo Rizzo, and Alfonso Urso. 2015. Analysis of miRNA expression profiles in breast cancer using biclustering. *BMC Bioinform.* 16, S4 (2015), S7. DOI: <https://doi.org/10.1186/1471-2105-16-S4-S7>

- [33] Jose L. Flores, Iñaki Inza, Pedro Larrañaga, and Borja Calvo. 2013. A new measure for gene expression biclustering based on non-parametric correlation. *Comput. Meth. Progr. Biomed.* 112, 3 (2013), 367–397. DOI: <https://doi.org/10.1016/j.cmpb.2013.07.025>
- [34] Cristian Andrés Gallo, Jessica Andrea Carballido, and Ignacio Ponzoni. 2009. BiHEA: A hybrid evolutionary approach for microarray biclustering. In *Brazilian Symposium on Bioinformatics*. Springer, 36–47. DOI: https://doi.org/10.1007/978-3-642-03223-3_4
- [35] Cristian Andrés Gallo, Jessica Andrea Carballido, and Ignacio Ponzoni. 2009. Microarray biclustering: A novel memetic approach based on the PISA platform. In *European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*. Springer, 44–55. DOI: https://doi.org/10.1007/978-3-642-01184-9_5
- [36] G. Getz, E. Levine, and E. Domany. 2000. Coupled two-way clustering analysis of gene microarray data. *Proc. Nat. Acad. Sci.* 97, 22 (2000), 12079–12084. DOI: <https://doi.org/10.1073/pnas.210134797>
- [37] Raul Giraldez, Federico Divina, Beatriz Pontes, and Jesús S. Aguilar-Ruiz. 2007. Evolutionary search of biclusters by minimal intrafluctuation. In *IEEE International Fuzzy Systems Conference*. 1–6. DOI: <https://doi.org/10.1109/FUZZY.2007.4295631>
- [38] Maryam Golchin, Seyed Hashem Davarpanah, and Alan Wee-Chung Liew. 2015. Biclustering analysis of gene expression data using multi-objective evolutionary algorithms. In *International Conference on Machine Learning and Cybernetics*. IEEE, 505–510. DOI: <https://doi.org/10.1109/icmlc.2015.7340608>
- [39] Maryam Golchin and Alan Wee Chung Liew. 2017. Parallel biclustering detection using strength Pareto front evolutionary algorithm. *Inf. Sci.* 415–416 (2017), 283–297. DOI: <https://doi.org/10.1016/j.ins.2017.06.031>
- [40] Jiajun Gu and Jun S. Liu. 2008. Bayesian biclustering of gene expression data. *BMC Genom.* 9, Suppl 1 (2008), S4. DOI: <https://doi.org/10.1186/1471-2164-9-s1-s4>
- [41] Mayetri Gupta, Ching-Lung Cheung, Yi-Hsiang Hsu, Serkalem Demissie, L. Adrienne Cupples, Douglas P. Kiel, and David Karasik. 2011. Identification of homogeneous genetic architecture of multiple genetically correlated traits by block clustering of genome-wide associations. *J. Bone Miner. Res.* 26, 6 (2011), 1261–1271. DOI: <https://doi.org/10.1002/jbmr.333>
- [42] D. Gutiérrez-Avilés, C. Rubio-Escudero, F. Martínez-Álvarez, and J. C. Riquelme. 2014. TriGen: A genetic algorithm to mine triclusters in temporal gene expression data. *Neurocomputing* 132 (2014), 42–53. DOI: <https://doi.org/10.1016/j.neucom.2013.03.061>
- [43] J. A. Hartigan. 1972. Direct clustering of a data matrix. *J. Amer. Statist. Assoc.* 67, 337 (1972), 123. DOI: <https://doi.org/10.2307/2284710>
- [44] Sepp Hochreiter, Ulrich Bodenhofer, Martin Heusel, Andreas Mayr, Andreas Mitterecker, Adetayo Kasim, Tatsiana Khamiakova, Suzy Van Sanden, Dan Lin, Willem Talloen, Luc Bijnens, Hinrich W. H. Göhlmann, Ziv Shkedy, and Djork-Arné Clevert. 2010. FABIA: Factor analysis for bicluster acquisition. *Bioinformatics* 26, 12 (2010), 1520–1527. DOI: <https://doi.org/10.1093/bioinformatics/btq227>
- [45] John H. Holland. 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Michigan.
- [46] Danilo Horta and Ricardo J. G. B. Campello. 2014. Similarity measures for comparing biclusterings. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 11, 5 (2014), 942–954. DOI: <https://doi.org/10.1109/TCBB.2014.2325016>
- [47] Qinghua Huang, Xianhai Huang, Zhoufan Kong, Xuelong Li, and Dacheng Tao. 2019. Bi-phase evolutionary searching for biclusters in gene expression data. *IEEE Trans. Evolut. Comput.* 23, 5 (2019), 803–814. DOI: <https://doi.org/10.1109/tevc.2018.2884521>
- [48] Murat Iskar, Georg Zeller, Peter Blattmann, Monica Campillos, Michael Kuhn, Katarzyna H. Kaminska, Heiko Runz, Anne-Claude Gavin, Rainer Pepperkok, Vera van Noort, and Peer Bork. 2013. Characterization of drug-induced transcriptional modules: Towards drug repositioning and functional understanding. *Molec. Syst. Biol.* 9, 1 (2013), 662. DOI: <https://doi.org/10.1038/msb.2013.20>
- [49] Adán José-García and Wilfrido Gómez-Flores. 2016. Automatic clustering using nature-inspired metaheuristics: A survey. *Appl. Soft Comput.* 41 (2016), 192–213. DOI: <https://doi.org/10.1016/j.asoc.2015.12.001>
- [50] Adán José-García, Julia Handl, Wilfrido Gómez-Flores, and Mario Garza-Fabre. 2019. Many-view clustering: An illustration using multiple dissimilarity measures. In *Genetic and Evolutionary Computation Conference (GECCO'19)*. ACM Press, 213–214. DOI: <https://doi.org/10.1145/3319619.3323365>
- [51] Adán José-García, Julia Handl, Wilfrido Gómez-Flores, and Mario Garza-Fabre. 2021. An evolutionary many-objective approach to multiview clustering using feature and relational data. *Appl. Soft Comput.* 108 (2021), 107425. DOI: <https://doi.org/10.1016/j.asoc.2021.107425>
- [52] Adán José-García, Julie Jacques, Vincent Sobanski, and Clarisse Dhaenens. 2023. *Biclustering Algorithms based on Metaheuristics: A Review*. Springer Nature, 39–71. DOI: https://doi.org/10.1007/978-981-19-3888-7_2
- [53] Adán José-García, Julie Jacques, Alexandre Filiot, Julia Handl, David Launay, Vincent Sobanski, and Clarisse Dhaenens. 2022. Multi-view clustering of heterogeneous health data: Application to systemic sclerosis. In *Parallel Problem Solving from Nature – PPSN XVII*, Vol. 13399. Springer International Publishing, Dortmund, Germany, 352–367. DOI: https://doi.org/10.1007/978-3-031-14721-0_25

- [54] James Kennedy and R. Eberhart. 1995. Particle swarm optimization. In *International Conference on Neural Networks*, Vol. 4. IEEE, 1942–1948. DOI : <https://doi.org/10.1109/icnn.1995.488968>
- [55] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. 1983. Optimization by simulated annealing. *Science* 220, 4598 (1983), 671–680. DOI : <https://doi.org/10.2307/1690046>
- [56] Y. Kluger. 2003. Spectral biclustering of microarray data: Coclustering genes and conditions. *Genome Res.* 13, 4 (2003), 703–716. DOI : <https://doi.org/10.1101/gr.648603>
- [57] Zhoufan Kong, Qinghua Huang, and Xuelong Li. 2019. Bi-phase evolutionary biclustering algorithm with the NSGA-II algorithm. In *IEEE International Conference on Advanced Robotics and Mechatronics (ICARM'19)*. IEEE, 146–149. DOI : <https://doi.org/10.1109/icarm.2019.8834068>
- [58] Manuel Laguna and Rafael Martí. 2006. Scatter search. In *Metaheuristic Procedures for Training Neural Networks*. Springer, 139–152. DOI : https://doi.org/10.1007/0-387-33416-5_7
- [59] Annu Lambora, Kunal Gupta, and Kriti Chopra. 2019. Genetic algorithm—A literature review. In *International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon'19)*. 380–384. DOI : <https://doi.org/10.1109/COMITCon.2019.8862255>
- [60] Mohsen Lashkargir, S. Amirhassan Monadjemi, and Ahmad Baraani Dastjerdi. 2009. A new biclustering method for gene expression data based on adaptive multiobjective particle swarm optimization. In *International Conference on Computer and Electrical Engineering*. IEEE, 559–563. DOI : <https://doi.org/10.1109/iccee.2009.245>
- [61] Guojun Li, Qin Ma, Haibao Tang, Andrew H. Paterson, and Ying Xu. 2009. QUBIC: A qualitative biclustering algorithm for analyses of gene expression data. *Nucleic Acids Res.* 37, 15 (2009), e101–e101. DOI : <https://doi.org/10.1093/nar/gkp491>
- [62] Qin Lin, Huailing Zhang, Xizhao Wang, Yun Xue, Hongxin Liu, and Changwei Gong. 2019. A novel parallel biclustering approach and its application to identify and segment highly profitable telecom customers. *IEEE Access* 7 (2019), 28696–28711. DOI : <https://doi.org/10.1109/ACCESS.2019.2898644>
- [63] Junwan Liu, Zhoujun Li, Xiaohua Hu, and Yiming Chen. 2009. Biclustering of microarray data with MOSPO based on crowding distance. *BMC Bioinform.* 10, S4 (2009), S9. DOI : <https://doi.org/10.1186/1471-2105-10-s4-s9>
- [64] Junwan Liu, Zhoujun Li, Xiaohua Hu, and Yiming Chen. 2009. Multi-objective ant colony optimization biclustering of microarray data. In *IEEE International Conference on Granular Computing*. IEEE, 424–429. DOI : <https://doi.org/10.1109/grc.2009.5255086>
- [65] Junwan Liu, Zhoujun Li, Feifei Liu, and Yiming Chen. 2008. Multi-objective particle swarm optimization biclustering of microarray data. In *IEEE International Conference on Bioinformatics and Biomedicine*. IEEE, 363–366. DOI : <https://doi.org/10.1109/bibm.2008.17>
- [66] X. Liu and L. Wang. 2007. Computing the maximum similarity bi-clusters of gene expression data. *Bioinformatics* 23, 1 (2007), 50–56. DOI : <https://doi.org/10.1093/bioinformatics/btl560>
- [67] Yin Lu and Yongguo Liu. 2015. Biclustering of the gene expression data by coevolution cuckoo search. *Int. J. Bioautom.* 19, 2 (2015), 161–176.
- [68] Ons Maâtouk, Wassim Ayadi, HEND Bouziri, and Béatrice Duval. 2014. Evolutionary algorithm based on new crossover for the biclustering of gene expression data. In *International Conference on Pattern Recognition in Bioinformatics*. Springer, 48–59. DOI : https://doi.org/10.1007/978-3-319-09192-1_5
- [69] Ons Maâtouk, Wassim Ayadi, HEND Bouziri, and Béatrice Duval. 2019. Evolutionary biclustering algorithms: An experimental study on microarray data. *Soft Comput.* 23, 17 (2019), 7671–7697. DOI : <https://doi.org/10.1007/s00500-018-3394-4>
- [70] Ons Maâtouk, Wassim Ayadi, HEND Bouziri, and Béatrice Duval. 2021. Evolutionary local search algorithm for the biclustering of gene expression data based on biological knowledge. *Appl. Soft Comput.* 104 (2021), 107177. DOI : <https://doi.org/10.1016/j.asoc.2021.107177>
- [71] Ons Maatouk, Emna Ayari, HEND Bouziri, and Wassim Ayadi. 2022. BOBEA: A bi-objective biclustering evolutionary algorithm for genome-wide association analysis. In *Genetic and Evolutionary Computation Conference Companion (GECCO'22)*. Association for Computing Machinery, New York, NY, 344–347. DOI : <https://doi.org/10.1145/3520304.3528802>
- [72] Sara C. Madeira and Arlindo L. Oliveira. 2004. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 1, 1 (2004), 24–45.
- [73] Ujjwal Maulik, Anirban Mukhopadhyay, and Sanghamitra Bandyopadhyay. 2009. Finding multiple coherent biclusters in microarray data using variable string length multiobjective genetic algorithm. *IEEE Trans. Inf. Technol. Biomed.* 13, 6 (2009), 969–975. DOI : <https://doi.org/10.1109/titb.2009.2017527>
- [74] Ujjwal Maulik, Anirban Mukhopadhyay, Sanghamitra Bandyopadhyay, Michael Q. Zhang, and Xuegong Zhang. 2008. Multiobjective fuzzy biclustering in microarray data: Method and a new performance measure. In *IEEE Congress on Evolutionary Computation*. IEEE, 1536–1543. DOI : <https://doi.org/10.1109/cec.2008.4630996>

- [75] Sushmita Mitra and Haider Banka. 2006. Multi-objective evolutionary biclustering of gene expression data. *Pattern Recog.* 39, 12 (2006), 2464–2477. DOI: <https://doi.org/10.1016/j.patcog.2006.03.003>
- [76] A. Mukhopadhyay, Ujjwal Maulik, and Sanghamitra Bandyopadhyay. 2009. A novel coherence measure for discovering scaling biclusters from gene expression data. *J. Bioinform. Comput. Biol.* 07, 05 (2009), 853–868. DOI: <https://doi.org/10.1142/s0219720009004370>
- [77] N. Narmadha and R. Rathipriya. 2021. Gene ontology analysis of gene expression data using hybridized PSO triclustering. In *Machine Learning and Big Data Analytics Paradigms: Analysis, Applications and Challenges*. Springer, 437–466. DOI: https://doi.org/10.1007/978-3-030-59338-4_22
- [78] Juan A. Nepomuceno, Alicia Troncoso, and Jesús S. Aguilar-Ruiz. 2010. Evolutionary metaheuristic for biclustering based on linear correlations among genes. In *ACM Symposium on Applied Computing*. ACM Press, 1143. DOI: <https://doi.org/10.1145/1774088.1774329>
- [79] Juan A. Nepomuceno, Alicia Troncoso, and Jesús S. Aguilar-Ruiz. 2009. A hybrid metaheuristic for biclustering based on scatter search and genetic algorithms. In *International Conference on Pattern Recognition in Bioinformatics*. Springer, 199–210. DOI: https://doi.org/10.1007/978-3-642-04031-3_18
- [80] Juan A. Nepomuceno, Alicia Troncoso, and Jesús S. Aguilar-Ruiz. 2011. Biclustering of gene expression data by correlation-based scatter search. *BioData Min.* 4, 1 (2011), 3. DOI: <https://doi.org/10.1186/1756-0381-4-3>
- [81] Juan A. Nepomuceno, Alicia Troncoso, and Jesús S. Aguilar-Ruiz. 2015. Scatter search-based identification of local patterns with positive and negative correlations in gene expression data. *Appl. Soft Comput.* 35 (2015), 637–651. DOI: <https://doi.org/10.1016/j.asoc.2015.06.019>
- [82] Juan A. Nepomuceno, Alicia Troncoso, Isabel A. Nepomuceno-Chamorro, and Jesús S. Aguilar-Ruiz. 2015. Integrating biological knowledge based on functional annotations for biclustering of gene expression data. *Comput. Meth. Prog. Biomed.* 119, 3 (2015), 163–180. DOI: <https://doi.org/10.1016/j.cmpb.2015.02.010>
- [83] Juan A. Nepomuceno, Alicia Troncoso, Isabel A. Nepomuceno-Chamorro, and Jesús S. Aguilar-Ruiz. 2018. Pairwise gene GO-based measures for biclustering of high-dimensional expression data. *BioData Min.* 11, 1 (2018), 4. DOI: <https://doi.org/10.1186/s13040-018-0165-9>
- [84] Kath Nicholls and Chris Wallace. 2021. Comparison of sparse biclustering algorithms for gene expression datasets. *Brief. Bioinform.* 22, 6 (05 2021). DOI: <https://doi.org/10.1093/bib/bbab140>
- [85] Patryk Orzechowski and Boryczko Krzysztof. 2016. Text mining with hybrid biclustering algorithms. In *Artificial Intelligence and Soft Computing*. Springer, 102–113. DOI: https://doi.org/10.1007/978-3-319-39384-1_9
- [86] Patryk Orzechowski, Moshe Sipper, Xiuzhen Huang, and Jason H. Moore. 2018. EBIC: An evolutionary-based parallel biclustering algorithm for pattern discovery. *Bioinformatics* 34, 21 (05 2018), 3719–3726. DOI: <https://doi.org/10.1093/bioinformatics/bty401>
- [87] P. Orzechowski, K. Boryczko, and J. H. Moore. 2019. Scalable biclustering—The future of big data exploration? *Gigascience* 7 (2019), giz078. DOI: <https://doi.org/10.1093/gigascience/giz078>
- [88] Victor A. Padilha and Ricardo J. G. B. Campello. 2017. A systematic comparative evaluation of biclustering techniques. *BMC Bioinform.* 18, 1 (2017), 1–25. DOI: <https://doi.org/10.1186/s12859-017-1487-1>
- [89] Beatriz Pontes, Raúl Giráldez, and Jesús S. Aguilar-Ruiz. 2013. Configurable pattern-based evolutionary biclustering of gene expression data. *Algor. Molec. Biol.* 8, 1 (2013), 4. DOI: <https://doi.org/10.1186/1748-7188-8-4>
- [90] Beatriz Pontes, Raúl Giráldez, and Jesús S. Aguilar-Ruiz. 2015. Biclustering on expression data: A review. *J. Biomed. Inform.* 57 (2015), 163–180. DOI: <https://doi.org/10.1016/j.jbi.2015.06.028>
- [91] Beatriz Pontes, Raúl Giráldez, and Jesús S. Aguilar-Ruiz. 2015. Quality measures for gene expression biclusters. *PLoS One* 10 (2015), e0115497. DOI: <https://doi.org/10.1371/journal.pone.0115497>
- [92] Amela Prelić, Stefan Bleuler, Philip Zimmermann, Anja Wille, Peter Bühlmann, Wilhelm Gruissem, Lars Hennig, Lothar Thiele, and Eckart Zitzler. 2006. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics* 22, 9 (2006), 1122–1129. DOI: <https://doi.org/10.1093/bioinformatics/btl060>
- [93] Qingfu Zhang and Hui Li. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evolut. Comput.* 11, 6 (2007), 712–731. DOI: <https://doi.org/10.1109/tevc.2007.892759>
- [94] Qinghua Huang, Dacheng Tao, Xuelong Li, and A. Liew. 2012. Parallelized evolutionary learning for detection of biclusters in gene expression data. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 9, 2 (2012), 560–570. DOI: <https://doi.org/10.1109/tcb.2011.53>
- [95] R. Rathipriya, K. Thangavel, and J. Bagyamani. 2011. Binary particle swarm optimization based biclustering of web usage data. *Int. J. Comput. Applic.* 25, 2 (2011), 43–49. DOI: <https://doi.org/10.5120/3001-4036>
- [96] David J. Reiss, Christopher L. Plaisier, Wei-Ju Wu, and Nitin S. Baliga. 2015. cMonkey2: Automated, systematic, integrated detection of co-regulated gene modules for any organism. *Nucleic Acids Res.* 43, 13 (2015), e87–e87. DOI: <https://doi.org/10.1093/nar/gkv300>
- [97] Margarita Reyes-Sierra and Carlos A. Coello Coello. 2006. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *Int. J. Comput. Intell. Res.* 2, 3 (2006), 287–308.

- [98] Domingo S. Rodriguez-Baena, Antonio J. Perez-Pulido, and Jesús S. Aguilar-Ruiz. 2011. A biclustering algorithm for extracting bit-patterns from binary datasets. *Bioinformatics* 27, 19 (2011), 2738–2745. DOI : <https://doi.org/10.1093/bioinformatics/btr464>
- [99] Pracheta Sahoo, Sudipta Acharya, and Sriparna Saha. 2016. Automatic generation of biclusters from gene expression data using multi-objective simulated annealing approach. In *International Conference on Pattern Recognition*. IEEE, 2174–2179. DOI : <https://doi.org/10.1109/icpr.2016.7899958>
- [100] Rodrigo Santamaría, Luis Quintales, and Roberto Therón. 2007. Methods to bicluster validation and comparison in microarray data. In *Intelligent Data Engineering and Automated Learning (IDEAL '07)*. 780–789. DOI : https://doi.org/10.1007/978-3-540-77226-2_78
- [101] Khedidja Seridi, Laetitia Jourdan, and El-Ghazali Talbi. 2011. Multi-objective evolutionary algorithm for biclustering in microarrays data. In *IEEE Congress of Evolutionary Computation*. IEEE, 2593–2599. DOI : <https://doi.org/10.1109/cec.2011.5949941>
- [102] Khedidja Seridi, Laetitia Jourdan, and El-Ghazali Talbi. 2015. Using multiobjective optimization for biclustering microarray data. *Appl. Soft Comput.* 33 (2015), 239–249. DOI : <https://doi.org/10.1016/j.asoc.2015.03.060>
- [103] Akdes Serin and Martin Vingron. 2011. DeBi: Discovering differentially expressed biclusters using a frequent itemset approach. *Algor. Molec. Biol.* 6, 1 (2011), 18. DOI : <https://doi.org/10.1186/1748-7188-6-18>
- [104] Andrey A. Shabalin, Victor J. Weigman, Charles M. Perou, and Andrew B. Nobel. 2009. Finding large average submatrices in high dimensional data. *Ann. Appl. Statist.* 3, 3 (2009), 985–1012. DOI : <https://doi.org/10.1214/09-aos239>
- [105] Miguel G. Silva, Sara C. Madeira, and Rui Henriques. 2022. Water consumption pattern analysis using biclustering: When, why and how. *Water* 14, 12 (2022). DOI : <https://doi.org/10.3390/w14121954>
- [106] M. Srinivas and L. M. Patnaik. 1994. Genetic algorithms: A survey. *Computer* 27, 6 (1994), 17–26. DOI : <https://doi.org/10.1109/2.294849>
- [107] Rainer Storn and Kenneth Price. 1997. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* 11, 4 (1997), 341–359.
- [108] Jianjun Sun and Qinghua Huang. 2023. Two stages biclustering with three populations. *Biomed. Sig. Process. Contr.* 79 (2023), 104182. DOI : <https://doi.org/10.1016/j.bspc.2022.104182>
- [109] Jianjun Sun and Yu Zhang. 2022. Recommendation system with biclustering. *Big Data Min. Analyt.* 5, 4 (2022), 282–293. DOI : <https://doi.org/10.26599/BDMA.2022.9020012>
- [110] P. SwathyPriyadharsini and K. Premalatha. 2023. Hybrid cuckoo search with clonal selection for triclustering gene expression data of breast cancer. *IETE Journal of Research* 69, 5 (2023), 1–9. DOI : [10.1080/03772063.2021.1911691](https://doi.org/10.1080/03772063.2021.1911691)
- [111] El Ghazali Talbi. 2009. *Metaheuristics from Design to Implementation*. John Wiley & Sons.
- [112] A. Tanay, R. Sharan, and R. Shamir. 2002. Discovering statistically significant biclusters in gene expression data. *Bioinformatics* 18, Suppl 1 (2002), S136–s144. DOI : https://doi.org/10.1093/bioinformatics/18.suppl_1.S136
- [113] Li Teng and Laiwan Chan. 2008. Discovering biclusters by iteratively sorting with weighted correlation coefficient in gene expression data. *J. Sig. Process. Syst.* 50 (2008), 267–280. DOI : <https://doi.org/10.1007/s11265-007-0121-2>
- [114] K. Thangavel, J. Bagyamani, and R. Rathipriya. 2012. Novel hybrid PSO-SA model for biclustering of expression data. *Proced. Eng.* 30 (2012), 1048–1055. DOI : <https://doi.org/10.1016/j.proeng.2012.01.962>
- [115] Maxence Vandromme, Julie Jacques, Julien Taillard, Laetitia Jourdan, and Clarisse Dhaenens. 2020. A biclustering method for heterogeneous and temporal medical data. *IEEE Trans. Knowl. Data Eng.* 34, 2 (2020), 506–518.
- [116] Nishchal K. Verma, Teena Sharma, Sonal Dixit, Pooja Agrawal, Sourya Sengupta, and Vikas Singh. 2021. BIDEAL: A toolbox for bicluster analysis—generation, visualization and validation. *SN Comput. Sci.* 2, 1 (2021), 24. DOI : <https://doi.org/10.1007/s42979-020-00411-9>
- [117] Jiasheng Wang, Hong Song, and Xiaofeng Zhou. 2015. A collaborative filtering recommendation algorithm based on biclustering. In *IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER'15)*. 803–807. DOI : <https://doi.org/10.1109/CYBER.2015.7288046>
- [118] Shan Wang, Yanbin Yin, Qin Ma, Xiaojia Tang, Dongyun Hao, and Ying Xu. 2012. Genome-scale identification of cell-wall related genes in Arabidopsis based on co-expression network analysis. *BMC Plant Biol.* 12, 1 (2012), 138. DOI : <https://doi.org/10.1186/1471-2229-12-138>
- [119] Yi Kan Wang, Cristin G. Print, and Edmund J. Crampin. 2013. Biclustering reveals breast cancer tumour subgroups with common clinical features and improves prediction of disease recurrence. *BMC Genom.* 14, 1 (2013), 102. DOI : <https://doi.org/10.1186/1471-2164-14-102>
- [120] Juan Xie, Anjun Ma, Anne Fennell, Qin Ma, and Jing Zhao. 2019. It is time to apply biclustering: A comprehensive review of biclustering applications in biological and biomedical data. *Brief. Bioinform.* 20, 4 (2019), 1450–1465. DOI : <https://doi.org/10.1093/bib/bby014>
- [121] Min Xiong, Bin Li, Qiang Zhu, Yun-Xing Wang, and Hong-Yu Zhang. 2014. Identification of transcription factors for drug-associated gene modules and biomedical implications. *Bioinformatics* 30, 3 (2014), 305–309. DOI : <https://doi.org/10.1093/bioinformatics/btt683>

- [122] Ruolin Yang and Xiangfeng Wang. 2013. Organ evolution in angiosperms driven by correlated divergences of gene sequences and expression patterns. *Plant Cell* 25, 1 (2013), 71–82. DOI: <https://doi.org/10.1105/tpc.112.106716>
- [123] Wen-Hui Yang, Dao-Qing Dai, and Hong Yan. 2011. Finding correlated biclusters from gene expression data. *IEEE Trans. Knowl. Data Eng.* 23, 4 (2011), 568–584. DOI: <https://doi.org/10.1109/TKDE.2010.150>
- [124] Xin-She Yang and Suash Deb. 2014. Cuckoo search: Recent advances and applications. *Neural Comput. Applic.* 24, 1 (2014), 169–174. DOI: <https://doi.org/10.1007/s00521-013-1367-1>
- [125] Lu Yin, Junlin Qiu, and Shangbing Gao. 2018. Biclustering of gene expression data using cuckoo search and genetic algorithm. *Int. J. Pattern Recog. Artif. Intell.* 32, 11 (2018), 1850039. DOI: <https://doi.org/10.1142/s0218001418500398>
- [126] K. Y. Yip, D. W. Cheung, and M. K. Ng. 2004. HARP: A practical projected clustering algorithm. *IEEE Trans. Knowl. Data Eng.* 16, 11 (2004), 1387–1397. DOI: <https://doi.org/10.1109/TKDE.2004.74>
- [127] J. Yoo and P. Hajela. 1999. Immune network simulations in multicriterion design. *Structur. Optim.* 18, 2-3 (1999), 85–94. DOI: <https://doi.org/10.1007/BF01195983>
- [128] Eckart Zitzler and Simon Künzli. 2004. Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature - PPSN VIII*, Vol. 3242. Springer, Birmingham, UK, 832–842. DOI: https://doi.org/10.1007/978-3-540-30217-9_84
- [129] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. 2001. *SPEA2: Improving the Strength Pareto Evolutionary algorithm*. Technical Report. Swiss Federal Institute Technology, Zurich, Switzerland.

Received 30 November 2022; revised 20 June 2023; accepted 21 August 2023