ORIGINAL RESEARCH

Bio-inspired multiobjective clustering optimization: A survey and a proposal

Danilo Cunha*[1], Dávila Cruz[1], Alexandre Politi[1], Leandro Nunes de Castro[1], Renato Dourado Maia[2]

[1] *Natural Computing and Machine Learning Laboratory, Computing and Informatics Faculty & Graduate Program in Electrical Engineering and Computing, Mackenzie Presbyterian University, São Paulo, Brazil*
[2] *Computer Science Department, State University of Montes Claros, Montes Claros, Brazil*

Received: November 30, 2016 **Accepted:** March 15, 2017 **Online Published:** March 27, 2017
DOI: 10.5430/air.v6n2p10 **URL:** https://doi.org/10.5430/air.v6n2p10

ABSTRACT

Multiobjective clustering techniques have been used to simultaneously consider several complementary aspects of clustering quality. They optimize two or more cluster validity indices simultaneously, they lead to high-quality results, and have emerged as attractive and robust alternatives for solving clustering problems. This paper provides a brief review of bio-Inspired multiobjective clustering, and proposes a bee-inspired multiobjective optimization (MOO) algorithm, named cOptBees-MO, to solve multiobjective data clustering problems. In its survey part, a brief tutorial on MOO and multiobjective clustering optimization (MOCO) is presented, followed by a review of the main works in the area. Particular attention is given to the many objective functions used in MOCO. To evaluate the performance of the algorithm it was executed for various datasets and the results presented high quality clusters, diverse solutions an the automatic determination of a suitable number of clusters.

Key Words: Clustering, Multiobjective optimization (MOO), Multiobjective clustering optimization (MOCO), Bee-inspired algorithm

1. INTRODUCTION

Data clustering consists of finding groups of data in datasets based on the similarities or dissimilarities of their objects.[1] The objects in a cluster may present high similarity among themselves and high dissimilarity to objects of the other clusters. To calculate the similarity of objects different measures can be used, depending on each problem. An important feature of the clustering task is its unsupervised nature, in other words, the information about to which cluster each object belongs is not known a priori. Considering the existence of several internal and external criteria to assess the quality of a clustering solution,[2] it is increasingly necessary to see clustering as a multiobjective problem, with conflicting objectives, such as the overall deviation and connectivity. The non-dominated set allows the observation of feasible solutions found by optimization algorithms by considering different objectives.

Population-based algorithms are often used to deal with multiobjective problems due to their ability to perform a guided search over the search space and to maintain diversity of non-dominated solutions. Some variations of multiobjective evolutionary algorithms use an external population that serves as a memory for the evolution process.[3] This population is used to store the best solutions found during all generations. This paper proposes a multiobjective clustering algorithm inspired by the foraging behavior of bee colonies,

*Correspondence: Danilo Cunha; Email: danilocunha85@gmail.com; Address: Computing and Informatics Faculty & Graduate Program in Electrical Engineering and Computing, Mackenzie Presbyterian University, São Paulo, Brazil.

called cOptBees-MO, which is a multiobjective version for the cOptBees clustering algorithm.[4] cOptBees is able to find optimal clustering within datasets, maintain the diversity of solutions and estimate the adequate number of clusters. The cOptBees-MO, as well as its parent algorithm, manipulates a swarm of artificial bees who play different roles during the search phase. A bee can act as a recruiter, recruited or a scout. Another important aspect is the application of local search operators that contribute to the better exploration of the search space. These and other features allow cOptBees-MO to be competitive when compared to other algorithms from the literature which require a smart initialization process or the number of clusters to be input a priori.

Performance experiments were conducted in two phases: The first with three machine learning consolidated datasets, whose results were compared to those reported by two other bioinspired algorithms from the literature; and in the second set of experiments seven well-known datasets were used and the results compared to other bio-inspired algorithms from the multiobjective clustering literature. The obtained results open up avenues for a number of new researches in the multiobjective clustering area.

This paper has the following organization. Section 2 introduces multiobjective optimization (MOO) and Section 3 presents multiobjective clustering optimization (MOCO). Section 4 presents a review of bioinspired multiobjective algorithms. Section 5 surveys the most common objective functions used in multiobjective clustering and Section 6 introduces the proposed multiobjective clustering algorithm. Section 7 shows the datasets and experimental results. Section 8 presents the conclusions with general comments about the results and perspectives for future investigations.

2. MULTIOBJECTIVE OPTIMIZATION

In the optimization context, there is a class of problems called multiobjective optimization (MOO)[5] whose main characteristic is the presence of two or more objective functions, which are usually in conflict with one another, that must be optimized simultaneously. This type of situation occurs naturally in many practical problems, for instance, in a generic design engineering it is common for its designers to want to minimize production costs and the time taken for completing the project, while still optimizing various performance indices. MOO problems consist of finding a set of decision variables which satisfies all the constraints of the problem and optimizes a set of objective functions simultaneously.[6] Formally, the MOO task is defined as follows:[5–7]

Definition 1: A MOO solution minimizes (or maximizes) $F(\overrightarrow{x}) = (f_1(\overrightarrow{x}), \cdots, f_n(\overrightarrow{x}))$ subject to: $g_i(\overrightarrow{x}) \leq 0$, $i = \{1, 2, \cdots, m\}$ and $h_j(\overrightarrow{x}) = 0, j = \{1, 2, \cdots, p\}$, $\overrightarrow{x} \in \Omega$, where $\overrightarrow{x} = (x_1, \cdots, x_d)$ is a d-dimensional decision variable vector within some universe Ω. Equations $g_i(\overrightarrow{x}) \leq 0$ and $h_j(\overrightarrow{x}) = 0$ represent constraints that must be met during the minimization (or maximization) of $F(\overrightarrow{x})$, and Ω contains the values of $\overrightarrow{x}$ that satisfy an evaluation of $F(\overrightarrow{x})$. The n objective functions may be continuous or discrete and linear or nonlinear.

A fundamental difference between a mono-objective and a multiobjective problem is the solution concept. In the mono-objective problem, an optimal solution corresponds to a vector x that satisfies all the constraints (a condition that characterizes it as a feasible solution) and finds an extreme value of the objective function (max or min). On the other hand, in multiobjective problems, the notion of optimal solution gives way to the concept of efficient solutions. An efficient solution is characterized by the absence of another feasible solution that improves simultaneously all objectives. That is, the improvement in one of the objective functions can only be achieved by degrading the value of at least one other objective. The efficient solution concept is formalized as follows:[5,6]

Definition 2: Vector $\overrightarrow{x^*} = (x_1^*, \cdots, x_d^*)$ is an efficient solution (in a minimizing problem) if and only if there is no $\overrightarrow{x} \in S$ such that $f_i(\overrightarrow{x}) \leq f_i(\overrightarrow{x^*})$ for all $i \in \{1, \cdots, n\}$, with at least one strict inequality, where S is the feasible search space.

Another common concept in MOO (Definition 3) is the weak dominated solution; which can be understood as a relaxation of Definition 2.

Definition 3: A vector $\overrightarrow{x^*} = (x_1^*, \cdots, x_d^*)$ is a weak efficient solution if and only if there is no $\overrightarrow{x} \in S$ such that $f_i(\overrightarrow{x}) < f_i(\overrightarrow{x^*}), \forall i \in \{1, \cdots, n\}$.

Classically, in mono-objective optimization, the process of searching the optimum point is strictly technical, in the sense that the solution is implicit in the model. Thus, the work of the solution algorithm is to just find it.

The notion of optimum changes for multiobjective problems, because MOO aims at finding good compromises among the objective functions instead of a single solution, as well as in global optimization.[6] This solution concept, no longer understood as a single point, but now as a number of great options, was initially proposed by Edgewrth[8] and later generalized by Pareto *et al.*,[9] thus being called Pareto Optimum Front, as formalized in Definition 4.[5–7]

Definition 4: The set P of all efficient solutions (including the weak solutions) is named Pareto optimal front.

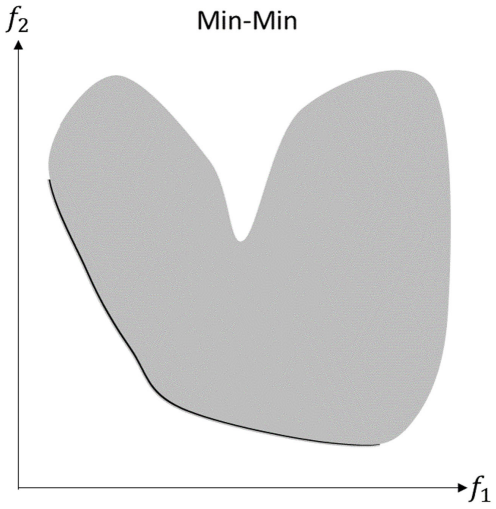

Figure 1. Example of Pareto optimal front (dark line) for two objective functions f_1 and f_2

In order to illustrate the concept of Pareto optimum front, Figure 1 shows an example with two objective functions ($f_1(\overrightarrow{x})$ and $f_2(\overrightarrow{x})$) to be minimized simultaneously.

In Figure 1, the shaded area represents the feasible region, that is, set of vector solutions that satisfies all constraints. The dark line on the edge is the Pareto optimum front. By defining a boundary in the space of the functions, this region is called the Pareto front.[5] It can be seen that the points on the Pareto set do not allow to choose between one solution and another without degrading one of the two functions. The best solution is subjective and an expert can choose one or more solutions according to the problem requirements.[5, 10, 11]

In a population-based algorithm a set of random points are started within the decision space, where each point represents a candidate solution for solving the multiobjective problem, the objective space has the mapped solution for the selected fitness functions.

Figure 2a represents the solutions inside the decision space as circles and the ones in the objective space as squares and stars. The stars are the candidate solutions inside the Pareto front (solutions which are non-dominated), and the squares are the dominated solutions. Figure 2b represents an intermediary step of the non-dominated set concept, where the candidate solutions move closer to the Pareto front according to movimetation in decision space and explore different areas, becoming part of the Pareto front. Finally, in Figure 2c, all candidate solutions appear over the Pareto front, exploring promising and heterogeneous areas in the decision space.

The algorithm's search phase is made over the decision space by considering all involved variables, meanwhile its evaluation phase is performed over the objective space by considering all objective functions involved and the Pareto dominance concept. Under the Pareto perspective, a solution $\overrightarrow{u}$ dominates another solution $\overrightarrow{v}$ if $\overrightarrow{u}$ is not worse than $\overrightarrow{v}$ in all objectives (fitness functions); and $\overrightarrow{u}$ is strictly dominated if it is better than $\overrightarrow{v}$ in at least one objective.

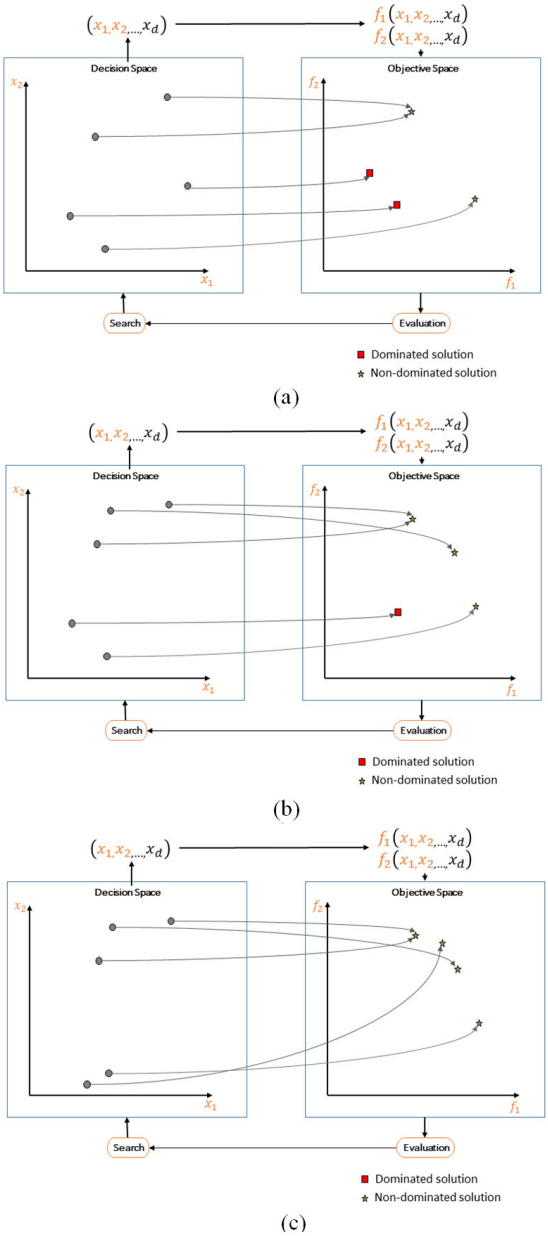

Figure 2. Decision space and objective space for population-based algorithms (adapted from Gruna[12] and Zitzler *et al.*[13])

As MOO provides advance in numerical optimization,[14] researchers soon started using it in different tasks, such as knowledge-discovery in databases, including feature selec-

tion,[15] association rule mining,[16] classification,[17] and clustering.[18]

3. Multiobjective clustering optimization (MOCO)

Multiobjective clustering is a process performed in two steps:[19] first, the independent or parallel discovery of clusters; and second the search for an optimal partition of the found clusters. Multiobjective clustering techniques have been used to simultaneously consider several complementary aspects of clustering quality.[20] They optimize two or more cluster validity indices (as well as the number of clusters) simultaneously, leading to high quality results, and have emerged as attractive and robust alternatives for clustering problems.[11] An important step to multiobjective clustering is the choice of suitable objective functions. Different cluster validity indices in different combinations are used in multiobjective clustering algorithms. In Mukhopadhyay *et al.*,[11] the authors divided the cluster validity indices into two groups: 1) Indices based on cluster prototypes; 2) Those based on cluster labels.

The objective function used by a clustering algorithm may not indicate the partitions' quality due to different arrangements of objects over the dataset. The quality of each cluster must be assessed by the clustering algorithm that generated it and an external evaluation criterion not used during the search phase in order to validate the final clusters found.[21,22]

The main approaches to MOO are:[23] 1) transformation of the multiobjective problem into a mono objective one by means of a weighted formula; 2) ordering the objectives according to their importance; and 3) the Pareto approach, where multiobjective functions are optimized simultaneously to generate a resultant set with a number of non-dominated solutions.[23] Population-based algorithms, for example genetic,[24,25] bee inspired[4,26] and ant colony[27] algorithms are particularly suited for approximating the entire Pareto front because they work with a pool of solutions rather than a single candidate solution. This enables approximating several solutions of the Pareto set simultaneously in a unique algorithm run.

In a multiobjective clustering task, a candidate solution is a vector $\overrightarrow{x} = (x_1, x_2, \cdots, x_d)^T$, where x_i represents a decision variable in the decision space in which the search occurs, and d is the problem dimension. The objective vector is represented by $f(\overrightarrow{x}) = (f_1(\overrightarrow{x}), f_2(\overrightarrow{x}), \cdots, f_n(\overrightarrow{x}))^T$, where $f_i(\overrightarrow{x})$ represents its coordinate in the objective space in which the results are evaluated. The Pareto dominance concept is used to compare the quality of the solutions, because the quality of a candidate solution is measured as a vector instead of a scalar. Figure 3 illustrates MOCO by using the non-dominance Pareto concept. The small circles are solutions whose qualities are measured by the non-dominance concept, in this case we are maximizing both objective functions. As detailed in Figure 3, each solution in the objective space has all variables from the decision space in the left hand side of the picture.

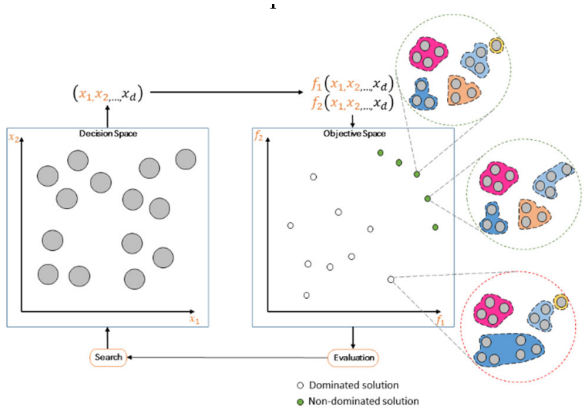

Figure 3. Decision space and objective space for population-based clustering algorithms

Table 1. Overview of different combinations of cluster validity indices to multiobjective evolutionary clustering (adapted from Mukhopadhyay *et al.*[11])

Objective functions	Cluster validity indices
Two-objective functions	Connectedness and cluster deviation
	J_m and XB
	Number of clusters and TWCV
	Cluster deviation and silhouette
	Intra-cluster entropy and cluster separation
	$\mathcal{I}$ and XB
	MinMaxCut and silhouette
	Cluster deviation and edge Index
	Normalized J_m and fuzzy separation
	DB and Dunn
	J_m and cluster separation
	Connectivity and cohesion
Three-objective functions	XB, $\mathcal{I}$ and J_m
	Average deviation, ABGSS and connectedness
Four-objective functions	Cluster separation, cluster dominance, maximum diameter and cluster deviation

4. MOCO: A review

Several multiobjective clustering approaches, bioinspired and non-bioinspired, have been proposed in the literature. The most popular one is the multiobjective evolutionary clustering algorithm.[11] In Mukhopadhyay *et al.*,[11] the authors present a review of multiobjective evolutionary clustering and introduce the main concepts of the area, such as the encoding strategies, choice of objective functions, effects of

objective functions on the encoding strategies, main evolutionary operators and different approaches for generating the final solution. The authors also present an overview of different combinations of cluster validity indices normally used in multiobjective evolutionary clustering algorithms, and that can be extended to other non-evolutionary approaches, present in the literature, as summarized in Table 1.

In the context of evolutionary algorithms, most multiobjective clustering algorithms are based on genetic algorithms.[11] The genetic algorithms (GAs), firsly, were applied to solve mono objective optimization problems. The simple genetic algorithm (SGA) is used when an optimization problem has an unique objective. In the real world, however, problems appear to be more complex and most of them require two or more objectives, which are to be optimized at the same time.

There are other evolutionary approaches also employed to solve such problems, such as differential evolution,[28,29] genetic programming,[30] and gene expression programing.[31] Other bioinspired methods have been used for multiobjective clustering, such as ant colony optimization,[32,33] artificial immune systems,[34] particle swarm optimization,[35] among others. Some of these works are summarized below. No work was found applying neural networks and bee inspired algorithms for solving multiobjective clustering tasks. Concerning neural networks, the self organizing maps have been addressed as useful tools for visualizing and analyzing the Pareto front.[36–39]

In Santos *et al.*,[32] it was presented an algorithm, called multi ant colony clustering algorithm (MACC), which works with two artificial colonies, each one with k bees working in parallel, to cluster a dataset taking account two different objectives: compactness and connectedness. Experiments were performed with the Iris dataset from UCI to assess the quality of the ant clustering algorithm for multiobjective problems, and the results were compared with other results present in the literature.

In Wu *et al.*,[33] the authors proposed the multiobjective ant colony optimization (MOACO) algorithm to address the multiobjective clustering problem in mobile ad hoc nets (MANETs). The algorithm optimizes three objectives simultaneously: number of cluster heads (an object that represents a group); number of nodes covered by each cluster head; and total power consumption (proportional to the distance between two communicating nodes). The performance of the MOACO was compared with three algorithms and the experimental results showed that MOACO can find a set of high-quality solutions.

The multiobjective k-means genetic algorithm (MOKGA) was presented in Ozyer *et al.*[40] The algorithm consists of an iterative process that integrates weighted k-means with a multiobjective genetic algorithm. The algorithm was applied to a well-known dataset from the literature. Two objective functions were used: inter-cluster distance and intra-cluster distance.

In Matake *et al.*,[41] the authors presented a multiobjective clustering algorithm for web mining. They applied multiobjective clustering with automatic determination of k (MOCK), a clustering algorithm based on a multiobjective genetic algorithm presented in Handl and Knowles.[42] Two objective functions were used: connectivity (based on cluster connectedness); and overall deviation (based on cluster compactness). The algorithm was tested in nine datasets present in the literature and the results were compared with k-means, agglutination methods, and other evolutionary algorithms.

In Bandyopadhyay and Saha,[43] the Variable String Length Point Symmetry-Based (VGAPS) clustering technique was presented. This technique is able to encoding of a variable number of clusters by means a variable string length GA. The Sym-index was used as the fitness function and it provides the most approximate partitioning even when the number of clusters is varied.

In Saha and Bandyopadhyay,[44] it was presented the GenClustMOO, a multiobjective clustering technique that optimizes three objective functions: compactness; total symmetry of the clusters; and connectedness. A simulated annealing based on a MOO method was used to optimizes the objective functions. The results were compared with that of MOCK, a single objective genetic algorithm based on an automatic clustering technique (VGAPS-clustering), k-means, and single linkage clustering. The technique was applied to nineteen artificial and seven real-world datasets.

The GenClustPESA2 algorithm,[44] has exactly the same approach as GenClustMOO, with the underlying MOO strategy replaced by the Pareto envelope-based selection algorithm II (PESA-II).

In Yang *et al.*,[45] the authors proposed an immune MOO algorithm, which has the features of adaptive rank clones and diversity maintenance by k-nearest-neighbor, besides incorporating two fuzzy clustering validity indices that are optimized simultaneously.

An additional difficulty in multiobjective problems is the visualization of the Pareto front for cases where there are more than three objective functions.[36] This is important to a more clear understanding of the tradeoff among the potential solutions.[37] A very promising approach, based on Self Organizing Maps, for visualizing the Pareto front has been

studied by several authors.[36–39]

The robustness and population-based nature of swarm intelligence (SI) algorithms are important and attractive features for MOO problems.[46] In the literature, different multiobjective clustering algorithms based on SI are presented.

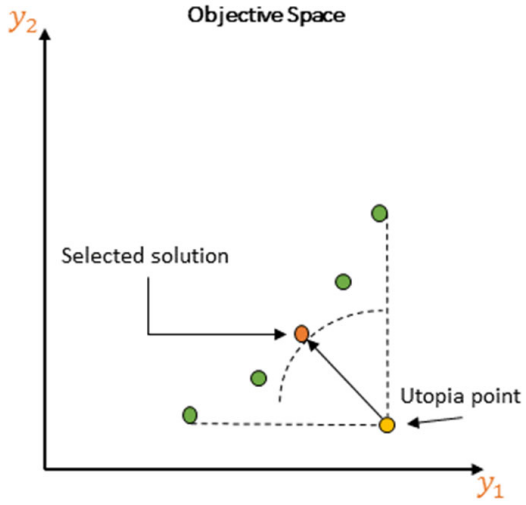

Figure 4. Distance technique to find out the final solution as the closest solution in the Pareto set to the utopia point (Adapted from Armano *et al.*[47]

The multiobjective clustering with particle swarm optimization (MCPSO) method,[47,48] has two main phases: optimization and decision making. In the first phase, two conflicting objective functions, based on connectivity and cohesion,are defined. The choose of the objective functions aims of obtaining well-separated, compact and connected clusters. In the end, the optimization phase products a set of optimal solutions for the clustering problem, known as the Pareto set solutions.[49] A distance-based technique is used to select the solution in the Pareto set solutions with the minimal distance from an ideal solution, called utopia point, which is illustrated in Figure 4. The connectivity is represented in the y_1 axis and it tends to be maximized meanwhile the cohesion is represented in the axis y_2 and it tends to be minimized.

The utopia location is defined as the intersection point of the lines passing through the top right and left solutions of the Pareto front in the area of all possible outcomes. Properly speaking, the coordinates of the utopia point are the best values obtained for the objective functions during the swarm optimization process.

5. Cluster Validity Criteria

An important aspect of multiobjective clustering is the choice of reliable objective functions that are to be optimized simultaneously. In general, cluster validity indices are chosen as the objective functions.[20,50] Several validity indices in different arrangements have been used in bio-inspired multiobjective clustering algorithms. These indices can be divided into two types:[11] 1) Based on cluster prototypes (internal indices); and 2) based on cluster labels (external indices). Table 2 summarizes the symbols used in the indices reviewed.

Table 2. Definitions for symbols in the cluster validity indices

Symbol	Description
n	Number of objects
d	Number of attributes
K	Number of clusters
m	Fuzzy exponent
$\vec{\mathbf{x}}_i$	i-th object
$\vec{\mathbf{z}}_k$	k-th cluster prototype
u_{ki}	Membership degree of the i-th object to the k-th cluster
$D(.,.)$	Distance function
C	Set of all clusters
C_k	k-th cluster

5.1 Internal indices

This section provides details of cluster validity indices based on cluster prototypes.

5.1.1 *Overall cluster deviation (Dev(C))*

The Dev(C)[51] is the summed distances between all objects and their corresponding cluster center:

$$Dev(C) = \sum_{k=1}^{K} \sum_{i=1}^{n_k} D(\vec{z_k}, \vec{x_i}) \tag{1}$$

where K is the number of cluster centroids in a clustering C; n_k is the number of objects in cluster k; $\vec{z_k}$ is the k-th cluster centroid; and $D(.,.)$ is the Euclidian distance. The overall deviation is to be minimized in order to obtain compact clusters. $Dev(C) \in [0,1]$, where values closer to 0 represent better clusters. This objective is similar to the objective of the k-means clustering.

5.1.2 J_m *index*

The J_m index[52] is formulated to be minimized by fuzzy C-means clustering:

$$J_m = \sum_{k=1}^{K} \sum_{i=1}^{n_k} u_{k_i}^m D(\vec{z_k}, \vec{x_i}) \tag{2}$$

where $D(\vec{z_k}, \vec{x_i})$ is the distance of the i-th data point $\vec{x_i}$ to the k-th cluster center $\vec{z_k}$. J_m represents the global fuzzy cluster variance. Smaller values of J_m correspond to more compact clusters and its value depends on the number of clusters K, gradually decreasing with an increase in K. Note

that J_m takes its minimum value 0 for $K = n$.[50] The normalized version of the J_m index, denoted by J_m, is defined as:

$$J_m = \sum_{k=i}^{K} \frac{\sum_{i=1}^{n_k} u_{ki}^m D(\overrightarrow{z_k}, \overrightarrow{x_i})}{\sum_{i=1}^{n_k} u_{ki}} \quad (3)$$

This one is also to be minimized to obtain more compact clusters.

5.1.3 *Davies-Bouldin (DB)*

The DB index[50] is a function of the ratio of the sum of within-cluster scatter to between-cluster separation. The scatter within the *i*-th cluster, S_i, is computed as:

$$S_k = \frac{1}{n_k} \sum_{i=1}^{n_k} D(\overrightarrow{z_k}, \overrightarrow{x_i}) \quad (4)$$

where n_k is the number of objects in cluster k.

The distance between two clusters C_i and C_j, d_{ij} is defined as the distance between their centers:

$$d_{ij} = D(\overrightarrow{z_i}, \overrightarrow{z_j}) \quad (5)$$

The DB index is then defined as:

$$DB = \frac{1}{K} \sum_{i=1}^{K} R_i \quad (6)$$

where

$$R_i = \max_{j, j \neq i} \left\{ \frac{S_i + S_j}{d_{ij}} \right\} \quad (7)$$

The DB index value must be minimized to achieve proper clustering.

5.1.4 *Xie-Beni (XB)*

The XB index is a function of the ratio of the total fuzzy cluster variance σ to the minimum separation (sep) of the clusters. Here σ and sep are formulated as:

$$\sigma = \sum_{k=1}^{K} \sum_{i=1}^{n_k} u_{k_i}^2 D(\overrightarrow{z_k}, \overrightarrow{x_i}) \quad (8)$$

$$sep = \min_{k \neq l} \{D(\overrightarrow{z_k}, \overrightarrow{z_l})\} \quad (9)$$

The XB index is then written as:

$$XB = \frac{\sigma}{n \times sep} = \frac{\sum_{k=1}^{K} \sum_{i=1}^{n_k} u_{k_i}^2 D(\overrightarrow{z_k}, \overrightarrow{x_i})}{n \times \min_{k \neq l} \{D(\overrightarrow{z_k}, \overrightarrow{z_l})\}} \quad (10)$$

In compact and well-separated clusterings, the value of σ should be small, while sep should be large, thereby giving smaller values of the XB index. Therefore, the objective is to minimize the XB index for achieving a proper clustering.[50]

5.1.5 *Intracluster entropy*

The intracluster entropy (H)[53] measures the average purity of clusters without using the class labels of the objects:

$$H = \sum_{i=1}^{K} [1 - H(C_i) \cdot g(\overrightarrow{x_i})]^{\frac{1}{k}} \quad (11)$$

where

$$H(C_i) = -\{g(\overrightarrow{z_i}) \log_2 g(\overrightarrow{z_i}) + [1 - g(\overrightarrow{z_i})] \log_2 [1 - g(\overrightarrow{z_i})]\} \quad (12)$$

The average similarity between a cluster center z_i and the objects belonging to cluster i is defined as:

$$g(\overrightarrow{z_i}) = \frac{1}{n} \sum_{j=1}^{n} \frac{1 + CO(\overrightarrow{z_i}, \overrightarrow{x_j})}{2} \quad (13)$$

where CO is the cosine distance defined as:

$$CO(\overrightarrow{z_i}, \overrightarrow{x_j}) = \frac{\langle \overrightarrow{z_i} \cdot \overrightarrow{x_j} \rangle}{|\overrightarrow{z_i}| \cdot |\overrightarrow{x_j}|} \quad (14)$$

Index H has to be maximized to obtain homogeneous clusters.

5.1.6 *Cluster separation*

The cluster separation (Sep) index, or intercluster distance, is computed as the average distance among the cluster centers:

$$Sep(C) = \frac{1}{K(K-1)} \sum_{i=1}^{K} \sum_{j=1,\ j \neq i}^{K} D(\overrightarrow{z_i}, \overrightarrow{z_j}) \quad (15)$$

$Sep(C)$ must be maximized to obtain well-separated clusters.

A fuzzy version of cluster separation is also available. The fuzzy separation S is computed as follows: the center $\overrightarrow{z_i}$ of the i-th cluster is assumed to be the center of a fuzzy set $\{\overrightarrow{z_j} \mid 1 \leq j \leq K, j \neq i\}$. Thus, the membership degree of each $\overrightarrow{z_i}$ to $\overrightarrow{z_j}$, $j \neq i$ is computed as:

$$u_{ij} = \frac{1}{\sum_{l=1,\ l \neq j}^{K} [\frac{D(\overrightarrow{z_j}, \overrightarrow{z_i})}{D(\overrightarrow{z_j}, \overrightarrow{z_l})}]^{\frac{2}{m-1}}}, i \neq j \quad (16)$$

Then, the fuzzy separation is defined as:

$$S = \sum_{i=1}^{K} \sum_{j=1,\ j \neq i}^{K} u_{ij}^m D(\overrightarrow{z_i}, \overrightarrow{z_j}) \quad (17)$$

5.1.7 *Average between group sum of squares (ABGSS)*

ABGSS is a variant of the cluster separation measure.[54] This computes the average distance of the cluster centers from the centroid of the dataset as:

$$ABGSS = \frac{\sum_{i=1}^{K} n_i \cdot D(\overrightarrow{z_i}, \overrightarrow{\overrightarrow{z}})}{K} \quad (18)$$

where $\overrightarrow{\overrightarrow{z}}$ is the dataset center, that is, the mean of all objects, and n_i is the number of objects in cluster i. The goal is to maximize $ABGSS$ to obtain well-separated clusters.

5.1.8 $\jmath$ *index*

The $\jmath$ index is defined as follows:

$$\jmath = \left(\frac{1}{K} \times \frac{E_1}{E_K} \times D_k\right)^{\gamma} \tag{19}$$

where

$$E_K = \sum_{k=1}^{K} \sum_{j=1}^{n_k} u_{ki} D(\overrightarrow{z_k}, \overrightarrow{x_j}) \tag{20}$$

$$D_K = max_{i,j=1}^{K} \{D(\overrightarrow{z_i}, \overrightarrow{z_j})\} \tag{21}$$

The $\jmath$ index is composed of three factors: $\frac{1}{K}$, $\frac{E_1}{E_K}$, and D_k. The first factor decreases $\jmath$ as K increases. The second factor represents the ratio between E_1 and E_K, where E_1 is constant for a given dataset, and E_K is reduced with an increase in K. Therefore the $\jmath$ index value increases with an increase in E_K. This, in turn, contributes to the formation of more compact clusters.And the third factor, D_k, which represents the highest separation between any pair of clusters, tend to increase with the value of K. However, it should be noted that D_k cannot be greater than the maximum separation between two points in the dataset. Hence, these factors critically balance one another through contention. The contrast between the different cluster configurations is controlled by the power γ. A larger value of index $\jmath$ implies a better clustering.[50]

5.1.9 *Sym-index*

The Sym-index[55] is a cluster validity index based on a point symmetry that depends on the distance $d_{ps}(\overrightarrow{x}, \overrightarrow{z})$. Given a single point $\overrightarrow{z}$, the symmetrical (reflected) point of $\overrightarrow{x}$ with respect to a particular center $\overrightarrow{z}$ is $(2.\overrightarrow{z} - \overrightarrow{x})$ and denote this by $\overrightarrow{x^*}$. Let $knear$ unique nearest neighbors of $\overrightarrow{x^*}$ be at Euclidean distances of $d_i s = 1, \cdots, knear$, then:

$$\begin{aligned} d_{ps}(\overrightarrow{x}, \overrightarrow{z}) &= d_{sym}(\overrightarrow{x}, \overrightarrow{z}) \times d_e(\overrightarrow{x}, \overrightarrow{z}) \\ &= \frac{\sum_{i=1}^{knear} d_i}{knear} \times d_e(\overrightarrow{x}, \overrightarrow{z}) \end{aligned} \tag{22}$$

where $d_e(\overrightarrow{x}, \overrightarrow{z})$ is the Euclidian distance between $\overrightarrow{x}$ and $\overrightarrow{z}$, and $d_{sym}(\overrightarrow{x}, \overrightarrow{z})$ is a symmetry measure of $\overrightarrow{x}$ with respect to $\overrightarrow{z}$. The value of knear cannot be igual 1, since in this case if $\overrightarrow{x^*}$ exists in the dataset, then $d_{ps}(\overrightarrow{x}, \overrightarrow{z}) = 0$ and, hence, there will be no impact over the Euclidian distance. By contrast, large values of knear may not be suitable because they may underestimate the symmetry of a point with respect to a particular cluster center. In this work, the value of knear is chosen equal to 2. The proper value of $knear$ depends on the distribution of the dataset.

Sym-index is a cluster validity function which measures the average symmetry with respect to the cluster centers. Assume a partition of the dataset $X = \{\overrightarrow{x_j} : j = 1, \cdots, n\}$ into K clusters where the center of cluster $\overrightarrow{z_i}$ is computed by using $\overrightarrow{z_i} = \frac{\sum_{j=1}^{n_i} \overrightarrow{x_j^i}}{n_i}$ where $n_i (i = 1, \cdots, K)$ is the number of points in cluster i and $\overrightarrow{x_j^i}$ denotes the j-th point of the i-th cluster. The new cluster validity function Sym is defined as:

$$Sym(K) = (\frac{1}{K} \times \frac{1}{\varepsilon_k} \times D_k) \tag{23}$$

where

$$\varepsilon_k = \sum_{i=1}^{K} E_i \tag{24}$$

such that

$$E_i = \sum_{j=1}^{n_i} d_{ps}^*(\overrightarrow{x_j^i}, \overrightarrow{z_i}) \tag{25}$$

$$D_k = max_{i,j=1}^{K} \|\overrightarrow{z_i} - \overrightarrow{z_j}\| \tag{26}$$

where D_K is the maximum Euclidean distance between two cluster centers among all pairs of centers. $d_{ps}^*(\overrightarrow{x_j^i}, \overrightarrow{z_i})$ is defined by equation (6) with some constraint. Here, the first knear nearest neighbors of $\overrightarrow{x_j^*} = (2.\overrightarrow{z_i} - \overrightarrow{x_j^i})$ will be searched among only those points within cluster i, $i.e$, the knear nearest neighbors of $\overrightarrow{x_j^*}$, the reflected point of $\overrightarrow{x_j^i}$ with respect to $\overrightarrow{z_i}$, and $\overrightarrow{x_j^i}$ should belong to the i-th cluster. This index is maximized in order to obtain the suitable number of clusters.

5.1.10 *Con-index*

The Con-index is based in a relative neighborhood graph measures the connectivity among a set of objects using the relative neighborhood graph and quantifies the connectivity degree of well-separated clusters.[56] To measure the distance between a pair of points two steps are required:

- Build a relative neighborhood graph of the dataset.
- Compute $d_{short}(\overrightarrow{x_i}, \overrightarrow{x_j})$, the distance between any two objects, $\overrightarrow{x_i}$ and $\overrightarrow{x_j}$. This distance is measured along the relative neighborhood graph and then find all possible paths among these objects along the relative neighborhood graph. Suppose there is a total of p paths between $\overrightarrow{x}$ and $\overrightarrow{y}$, and the number of edges along the i-th path is $nedge_i$, for $i = 1, \cdots, p$. If the edges along the i-th path are denoted as $ed_1^i, \cdots, ed_{edge_i}^i$ and the corresponding edge weights are $w(ed_1^i), \cdots, w(ed_{edge_i}^i)$, then the shortest distance between x and y is defined as follows:

$$d_{short}(\overrightarrow{x}, \overrightarrow{y}) = min_{i=1}^{p} max_{j=1}^{nedge} w(ed_j^i) \tag{27}$$

The Con-index is defined as follows:

$$Con = \frac{\sum_{i=1}^{K} \sum_{j=1}^{n_k} d_{short}(\overrightarrow{z_i}, \overrightarrow{x_j^i})}{n \times min_{i,j=1 \wedge i \neq j}^{K} d_{short}(\overrightarrow{z_i}, \overrightarrow{z_j})} \tag{28}$$

where $d_{short}(\overrightarrow{z_i}, \overrightarrow{x_j^i})$ is the shortest distance along the graph between the objects $\overrightarrow{z_i}$ and $\overrightarrow{x_j^i}$, the j-th object of the i-th cluster. The Con-index has to be minimized to obtains its minimum when clusters are connected as well as separated.

5.1.11 *I-index*

An Euclidian distance-based cluster validity index, I-index,[44] is defined as follows:

$$I(K) = (\frac{1}{K} \times \frac{\varepsilon_1}{\varepsilon_k} \times D_k)^p \tag{29}$$

where K is the number of clusters. Here $\varepsilon_k = \sum_{k=1}^{K} \sum_{j=1}^{n_k} d_e(\overrightarrow{z_k}, \overrightarrow{x_j^k})$ and $D_k = max_{i,j=1}^{K} d_e(\overrightarrow{z_i}, \overrightarrow{z_j})$ where $\overrightarrow{z_j}$ denotes the center of j-th cluster and $\overrightarrow{x_j^k}$ denotes the j-th object of the k-th cluster. The I-index is composed of three factors, namely $\frac{1}{K}$, $\frac{\varepsilon_1}{\varepsilon_k}$ and D_K.

5.2 External indices

There are some indices based on cluster labels, as discussed in this section.

5.2.1 *Dunn index*

Dunn index[57] is formulated as follow:

$$DN = min_{1 \le i \le k} \left\{ min_{1 \le i \le k, j \ne i} \left[\frac{\delta(C_i, C_j)}{min_{1 \le k \le K}(\Delta C_k)} \right] \right\} \tag{30}$$

In the original form, the Dunn index used the following forms of δ and Δ:

$$\delta(C_i, C_j) = \min_{x_i \in C_i, x_j \in C_j} \{D(\overrightarrow{x_i}, \overrightarrow{x_j})\} \tag{31}$$

$$\Delta C_i = max_{x_i, x_k \in C_i} \{D(\overrightarrow{x_i}, \overrightarrow{x_j})\} \tag{32}$$

A larger value of the Dunn index corresponds to compact and well-separated clusters. Finaly, the main objective is, therefore, to maximize its value.[50]

5.2.2 *Cluster connectedness (Conn(C))*

The $Conn(C)$ represents the connectedness of the clusters. This index is defined as:[44]

$$Conn(C) = \sum_{i=1}^{n} \left(\sum_{j=1}^{L} \overrightarrow{k_{i,nn_{ij}}} \right) \tag{33}$$

where $\overrightarrow{k_{i,nn_{ij}}} = \frac{1}{j}$ if $\nexists C_k : i \in C_k \wedge nn_{ij} \in C_k$, and $\overrightarrow{k_{i,nn_{ij}}} = 0$ otherwise. Here nn_{ij} is the j-th nearest neighbor of data point $i.L$, a user-defined parameter, decides the number of neighbors contributing to the connectedness measure. In order to facilitate. The objective is to minimize cluster connectedness for obtaining highly connected clusters.[50]

5.2.3 *Edge*

The Edge index[58] measures the overall summed distances on boundaries between the clusters. This index is a value of the difference in the boundary between the clusters:

$$Edge(C) = -\sum_{i=1}^{n} \sum_{j \in \mathbb{F}_i} \xi_{i,j} \tag{34}$$

where $\xi_{i,j} = D(\overrightarrow{x_i}, \overrightarrow{x_j})$ if $\nexists \mid C_k : i \in C_k \wedge j \in C_k$, and $\xi_{i,j} = 0$, otherwise. Where, $\mathbb{F}_i$ is the set of N nearest neighbors of i-th point. In turn, N is a user-defined integer number. The $Edge(C)$ index must be minimized in order to obtain a well-separated clustering solution.

5.2.4 *Silhouette*

The silhouette s_i of a point can be defined as:

$$s_i = \frac{b_i - a_i}{max\{a_i, b_i\}} \tag{35}$$

where a_i is the mean distance of a point $\overrightarrow{x_i}$ from the other points of the cluster to which $\overrightarrow{x_i}$ belongs, and let b_i be the minimum of the mean distances of $\overrightarrow{x_i}$ from the points of the other clusters. The silhouette index S[59] is the mean silhouette of all data points:

$$S = \frac{1}{n} \sum_{i=1}^{n} S_i \tag{36}$$

The silhouette index score lies between -1 and 1; a higher value corresponds to better clustering.[50]

5.2.5 *Silhouette (Simplified)*

The Silhouette measure, $s(\overrightarrow{x_i})$, for an object $\overrightarrow{x_i}$, is calculated by:

$$s(x_i) = \frac{b(\overrightarrow{x_i}) - a(\overrightarrow{x_i})}{max\{a(\overrightarrow{x_i}), b(\overrightarrow{x_i})\}} \tag{37}$$

where $a(\overrightarrow{x_i})$ is the dissimilarity between $\overrightarrow{x_i}$ and its centroid, and $b(x_i)$ is the smallest dissimilarity between $\overrightarrow{x_i}$ and the centroid of the other clusters. The global Silhouette (S) evaluates the clustering quality according to the set of objects and is calculated by:

$$S(C) = \frac{1}{N} \sum_{i=1}^{N} s(\overrightarrow{x_i}) \tag{38}$$

where N is the total number of objects[37] and C is the set of all clusters. $S(C) \in [-1, 1]$, where 1 represents better clusters.

5.2.6 *Min-Max Cut*

The Min-Max cut index[60] that evaluate the clustering based on the minimization of the distances among the points within a same cluster and the maximization of the distances of the points from different clusters:

$$MinMaxCut(C) = \sum_{i=1}^{K} \frac{\sum_{j \in C_i} \sum_{k \notin C_i} D(\overrightarrow{x_j}, \overrightarrow{x_k})}{\sum_{j \in C_i} \sum_{k \in C_i} D(\overrightarrow{x_j}, \overrightarrow{x_k})} \quad (39)$$

Maximization of the numerator ensures that the points within a cluster are distant from the other points and minimization of the denominator value ensures that the points within a cluster are close with each other. Thus, taken as a whole, the index must be maximized to obtain good-quality clusters.

5.2.7 *Total within-cluster variance (TWCV)*

The TWCV[61] is defined as:

$$TWCV = \sum_{i=1}^{n} \sum_{j=1}^{d} \overrightarrow{x_{ij}}^2 - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{j=1}^{d} \left(\sum_{\overrightarrow{x_i} \in C_k} \overrightarrow{x_{ij}} \right)^2 \quad (40)$$

where $\overrightarrow{x_{ij}}$ denotes the j-th feature value of the i-th data point, and n_k denotes the number of points in cluster C_k. The aim is to minimize $TWCV$ to obtain compact clusters.

Algorithm 1 cOptBees-MO

1: **Input:** n_{min} (initial number of active bees); n_{max} (maximum number of active bees); ρ (inhibition radius); n_{mean} (average foraging effort); p_{min} (minimum probability of a bee being a recruiter); p_{rec} (percentage of non-recruiter bees that will be actually recruited); r_{max} (maximum number of clusters).
2: Randomly generate a swarm.
3: **while** (stopping criterion is not met) **do**
   3.1 Evaluate the quality of the sites being explored.
   3.2 Apply local search.
   3.3 Determine the recruiter bees.
   3.4 Update the number of bees.
   3.5 Determine the recruited and scout bees.
   3.6 Perform the recruitment process.
   3.7 Perform the exploration process.
4: End while
5: Evaluate the quality of the sites being explored by the bees.
6: Apply local search.
7: **Output:** Swarm of bees and their respective fitness values.

6. cOptBees-MO: A MULTIOBJECTIVE BEE-INSPIRED CLUSTERING ALGORITHM

Natural computing[25] is a research field that aims to understand natural fenomena in terms of information processing and, from this understanding, to investigate, model, abstract and apply this knowledge in different contexts.[62,63] Natural computing is composed of many subfields, one of them is Swarm Intelligence,[64] which aims at investigating and designing problem solving tech¬niques inspired by the collective behavior of social animals. The number of researches about the Swarm Intelligence methods, more specifically those based on the behavior of social bees, has increased significantly over the past years.[65–68] Inspired by the collective decision making process of social bees many methods have been proposed in the literature and the popularity of these methods has stimulated the development of several data mining approaches, such as clustering algorithms.[69] In this context, this paper proposes a multiobjective clustering algorithm, called cOptBees-MO, inspired by the foraging behavior of bee colonies.

The method summarized in Algorithm 1, is an extension of cOptBees,[4,70] a monoobjective clustering algorithm able to generate and maintain diversity of candidate solutions in a way to find multiple local optima without compromising its global search capability.[71] cOptBees was first presented by Cruz *et al.*[4] and the algorithm was applied to different clustering problems, being capable of finding optimal clusters, generating and maintaining the diversity of solutions, and finding the correct number of clusters. A parametric sensitivity analysis of cOptBees was presented by Cruz *et al.*[72] and it was applied to training a Radial Basis Function (RBF) neural network.[73] In the multiobjective version of cOptBees, introduced here, clustering is performed by simultaneously optimizing more than one objective function.

In cOptBees-MO, as in cOptBees, the artificial bees can play different roles: 1) Recruiters, responsible for attracting other bees to explore a promising region of the search space; 2) recruited, those who explore a promising region of the search space; or 3) scouts, responsible for randomly looking for new promising regions of the space.[71]

cOptBees-MO is summarized in Algorithm 1 and its main steps are detailed in the following sections.

6.1 Encoding scheme

The proposed algorithm works with a swarm of N artificial bees. For each bee, the objects in the database are associated with the nearest prototype. Initially, the swarm is randomly generated, respecting the maximum number of clusters, rmax (an input parameter of cOptBees-MO), and the number of bees is updated at each iteration, as will be shown in Section 6.3.

Each artificial bee encodes a potential solution for the clustering problem. A bee is composed of a set of prototypes and is defined by a matrix $B \in \mathbb{R}^{p \times r_{max}}$, where p is the dimension of the input vector, and rmax is the maximum number of clusters in a clustering. Thus, in a given column j, rows 1 to p represent the dimensions of prototype c_j and the last row is a threshold value, $L_j \in [0, 1]$, that defines if the

centroid $\overrightarrow{z_j}$ is active or not. The centroid $\overrightarrow{z_j}$ is active when its threshold is greater than or equal to 0.5. Figure 5 shows how an artificial bee is represented in a matrix format.[70,74]

$$B = \begin{bmatrix} z_{1,1} & \cdots & z_{1,rMax} \\ z_{2,1} & \cdots & z_{2,rMax} \\ \vdots & \cdots & \vdots \\ z_{p,1} & \ddots & z_{p,rMax} \\ L_1 & \cdots & L_{rMax} \end{bmatrix}$$

Figure 5. Matrix representation of a bee[70]

6.2 Determining the recruiter bees

The total number of recruiter bees in each iteration is determined in three steps. Firstly, a probability pr_i of being a recruiter bee is calculated for each active bee:

$$pr_i = \left(\frac{1 - pr_{min}}{Q_{max} - Q_{min}}\right) \cdot (q_i - Q_{min}) + pr_{min} \quad (41)$$

where q_i is the quality region being explored by bee i, Q_{min} and Q_{max} are, respectively, the minimum and maximum qualities among the region being explored by each active bee in the current iteration (the quality of a region is determined by an objective-function). In the second step the bees are classified as recruiters or non-recruiters according to the pr_i previously calculated. The third step consists of inhibiting those recruiters who explore the same regions of the search space. For this, the recruiter bees are processed from best to worst qualities of the region explored and, for each recruiter bee, the other recruiters who have a high similarity are inhibited. The recruiters that were inhibited are classified as non-recruiters.[75] In cOptBees-MO the similarity degree between two bees is determined based on the objects classified in the same cluster: The greater the number of objects classified in the same cluster, the greater the similarity degree. If the similarity between two bees is greater than or equal to the inhibition radius ρ, which is an input parameter, the bee that explores the worst site is inhibited. This process avoids that many recruiters explore the same promising regions of the search space.[70]

6.3 Updating the swarm size

This step aims to adjust the foraging effort according to the number of recruiters. At each iteration, $n_d = (n_r+1)\cdot n_{mean}$ determines the required number of bees in the swarm. The imput parameter ε is the average foraging effort, *i.e.*, it determines the desired number of non-recruiter bees for each recruiter, and n_r is the number of recruiters to be determined in Section 6.3. If n_d is greater than the current number of active bees in the swarm, $n_{adjust} = n_d - n_{active}$ is the number of bees that have to become active to achieve n_d active bees; if this number is less than the current number of active bees, $n_{adjust} = n_{active} - n_d$ bees have to become inactive to achieve n_d active bees. The swarm size is limited by the input parameters n_{max} and n_{min} that represent the maximum and minimum numbers of active bees, respectively. If $n_d > n_{max}$, then n_d is set to n_{max}; otherwise, if $n_d < n_{max}$, then n_d is set to n_{min}. Bees are selected taking account the corresponding qualities of the region they explore, from the worst to the best, for the inactivation process. The removal of a bee from the swarm depends on it being inactivated, whilst the insertion of a bee n the swarm depends on it being activated. Activated bees are placed in a random position in the search space, *i.e.*, the swarm size is updated dynamically in each iteration.[75]

6.4 Determining the scout and recruited bees

The number of artificial bees attracted by each recruiter is proportional to the quality of the explored food sources. Thus, a percentage of bees that was classified as non-recruiters are classified as recruited and associated with each recruiter proportionally. The number of non-recruiter bees is $n_{nr} = n_{active} - n_r$ and the number of recruited is $n_r = [p_{rec} \cdot n_{nr}]$, where p_{rec} is the percentage of non-recruiters that will be recruited and $[.]$ denotes the nearest integer function. The recruited bees exploit the promising regions already found by the recruiters. Each recruited bee is associated with the most similar recruiter. The bees that were not classified as recruited become scouts and are responsible for randomly exploring the search space to find new promising regions, reinforcing the generation and maintenance of diversity. The number of scout bees is $n_s = n_{nr} - n_r$.[75]

6.5 Recruiting

The recruiters explore promising regions of the search space and perform a recruitment process to attract the closest bees to the sites they explore. The recruitment, implemented by equation (42) or equation (43), is performed with 50% probability (for each equation), where α is an input parameter that represent the recruitment rate, $\overrightarrow{r}$ and $\overrightarrow{z_i}$ are, respectively, the recruiter and recruited bee, $\overrightarrow{u}$ is a random number with uniform distribution in the interval $[0, 1]$, $\overrightarrow{u}$ is a vector containing random numbers with uniform distribution in the interval $[0, 1]$ ($\overrightarrow{u}$ has the same dimension as $\overrightarrow{z_i}$ and $\overrightarrow{r}$) and $\otimes$ is the element-wise product:[75]

$$\overrightarrow{z_i} = \overrightarrow{z_i} + \overrightarrow{u} \cdot \alpha \cdot (\overrightarrow{r} - \overrightarrow{z_i}) \quad (42)$$

$$\overrightarrow{z_i} = \overrightarrow{z_i} + \alpha \cdot \overrightarrow{u} \otimes (\overrightarrow{r} - \overrightarrow{z_i}) \quad (43)$$

ISSN 1927-6974 E-ISSN 1927-6982

6.6 Exploration process

The exploration process allows that different regions of the search space to be explored by the scouts, and contributes to the discovery of new promissing regions. In this process the scout bees are positioned randomly in a new position in the search space.[75]

6.7 Calculating the fitness

In cOptBees-MO different combinations of objective functions can be used, and they may have a great impact on the algorithm's performance. In the experiments to be presented here two different scenarios were chosen:

- **Two objective functions:** The modified Silhouette[36,37] combined with the Dev(C);[11] and the modified Silhouette combined with the number of clusters.
- **Three objective functions:** The I-index, Con-index and Sym-index.[44]

In both scenarios, to improve the distinction between the quality of the solutions in the swarm, the concept of strength[3] was used in a simplified way.[76] Here, the strength measure was calculated to evaluate the solution quality considering the objective functions employed. The strength measure, St(i), is used as the fitness function in the search process of cOptBees-MO, instead of the Silhouette as in cOptBees:

$$St(i) = \frac{nDom_i}{tDom + 1} \tag{44}$$

where $nDom_i$ is the number of solutions dominated by or equal to solution i in relation to the objective values, $tDom$ is the total number of solutions dominated by all solutions, *i.e.*, it is the number of solutions dominated by each solution.

In the scenarios evaluated the objective functions used were: the modified Silhouette[36,37] combined with the Dev(C),[11] and the Modified Silhouette combined with the Number of Clusters; and in the second scenario, three different cluster validity indices are considered: Con-index, Sym-index and I-index.[44] These objective functions capture three different aspects of a clustering solution: the first quantifies the symmetry of a particular partitioning; the second and third measure, respectivelly, assess the connectedness and compactness of a solution taking into accont the Euclidean distance. These three indices were detailed in Section 5.

7. Experimental results

To assess the performance of cOptBees-MO, a Matlab R2014b[77] implementation was made and it was applied to different datasets from the UCI Machine Learning Repository. The results obtained by cOptBees-MO were compared to those obtained by other multiobjective algorithms available in the literature.

Experiments were performed in two scenarios, as discussed above, and the results were compared to those obtained by other multiobjective algorithms available in the literature by means of the Minkowski Score (MS).[78] In the second scenario three objective functions were optimized simultaneously, according to.[44] The clustering quality was assessed by the F-Measure[44] and the results obtained were compared with those of other multiobjective clustering algorithms present in the literature.

7.1 Datasets

The following datasets from the UCI were used to assess the performance of cOptBees-MO and its competitors:

- Congressional Votes: The dataset is composed of 435 instances, each one characterized by 16 Boolean attributes. It is the United States Congressional voting records in 1984 separeted in two classes: Republicans and Democrats;
- Soybean: Contains 47 instances on soybean diseases. Each object has 35 categorical attributes and is classified as one of the four diseases;
- Zoo: Composed of 101 instances of animals, each one represented by its name plus 16 attributes. The dataset consists of 7 different animal classes;
- Spiral: composed of 1,000 bidimensional instances distributed over 2 spiral clusters;
- Iris: Composed of 150 instances characterized by four attributes. The instances are separated in 3 clusters (Setosa; Versicolor; and Virginica), each one composed of 50 instances and two of them non-linearly separable;
- Cancer: composed of 683 data points and nine features. The data points are divided in two linearly separable classes: malignant and benign;
- Newthyroid: composed of 215 instances, each one characterized by five attributes. The data are distributed over 3 classes: euthyroidism, hypothyroidism and hyperthyroidism;
- Wine: Composed of 178 instances with 13 attributes resulting from a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars;
- LiverDisorder: Composed of 345 instances with 6 attributes each, separated in two classes;
- LungCancer: This dataset contains 32 instances having 56 attributes that describe 3 types of pathological lung cancers; and
- Glass: Composed of 214 instances having 9 attributes. The instances are divided into 6 categories.

7.2 Scenario I

In the first set of experiments the cOptBees-MO algorithm was applied to three datasets: Votes, Zoo and Soybean. The algorithm was run 20 times for each dataset, with the following input parameters (defined based on previous experiments): $n_{min} = 50$; $n_{max} = 100$; $\varepsilon = 10$; $r_{max} = 14$; $p_{min} = 0.1$; prec varying linearly between 0.1 and 0.5 with the number of iterations; ρ linearly varying between 0.1 and 0.5 with the number of iterations; number of iterations = 50.

Let matrices C and T be the clustering result and the true clustering, respectively. A clustering solution C is a $n \times n$ matrix for a set of n elements, where $C_{ij} = 1$ if point i and j are in the same cluster according to the solution, and $C_{ij} = 0$ otherwise. We compared the obtained results with those presented by Mukhopadhyay *et al.*[79] considering the Minkowski Score (MS).[78] The algorithms used for comparison were the SGA[80] and MOGA.[81] The MS of a clustering result C with reference to T, the matrix corresponding to the correct clustering, is defined as:

$$MS(TC) = \frac{\|T - C\|}{\|T\|} \tag{45}$$

where $\|T\| = \sqrt{\sum_i \sum_j T_{i,j}}$. Thus, the obtained MS is the normalized distance between the two matrices, T and C. Lower values of MS imply better clustering solutions, and a perfect solution has a score of zero.

Table 3. Average Minkowski score for the SGA, MOGA, cOptBees-MO$_1$ and cOptBees-MO$_2$ over the Votes, Zoo and Soybean datasets

Dataset	SGA	MOGA	cOptBees-MO$_1$	cOptBees-MO$_2$
Votes	0.6550	0.5344	0.6290	0.164
Zoo	0.5096	0.3807	0.6679	0.000
Soybean	0.4147	0.0000	0.6604	0.000

Table 3 presents the average Minkowski Score for the SGA, MOGA and cOptBees-MO for the Votes, Zoo and Soybean datasets. Two versions of cOptBees-MO were used: cOptBees-MO$_1$ was configured to optimize simultaneously the Silhouette and Cluster Deviation, and cOptBees-MO$_2$ was configured to optimize simultaneously Silhouette and the Number of Clusters.

As can be observed in the results presented in Table 3, cOptBees-MO does not perform well when combining the Silhouette with the Cluster Deviation, but when it is trained to optimize the Silhouette and the Number of Clusters its performance is significantly increased, beating all the other algorithms for all problems. This result may be a consequence of the fact that the mathematical definition of the Minkowski Score tends to provide a much lower quality when the solution presents a number of clusters different from the desired one.

7.3 Scenario II

In the second set of experiments the cOptBees-MO algorithm was applied over eight datasets available in the Machine Learning Repository from University of California, Irvine (UCI): Spiral; Iris; Cancer; Newthyroid; Wine; LiverDisorder; LungCancer; and Glass. The algorithm was run 30 times for each dataset, with the following input parameters (defined based on previous experiments): $n_{min} = 100$; $n_{max} = 100$; $\varepsilon = 10$; $r_{max} = 8$; $p_{min} = 0.1$; $p_{rec} = 0.7$; with the number of iterations; $\rho = 0.3$; number of iterations = 50. We evaluated the quality of the clustering based on the F-Measure[44] and the results were compared with those presented by four multiobjective clustering algorithms:[44] GenClustMOO, MOCK, VGAPS and GenClustPESA2.

Table 4. Average F-measure values and standard deviations for each dataset by GenClustMOO, MOCK, VGAPS, GenClustPESA2 and cOptBees-MO clustering algorithms

Dataset	GenClustMOO	MOCK	VGAPS	GenClustPESA2	cOptBees-MO
Iris	0.788 ± 0.011	0.775 ± 0.022	0.457 ± 0.013	**0.926** ± 0.015	0.830 ± 0.024
Cancer	0.969 ± 0.009	0.819 ± 0.014	0.953 ± 0.012	**0.979** ± 0.014	0.915 ± 0.021
Newthy	**0.863** ± 0.016	0.739 ± 0.014	0.659 ± 0.011	0.687 ± 0.015	0.838 ± 0.021
Wine	0.709 ± 0.012	**0.726** ± 0.002	0.617 ± 0.008	0.437 ± 0.012	0.640 ± 0.007
LiverDis	0.673 ± 0.002	0.671 ± 0.012	**0.705** ± 0.009	0.603 ± 0.015	0.612 ± 0.050
LungCan	0.802 ± 0.014	0.443 ± 0.011	0.741 ± 0.008	**0.843** ± 0.002	0.731 ± 0.079
Glass	0.494 ± 0.012	0.534 ± 0.006	0.534 ± 0.008	0.534 ± 0.012	**0.797** ± 0.040

Table 4 presents the average F-Measure values for the different datasets by GenClustMOO, MOCK, VGAPS, GenClustPESA2 and cOptBees-MO clustering algorithms. The best methods for each data set are marked in bold. The F-Measure (FM) and the number of clusters (OC) of the best solution obtained by GenClustMOO, MOCK, VGAPS, GenClustPESA2 and cOptBees-MO clustering algorithms for each dataset are presented in Table 5. Here d and k denote the dimension and

the correct number of clusters, respectively.

The results exhibited in Table 5 show that the cOptBees-MO had a competitive performance when compared with other multiobjective clustering approaches from the literature. When compared with GenClustMOO, MOCK, VGAPS, GenClustPESA2 algorithms, the cOptBees-MO presented better performance for the Glass data, with an average F-measure of 0.797 $\pm$ 0.040, followed by MOCK, VGAPS, GenClustPESA2 with average F-Measure equals to 0.534. The GenClustMOO algorithm obtained the best performances for the Iris, Cancer, LungCancer and Glass datasets.

Table 5. The F-Measure (FM) and number of clusters (OC) of the best solution obtained by GenClustMOO, MOCK, VGAPS, GenClustPESA2 and cOptBees-MO clustering algorithms for each dataset

Dataset	Instances	*d*	*k*	GenClustMOO		MOCK		VGAPS		GenClustPESA2		cOptBees-MO	
				OC	*FM*	*OC*	*FM*	*OC*	*FM*	*OC*	*FM*	*OC*	*FM*
Iris	150	4	3	3	0.79	2	0.78	3	0.76	3	**0.93**	3	0.86
Cancer	683	9	2	2	**0.97**	2	0.82	2	0.95	2	**0.97**	3	0.94
Newthy	215	5	3	3	**0.86**	2	0.74	5	0.66	9	0.69	4	**0.86**
Wine	178	13	3	3	0.71	3	**0.73**	6	0.62	13	0.44	2	0.65
LiverDis	345	6	2	2	0.67	2	0.67	2	**0.70**	5	0.60	2	0.67
LungCan	33	56	2	2	0.80	7	0.44	3	0.74	4	**0.84**	2	0.83
Glass	214	9	6	6	0.49	5	0.53	5	0.53	5	0.53	2	**0.88**

Considering the best solution of the thirty execution, the cOptBees-MO obtained the best F-Measure values for the NewThyroid and Glass datasets. For the NewThyroid dataset the best solution presented F-Measure value equals to 0.86 and 4 clusters. For the Glass dataset the F-Measure was equal to 0.88 and 2 clusters, followed by MOCK, VGAPS and GenClustPESA2 algorithms that presented an F-Measure of 0.53.

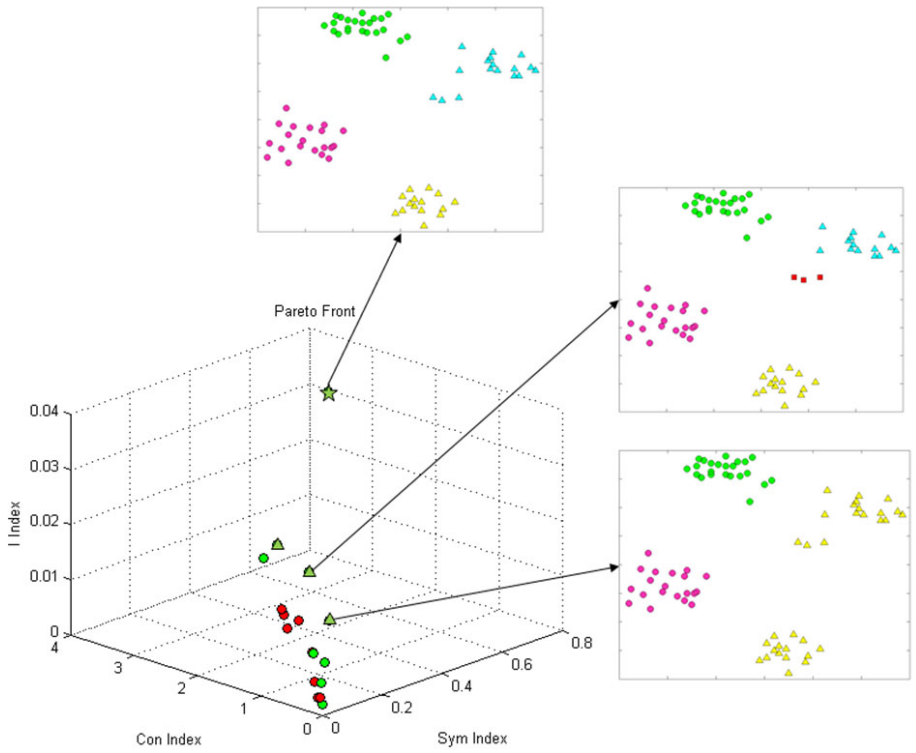

Figure 6. Solutions presented by the cOptBees-MO for the Ruspini

To illustrate the diversity of solutions obtained by cOptBees-MO, Figure 6 presents the solutions obtained by the algorithm for the well-known Ruspini dataset. In the function space presented, the star (★) represents a solution encoded by a recruiter bee on the non-dominated set; the triangles (▲) are the recruiter bees that are outside the Pareto front; and the circles (•) are the non-recruiter bees. The clustering represented by three of these bees are presented on the right hand side of the picture. This example shows that the cOptBees-MO is able to maintain the diversity of solutions, both at the Pareto front and outside it.

8. Discussion and future trends

This paper presented a brief survey of bio-inspired multiobjective clustering algorithms, emphasizing a description of the task and the evaluation metrics to be used, and then proposed a multiobjective clustering algorithm, named cOptBees-MO, inspired by the foraging behavior of bee colonies. To evaluate the performance of cOptBees-MO two different scenarios were considered: Scenario I with three different datasets and comparisons with SGA and MOGA; and Scenario II with seven datasets and comparisons with GenClustMOO, MOCK, VGAPS and GenClustPESA2. Different types and combinations of objective functions were also considered in each scenario.

The results showed that cOptBees-MO is competitive when compared to other multiobjective clustering algorithms from the literature. To achieve a better performance with the Minkowski score (Scenario I) and the F-Measure (Scenario II) cOptBees-MO was run with the Silhouette and the Number of Clusters; and Sym-index, I-index and Con-index as objective functions, respectively. The results showed a better performance of Silhouette and Number of Clusters as objective functions.

The algorithm proposed obtained a good performance, being able to find high quality cluster partitions in datasets without the need to inform the correct number of clusters in the dataset. By contrast, SGA and MOGA rely on this kind of information to be used, as well as GenClustMOO, MOCK, VGAPS and GenClustPESA2 and the last one uses k-means as a heuristic initialization process. The cOptBees-MO has shown to be able to generate and maintain the diversity of solutions by finding multiple suboptimal solutions in a single run, a feature useful for solving MOO problems, such as optimal multiobjective data clustering.

In a summarized way, the main advantages of using cOptBees-MO are the fact that it does not require the value k, the number of clusters, and neither the use of external memory to store the best solutions found during the search process, like other algorithms from the literature. It automatically finds the number of clusters by applying its artificial bees with different bee-inspired behaviors to explore and map the search space.

As future work, keeping in mind that agents with different bee-inspired behaviors are able to find feasible and competitive clustering solutions, our efforts will be concentrated on local operators to improve the Minkowski Score and F-Measure indices over distinct dataset types. Furthermore, by considering the results presented by cOptBees-MO, it is important to further explore other objective functions over real and artificial clustering datasets. The objective functions chosen are directly connected to the quality of the obtained clusterings. Therefore, they are constant targets of study in clustering analysis. It is also important to emphasize the influence of objective functions in the MOO algorithms to find high quality clusterings.

Acknowledgements

The authors thank Capes, CNPq, Fapesp, and MackPesquisa for the financial support.

References

[1] Jain AK. Data clustering: 50 years beyond K-means. Pattern Recognition Letters. 2010; 31(8): 651-66. https://doi.org/10.1016/j.patrec.2009.09.011

[2] Halkidi M, Batistakis Y, Vazirgiannis M. Clustering algorithms and validity measures. Thirteenth International Conference on Scientific and Statistical Database Management. 2001: 3-22. https://doi.org/10.1109/ssdm.2001.938534

[3] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. Tikreport. Athens; 2011. PMid:20649424.

[4] Cruz DPF, Maia RD, de Castro LN, et al. A Bee-Inspired Algorithm for Optimal Data Clustering. IEEE World Congress on Computational Evolutionary. 2013: 3140-7. https://doi.org/10.1109/cec.2013.6557953

[5] Deb K. Multi-Objective Optimization using Evolutionary Algorithms, England: John Wiley and Sons; 2001.

[6] Coello CC, Lamont G. Evolutionary Algorithms for Solving Multi-Objective Problems. Berlin, Germany: Springer; 2007.

[7] Bandyopadhyay S. Unsupervised Classification - Similarity Measures, Classical and Metaheuristic Approaches, and Applications. Berlin: Springer Science & Business Media; 2013.

[8] Edgeworth FY. Mathematical Physics. London, England; 1881.

[9] Pareto V. Cours D'Economie Politique. F. Rouge, Lausanne, Switzerland; 1896.

[10] Maulik U, Bandyopadhyay S, Mukhopadhyay A. Multiobjective Genetic Algorithms for Clustering: Applications in Data Mining and Bioinformatics. Springer Science & Business Media; 2011. https://doi.org/10.1007/978-3-642-16615-0

[11] Mukhopadhyay A, Maulik U, Bandyopadhyay S. A Survey of Multiobjective Evolutionary Clustering. ACM Computing Surveys (CSUR). 2015; 47(4): 61. https://doi.org/10.1145/2742642

[12] Gruna R. The Wolfram Demonstration Project - Evolutionary Multiobjective Optimization. 2015. [Online]. Available: http://demonstrations.wolfram.com/EvolutionaryMultiobjectiveOptimization.

[13] Zitzler E, Laumanns M, Bleuler S. A tutorial on evolutionary multiobjective optimization. In Metaheuristics for multiobjective optimisation. 2004: 3-37. https://doi.org/10.1007/978-3-642-17144-4_1

[14] Nocedal J, Wright S. Numerical optimization. Springer Science & Business Media; 2006.

[15] Xue B, Zhang M, Browne WN. Particle swarm optimization for feature selection in classification: a multi-objective approach. IEEE transactions on cybernetics. 2013; 43(6): 1656-71. PMid:24273143. https://doi.org/10.1109/TSMCB.2012.2227469

[16] Ghosh A, Nath B. Multi-objective rule mining using genetic algorithms. Information Sciences. 2004; 163(1): 123-33. https://doi.org/10.1016/j.ins.2003.03.021

[17] Moya MM, Hush DR. Network constraints and multi-objective optimization for one-class classification. Neural Networks. 1996; 9(3): 463-74. https://doi.org/10.1016/0893-6080(95)00120-4

[18] Bandyopadhyay S, Maulik U, Mukhopadhyay A. Multiobjective genetic clustering for pixel classification in remote sensing imagery. IEEE transactions on Geoscience and Remote Sensing. 2007; 45(5): 1506-11. https://doi.org/10.1109/TGRS.2007.892604

[19] Das S, Abraham A, Konar A. Metaheuristic clustering. Springer; 2009.

[20] Handl J, Knowles J. Multi-Objective Clustering and Cluster Validation. Multi-Objective Machine Learning Studies in Computational Intelligence. 2006; 16: 21-47. https://doi.org/10.1007/3-540-33019-4_2

[21] Rendón E, Abundez I, Arizmendi A, et al. Internal versus external cluster validation indexes. International Journal of computers and communications. 2011: 27-34.

[22] Rendón E, Abundez IM, Gutierrez C, et al. A comparison of internal and external cluster validation indexes. 2011 American Conference. 2011; 29: 158-63.

[23] Freitas AA. A critical review of multi-objective optimization in data mining: a position paper. SIGKDD Explor. 2004; 6(2): 77-86. https://doi.org/10.1145/1046456.1046467

[24] Davis L. Handbook of genetic algorithms. 1991.

[25] de Castro LN. Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications. Chapman & Hall/CRC; 2006.

[26] Karaboga D, Akay B. A comparative study of Artificial Bee Colony algorithm. Applied Mathematics and Computation. 2009; 214(1): 108-32. https://doi.org/10.1016/j.amc.2009.03.090

[27] Dorigo M, Birattari M, Stutzle T. Ant colony optimization. IEEE Computational Intelligence Magazine. 2006; 1(4): 28-39. https://doi.org/10.1109/MCI.2006.329691

[28] Suresh K, Kundu D, Ghosh S, et al. Multi-Objective Differential Evolution for Automatic Clustering with Application to Micro-Array Data Analysis. Sensors. 2009; 9(5): 3981-4004. PMid:22412346. https://doi.org/10.3390/s90503981

[29] Suresh K, Kundu D, Ghosh S, et al. Data Clustering Using Multiobjective Differential Evolution Algorithms. Fundamenta Informaticae XXI. 2009; 97(4): 1001-24.

[30] Tout S, Sverdlik W, Sun J. Parallel hybrid clustering using genetic programming and multi-objective fitness with density (PYRAMID). Proceedings of the 2006 International Conference on Data Mining (DMIN'06). 2006: 197-203.

[31] Zheng Y, Jia L, Cao H. Multi-Objective Gene Expression Programming for Clustering. Information Technology And Control. 2012; 41(3): 283-94. https://doi.org/10.5755/j01.itc.41.3.1330

[32] Santos DS, Oliveira D, Bazzan ALC. A multiagent, multiobjective clustering algorithm. In Data Mining and Multi-agent Integration. Springer US. 2009: 239-49. https://doi.org/10.1007/978-1-4419-0522-2_16

[33] Wu CW, Chiang TC, Fu LC. An ant colony optimization algorithm for multi-objective clustering in mobile ad hoc networks. IEEE Congress on Evolutionary Computation (CEC). 2014: 2963-8. https://doi.org/10.1109/cec.2014.6900458

[34] Gong M, Xidian X, Zhang L, et al. Solving multiobjective clustering using an immune-inspired algorithm. IEEE Congress on Evolutionary Computation (CEC). 2007: 15-22.

[35] Nanda SJ, Panda G. Automatic clustering algorithm based on multiobjective Immunized PSO to classify actions of 3D human models. Engineering Applications of Artificial Intelligence. 2013; 26(5-6): 1429-41. https://doi.org/10.1016/j.engappai.2012.11.008

[36] Chen S, Amid D, Schreck T. Self-Organizing Maps for Multi-Objective Pareto Frontiers. Visualization Symposium (PacificVis), 2013 IEEE. 2013: 153-60. https://doi.org/10.1109/pacificvis.2013.6596140

[37] Hironaka A, Okamoto T, Hirata H. Analysis and Improvements of the Pareto Optimal Solution Visualization. SICE Journal of Control, Measurement, and System Integration. 2015; 8(1): 34-43. https://doi.org/10.9746/jcmsi.8.34

[38] Xu R, Wunsch D. Survey of Clustering Algorithms. IEEE TRANSACTIONS ON NEURAL NETWORKS. 2005; 16(3): 645-78. PMid:15940994. https://doi.org/10.1109/TNN.2005.845141

[39] Yoshimi M, Kuhara T, Hiroyasu T. Visualization of Pareto Solutions by Spherical Self-Organizing Map and It's Acceleration on a GPU. Journal of Software Engineering and Applications. 2012 March: 129-37.

[40] Özyer T, Alhajj R, Barker K. Clustering by Integrating Multi-objective Optimization with Weighted K-Means and Validity Analysis. Intelligent Data Engineering and Automated Learning – IDEAL. 2006; 4224: 454-63. https://doi.org/10.1007/11875581_55

[41] Matake N, Hiroyasu T, Miki M, et al. Multiobjective Clustering with Automatic k-determination for Large-scale Data. Proceedings of the 9th annual conference on Genetic and evolutionary computation. 2007: 861-8. https://doi.org/10.1145/1276958.1277126

[42] Handl J, Knowles J. Exploiting the Trade-off - The Benefits of Multiple Objectives in Data Clustering. In Evolutionary Multi-Criterion Optimization, Springer Berlin Heidelberg. 2005: 547-60.

[43] Bandyopadhyay S, Saha S. A point symmetry based clustering technique for automatic evolution of clusters. IEEE Trans Knowl Data Eng. 2008; 20(11): 1-17. https://doi.org/10.1109/TKDE.2008.79

[44] Saha S, Bandyopadhyay S. A generalized automatic clustering algorithm in a multiobjective framework. Applied Soft Computing. 2013 January; 13(1): 89-108. https://doi.org/10.1016/j.asoc.2012.08.005

[45] Yang D, Jiao L, Liu F. Artificial immune multi-objective SAR image segmentation with fused complementary features. Information Sciences. 2011; 181(13): 2797-812. https://doi.org/10.1016/j.ins.2011.02.025

[46] Mishra BSP, Dehuri S, Cho SB. Swarm Intelligence in Multiple and Many Objectives Optimization: A Survey and Topical Study on EEG

Signal Analysis. Multi-objective Swarm Intelligence. 2015: 27-73. https://doi.org/10.1007/978-3-662-46309-3_2

[47] Armano G, Farmani MR. Multiobjective clustering analysis using particle swarm optimization. Expert Systems with Applications: An International Journal. 2016 August ; 55: 184-93. https://doi.org/10.1016/j.eswa.2016.02.009

[48] Kennedy JF, Eberhart RC, Shi Y. Swarm intelligence. Morgan Kaufmann; 2001.

[49] Kasprzak EM, Lewis KE. Pareto analysis in multiobjective optimization using the collinearity theorem and scaling method. Structural and Multidisciplinary Optimization. 2001; 22(3): 208-18. https://doi.org/10.1007/s001580100138

[50] Maulik U, Bandyopadhyay S, Mukhopadhyay A. Multiobjective Genetic Algorithms for Clustering: Applications in Data Mining and Bioinformatics. Springer Science & Business Media; 2011. https://doi.org/10.1007/978-3-642-16615-0

[51] Handl J, Knowles J. An evolutionary approach to multiobjective clustering. IEEE transactions on Evolutionary Computation. 2007; 11: 56-76. https://doi.org/10.1109/TEVC.2006.877146

[52] Bezdek JC. Pattern Recognition with Fuzzy Objective Function Algorithms. Norwell, MA: Kluwer Academic Publishers; 1981. https://doi.org/10.1007/978-1-4757-0450-1

[53] Ripon KSN, Tsang CH, Kwong S. Multi-objective data clustering using variable-length real jumping genes genetic algorithm and local search method. The 2006 IEEE International Joint Conference on Neural Network Proceedings. 2006: 3609-16. https://doi.org/10.1109/ijcnn.2006.247372

[54] Kirkland O, Rayward-Smith VJ, de la Iglesia B. A novel multiobjective genetic algorithm for clustering. International Conference on Intelligent Data Engineering and Automated Learning. 2011: 317-326. https://doi.org/10.1007/978-3-642-23878-9_38

[55] Bandyopadhyay S, Saha S. GAPS: A clustering method using a new point symmetry-based distance measure. Pattern Recognition. 2007: 3430-51. https://doi.org/10.1016/j.patcog.2007.03.026

[56] Saha S, Bandyopadhyay S. Some connectivity based cluster validity indices. Applied Soft Computing. 2012: 1555-65. https://doi.org/10.1016/j.asoc.2011.12.013

[57] Dunn JC. Well-separated clusters and optimal fuzzy partitions. Journal of cybernetics. 1974; 4(1): 95-104. https://doi.org/10.1080/01969727408546059

[58] Shirakawa S, Nagao T. Evolutionary image segmentation based on multiobjective clustering. 2009 IEEE Congress on Evolutionary Computation. 2009: 2466-73. https://doi.org/10.1109/CEC.2009.4983250

[59] Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics. 1987; 20: 53-65. https://doi.org/10.1016/0377-0427(87)90125-7

[60] Ding CH, He X, Zha H, et al. A min-max cut algorithm for graph partitioning and data clustering. Proceedings IEEE International Conference Data Mining on ICDM 2001. 2001: 107-14. https://doi.org/10.1109/ICDM.2001.989507

[61] Liu Y, Özyer T, Alhajj R, et al. Integrating multi-objective genetic algorithm and validity analysis for locating and ranking alternative clustering. Informatica. 2005; 29: 33-40.

[62] Kari L, Rozenberg G. The Many Facets of Natural Computing. Commun. ACM. 2008; 51(10): 72-83. https://doi.org/10.1145/1400181.1400200

[63] de Castro LN, Xavier RS, Pasti R, et al. The Grand Challenges in Natural Computing Research: The Quest for a New Science. IJNCR. 2011; 2: 17-30. https://doi.org/10.4018/jncr.2011100102

[64] Kennedy J, Eberhart R, Shi Y. Swarm Intelligence. Morgan Kauffman Publishers; 2001.

[65] Karaboga D, Akay B. A survey: algorithms simulating bee swarm intelligence. Artificial Intelligence Review. 2009; 31(1): 61-85. https://doi.org/10.1007/s10462-009-9127-4

[66] Karaboga D, Ozturk C. Expert Systems with Applications. Applied Soft Computing. 2011; 11(1): 652-7. https://doi.org/10.1016/j.asoc.2009.12.025

[67] Zhang C, Ouyang D, Ning J. An artificial bee colony approach for clustering. Expert Systems with Applications. 2010; 37(7): 4761-7. https://doi.org/10.1016/j.eswa.2009.11.003

[68] Yan X, Zhu Y, Zou W, et al. A new approach for data clustering using hybrid artificial bee colony algorithm. Neurocomputing. 2012; 95: 241-50. https://doi.org/10.1016/j.neucom.2012.04.025

[69] Martens D, Baesens B, Fawcett T. Editorial Survey: Swarm Intelligence for Data Mining. Machine Learning. 2011; 82(1): 1-42. https://doi.org/10.1007/s10994-010-5216-5

[70] Cruz DPF, Maia RD, de Castro LN. A New Encoding Scheme for a Bee-Inspired Optimal Data Clustering Algorithm. 1st BRICS Countries Congress (BRICS-CCI). 2013: 136-41. https://doi.org/10.1109/brics-cci-cbic.2013.32

[71] Maia RD, de Castro LN, Caminhas WM. Collective Decision-Making by Bee Colonies as Model for Optimization-the OptBees Algorithm. Applied Mathematical Sciences. 2013; 7(87): 4327-51. https://doi.org/10.12988/ams.2013.35271

[72] Cruz DPF, Maia RD, De Castro LN. On the Sensitivity of a Bee-Inspired Algorithm to Its Internal Parameters. International Journal of Computer Information Systems and Industrial Management Applications. 2015; 7: 84-93.

[73] Cruz DPF, Maia RD, Silva LA, et al. BeeRBF: A Bee-Inspired Data Clustering Approach to Design RBF Neural Network Classifiers. Neurocomputing; 2015.

[74] Maia RD, Barbosa WO, de Castro LN. Colônias de Abelhas como Modelo para Agrupamento Ótimo de Dados: Uma proposta e análise paramétrica qualitativa. XIX Congresso Brasileiro de Automática. 2012: 623-30.

[75] Maia RD, de Castro LN, Caminhas WM. Collective decision-making by bee colonies as model for optimization - the OptBees algorithm. Applied Mathematical Sciences. 2013; 7(87): 4327-41. https://doi.org/10.12988/ams.2013.35271

[76] Vieira IN. Otimização Multi-Objetivo de Rotas de Dutos Submarinos através de Sistemas Imunológicos Artificiais. Rio de Janeiro: Universidade Federal do Rio de Janeiro; 2011.

[77] MathWorks. User's Guide (R2014b). 2014. Internet. Available from: https://www.mathworks.com/. [Accessed 2016].

[78] Jardine N, Sibson R. Mathematical Taxonomy. New York: Wiley; 1971.

[79] Mukhopadhyay A, Maulik U. Multiobjective approach to categorical data clustering. Evolutionary Computation. 2007: 1296-303. https://doi.org/10.1109/cec.2007.4424620

[80] Dhiraj K, Rath SK. Comparison of SGA and RGA based Clustering Algorithm for Pattern Recognition. International Journal of Recent Trends in Engineering. 2009 May; 1: 269-73.

[81] Dutta D, Dutta P, Sil J. Simultaneous feature selection and clustering for categorical features using multi objective genetic algorithm. Hybrid Intelligent Systems (HIS), 2012 12th International Conference on. IEEE. 2012: 191-6. https://doi.org/10.1109/his.2012.6421332