Journal of the Brazilian Computer Society, 2024, 30:1, doi: 10.5753/jbcs.2024.3483

An Information Theoretic Learning Artificial Immune Network for Alternative Clustering

Ederson Borges [**Federal Institute of São Paulo (IFSP)** | ***edersonborges@ifsp.edu.br***]
Guilherme Palermo Coelho [**Universidade Estadual de Campinas (UNICAMP) - Faculdade de Tecnologia (FT)** | ***gpcoelho@unicamp.br***]

Federal Institute of São Paulo (IFSP), Avenida Marginal, 585 - Bairro Fazenda Nossa Senhora Aparecida do Jaguari, São João da Boa Vista, SP, 13871-298, Brazil.

Received:23 June 2024 • **Accepted:** 09 August 2024 • **Published:** 16 October 2024

Abstract Clustering is an unsupervised task employed when there is no prior knowledge about the structure and information contained in the data. Nowadays the amount of information and the dimensionality of data increased. Due to this, several datasets contain samples that can be clustered in different ways, presenting different partitions. Classical algorithms tend to obtain a single partition per execution and also require information like the number of clusters. Immuno-inspired algorithms were developed to reduce some of these drawbacks. They can find alternative solutions without knowing the number of clusters, but high dimensionality reduces their performance leading to low convergence rates. Information Theoretic Learning (ITL) uses statistical information of the data regardless of prior knowledge of the structure of these data and the dimensionality involved. Applied in several papers for clustering, ITL-based algorithms tend to present good performance for this task. This paper presents an immuno-inspired ITL-based algorithm (ITL-aiNet) capable of finding and maintaining high-quality and diverse solutions for datasets regardless of their dimensionality and structure. Real-world image and document datasets of varying dimensions were used in the experiments, allowing different ways of clustering. The results were evaluated using external indices. The proposed approach was capable of maintaining high-quality and diverse solutions, compared to other strategies found in the literature. The indices used to measure the quality and diversity of solutions indicated that the algorithm is capable of finding and maintaining good solutions. Solutions that have greater diversity than other algorithms in some datasets and higher quality in others.

Keywords: Information Theory, Artificial Immune Systems, Clustering, Diversity Maintenance

1 Introduction

With the increase in the use of the Internet of Things and the spread of Big Data, different data sources became available, and several high-dimension datasets emerged. So, new possible interpretations of these data are expected [Wang *et al.*, 2019; Chikhi, 2016].

Consider that a person can be photographed in various positions with different cameras, or that the same news can be published in multiple media with different languages. Although they may appear differently, these datasets can still represent the same subjects depicted from multiple perspectives [Fu *et al.*, 2020].

Another example are web pages, which can be characterized by both the textual content they contain and the interconnected pages accessible through hyperlinks [Chao *et al.*, 2021]. Although distinct, these web pages might share thematic connections representing identical content. Moreover, they may address diverse subjects and establish links with other pages.

In the recent literature, this type of information that refers to instances where the same entity (data) can be analyzed from different viewpoints are known as *multi-view data*. Novel automated methodologies are being explored to unveil this array of interpretations within datasets, aiming to enhance comprehension and reveal potential associations among the constituent elements of a given database [Fu *et al.*, 2020].

Clustering appears as an option for finding multiple alternative information about the data. The first challenge of clustering is finding structures and patterns among data. So, methods are required to quantify this data and find relationships to form cohesive clusters between samples with similar characteristics and dissimilar from other clusters with different samples [Cunha *et al.*, 2017].

Distance measures, such as the Euclidean distance, are commonly used to calculate the similarity between objects in datasets. But, it is well-known that these measures do not work well on high dimensions [Araújo *et al.*, 2013].

With Information Theory (IT) concepts, it is possible to obtain information directly from the data, based on their statistical properties [Araújo *et al.*, 2013]. Shannon conceptualized IT to analyze messages transmitted over noisy channels, and since then, it has been used in several fields that require data analysis.

So IT was created to help study the theoretical issues of how to optimally encode messages according to their statistical structures. However, it goes beyond this area, contributing to digital communications by: (i) assisting in determining the transmission capacity of a channel; (ii) data compression with lossless compression algorithms like Huffman and lossy compression based on entropy coding; and (iii) in cryptography to measure the strength of algorithms and keys, among other applications [Cover and Thomas, 2005].

IT has also contributed to data mining tasks leading to a field of research named *Information Theoretic Learning* (ITL – [Principe, 2010]). Employing ITL in clustering is useful to find partitions, independently of the dataset dimension. But the task of finding partitions with diversity is complex since the algorithm must maintain not only one possible solution but multiple solutions [Kontonasios and De Bie, 2015]. The number of clusters in a partition is another challenge for many algorithms.

Several clustering algorithms in the literature consider the possibility of alternative solutions with different strategies [Aggarwal and Reddy, 2014]. Some examples, like MinCEntropy [Vinh and Epps, 2010], obtain an alternative partition for a given dataset and are based on Information Theory concepts. However, it requires a pre-definition of the number of clusters.

Other strategies, such as the *Multiple Independent Subspace Clusterings* (MISC) [Wang *et al.*, 2019], aim to find different subspaces among the attributes of the objects and, for the subspaces found, present a feasible solution. This method is limited by high computational cost when the number of attributes is high due to the calculation of matrix-based subspaces for each alternative clustering.

The bee-inspired multiobjective optimization (cOptBees-MO) algorithm uses swarm intelligence to find better partitions through mechanisms inspired by food resource exploration of bee colonies. The "bees" in the algorithm search for regions with better possibilities to generate clusters and the objects are associated with the closest bee. It is a dynamic algorithm with a limitation on the maximum number of possible clusters [Cunha *et al.*, 2017].

[Orouskhani *et al.*, 2020] shows a complex network-based algorithm proposed to attempt to determine central nodes to find better clustering configurations and, through node recombinations, it tries to generate offspring solutions from the initial one.The algorithm has high complexity because it is based on graph and matrix calculations to find the best connections between nodes. Another algorithm based on matrix computations is [Wan *et al.*, 2023]. In this algorithm, different matrices are combined to form a consensus matrix and find a new clustering for this matrix.

In this way, being aware of the difficulties currently encountered in the literature, such as finding the number of clusters automatically and working with spaces of different shapes and dimensions, this paper proposes a new clustering algorithm based on a model of artificial immune networks [de Castro and Von Zuben, 2002].

The algorithm, named *ITL-aiNet*, searches for and maintains multiple feasible, diverse, and high-quality solutions. Therefore, ITL-aiNet uses ITL to overcome the drawbacks of evaluating new structures directly from the data [Zhang *et al.*, 2013]. Besides, ITL-aiNet methods search for an ideal number of clusters for each alternative partition found.

In the context of Artificial Immune Networks for clustering, ITL can be used to assess the quality of the population. Mutation operators can also benefit from ITL concepts so that the number of clusters can be modified dynamically, and the best possible configuration for each candidate solution in the population can be identified.

Thus, this paper addresses the drawbacks found in previous algorithms through the use of bioinspired strategies and ITL concepts. We assess partitions using concepts from Information Theory and bring new methods (operators based on ITL concepts) to cluster objects from datasets. The contributions of this work can be summarized as:

- First, the development of a new algorithm that automatically searches for better data partitions, dynamically changing the number of clusters.
- The proposed algorithm uses ITL concepts in new operators that adjust previously formed clusters.
- Finally, a new clustering assessment index was proposed, based on the Silhouette criterion, incorporating the statistics of the input data.

This paper is organized as follows: basic concepts of Information Theory and Artificial Immune Networks are discussed in Section 2. Detailed implementation of the proposed clustering algorithm is described in Section 3. Section 4 shows the experimental methodology adopted in this paper. The experimental results are presented and discussed in Section 5. Finally, the conclusion and future works are shown in Section 6.

2 Theory: modeling and methods

Clustering is an unsupervised task that can be applied to several fields, in particular pattern recognition, image segmentation, and trend detection [Aggarwal and Reddy, 2014]. The main focus of clustering is, given a dataset of N objects, organizing these objects into K clusters, putting similar objects into the same cluster and different objects into distinct clusters [Zhang *et al.*, 2013].

Real-world datasets may present several characteristics for each object, which allows them to be clustered in different ways. In image datasets, for example, data objects may be clustered according to their shapes, colors, positioning, etc [Wang *et al.*, 2019], as shown in **Figure 1**. Algorithms must be prepared to search for multiple feasible partitions that represent the different aspects of such datasets.

Figure 1. Image samples of the ALOI dataset [Geusebroek *et al.*, 2005].

Several methodologies are used to obtain diverse partitions of datasets. The *Unguided Generation* [Aggarwal and Reddy, 2014] methodology runs algorithms with different input parameters and combines their results.

Clustering for Alternatives with Mutual Information (CAMI) [Dang and Bailey, 2010] is one example of Unguided Generation. It generates two clusterings per execution maximizing the likelihood of both partitions and, at the same time, minimizing the Mutual Information between them using a Gaussian mixture of models.

Other algorithms use subspace search strategies, like MISC [Wang *et al.*, 2019]. It tries to find subspaces between high-dimension spaces and one partition is found in each subspace. Some strategies employ the use of constraints, where a preliminary partition is presented and an alternative solution is sought as in the *Constrained Orthogonal Average Link Algorithm* (COALA) [Bae and Bailey, 2006]. This type of strategy imposes the need to know the previous partition.

Other strategies use the same algorithm to search for alternative partitions, with different inputs yielding different outcomes, as shown in Ferraria *et al.* [2023]. Thus, an example is the use of vector quantization to find different prototypes in data to be used as input data to the algorithms [Hossain *et al.*, 2013].

For this type of strategy, there is the problem of the algorithm used for the vector quantization, which is generally the K-Means. This kind of algorithm is known to work well in elliptical data distribution.

2.1 Information Theoretic Learning for clustering

A common problem for many data analysts lies in how to extract information contained in datasets [Principe, 2010]. Such information may provide new perspectives about the studied subject. One approach that inherently tries to tackle this problem is clustering, which has a multitude of different approaches in the literature.

One of these approaches is Information Theoretic Learning, which presents mechanisms that obtain data information directly from their statistical properties. Therefore, ITL allows the identification of new relationships of data samples [Principe, 2010].

As the name suggests, ITL is based on Information Theory, which has descriptors to quantify the information contained in data samples. One of the most well-known descriptors is entropy.

Information Entropy was originally proposed by Shannon [1948], and its concept was later expanded by different authors [Rényi, 1961; Havrda and Charvát, 1967]. It can be seen as a measure of uncertainty associated with a random variable, so it evaluates how much of an event is due to randomness [Araújo *et al.*, 2013].

Renyi generalized Shannon's Entropy including a free parameter (α) [Principe, 2010]. So, given a stochastic variable $\overrightarrow{x}$ with probability density function (PDF) $f(\overrightarrow{x})$, $\alpha \geq 0$, and $\alpha \neq 1$, Renyi's Entropy can be defined as in Equation 1 [Araújo *et al.*, 2013].

$$H_R(\overrightarrow{x}) = \frac{1}{1-\alpha} log \int f^{\alpha}(\overrightarrow{x}) d\overrightarrow{x}. \quad (1)$$

Renyi's Entropy has been used in clustering algorithms with α set to 2 [Zhang *et al.*, 2013]. Thus, it is known as *Renyi's Quadratic Entropy*. The PDF $f(X)$ can be estimated directly from the data, with non-parametric approaches such as the Parzen Window [Silva *et al.*, 2015], as depicted in Equation 2. Symmetric Gaussian Kernel is usually adopted as the Parzen Window G_σ [Zhang *et al.*, 2013], and σ is its covariance.

$$f(\overrightarrow{x}) = \frac{1}{N} \sum_{i=1}^{N} G_\sigma(\overrightarrow{x} - \overrightarrow{x_i}) \quad (2)$$

where N is the number of samples of the dataset, G_σ is a Gaussian function with σ covariance.

Substituting Equation 2 into Equation 1 and considering $\alpha = 2$ (Renyi's Quadratic Entropy), we get Equation 3. The argument of the logarithm is the *quadratic information potential estimator* ($V(X)$), given in Equation 4 [Principe, 2010].

$$\begin{aligned} H_R(X) &= -log \int f^2(\overrightarrow{x}) d\overrightarrow{x} \\ &= -log \int \left(\frac{1}{N} \sum_{i=1}^{N} G_\sigma(\overrightarrow{x} - \overrightarrow{x_i}) \right) \\ &\quad \left(\frac{1}{N} \sum_{i=1}^{N} G_\sigma(\overrightarrow{x} - \overrightarrow{x_j}) \right) d\overrightarrow{x} \\ &= -log \left(\frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} G_{2\sigma}(\overrightarrow{x_i} - \overrightarrow{x_j}) \right) \quad (3) \end{aligned}$$

$$V(X) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} G_{2\sigma}(\overrightarrow{x_i} - \overrightarrow{x_j}) \quad (4)$$

where $X = \{\overrightarrow{x_1}, ..., \overrightarrow{x_N}\}$, $\overrightarrow{x_i} \in R^d$ and $i = 1, .., N$ is a set of N objects of a dataset in a d-dimensional space. $V(.)$ can be used as the Information Potential Estimator of a dataset but also as a cluster or pairwise estimator of objects.

Besides the quadratic information potential estimator, other descriptors were developed for clustering. Two of them are the *Cross Information Potential* (CIP) [Principe, 2010] and the *Differential Entropy Clustering* [Jenssen, 2010; Principe, 2010]. The CIP was already used as a clustering index by Principe [2010], Araújo *et al.* [2013], and Borges and Coelho [2018].

The *Differential Entropy Clustering* descriptor uses entropy in a way so that an object that is not yet labeled can be assigned to a particular cluster. To illustrate how it works, consider **Figure 2**, in which C_1 and C_2 are distinct clusters with independent objects. A new data object x must be clustered into C_1 or C_2.

In this context, Differential Entropy Clustering proposes that x must be assigned to the cluster that presents the lowest entropy after x is assigned to it. So, if x is wrongly assigned to C_2 its entropy increases more than C_1. Therefore x must be assigned to C_1. Equation 5 shows the general case, with C_k clusters.

$$H(C_i + \overrightarrow{x}) - H(C_i) \leq H(C_k + \overrightarrow{x}) - H(C_k), \forall k \quad (5)$$

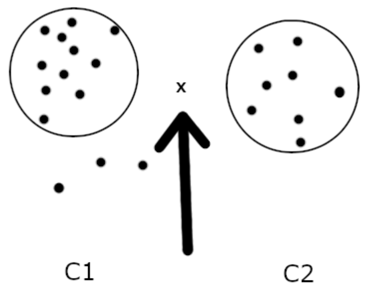


Figure 2. Using Differential Entropy Clustering to decide whether object x should be assigned to cluster C_1 or C_2.

where $C_k (k = 1, ..., K)$ is a set of clusters, K is the number of clusters, i is one of the clusters in the partition ($i \neq k$) and $H(C_k)$ represents the entropy of cluster C_k. So if object $\overrightarrow{x}$ is correctly assigned to a cluster i, the entropy of this cluster will increase less than the entropy of all other clusters in K.

2.2 Artificial Immune Networks

The Immune Network Theory, proposed by Jerne [1974], states that the immune system of vertebrates works as a self-organizing network, aiming to maintain its stability. The Immune Network Theory also states that cells of the immune system can also recognize each other, which may lead to positive or negative responses. The positive response is a stimulus for cell proliferation, while the negative response may even suppress the cells of the system [Borges *et al*., 2012].

These mechanisms can be an interesting inspiration for the development of algorithms since they present characteristics such as the stimulation of population diversity, learning (which can be seen as a micro-evolution), and memory [de Castro and Von Zuben, 2002]. These characteristics are important for the context of this research.

One of the first immune network-inspired algorithms for data analysis was aiNet (*Artificial Immune Network*), proposed by de Castro and Von Zuben [2002] for clustering. The immune-network model proposed in aiNet was later expanded and adapted to different problems, such as optimization and biclustering, leading to a broad family of algorithms [de França *et al*., 2010; Borges *et al*., 2012; Borges and Coelho, 2018].

One of these algorithms is the *Optimal Clustering Optimization aiNet* (ocopt-aiNet), developed by Borges *et al.* [2012] for clustering and later extended by the *Cross Information Potential aiNet* (CIP-aiNet) [Borges and Coelho, 2018].

The CIP-aiNet uses CIP as an inter-cluster evaluation function to assess the partitions. It has the drawback of only evaluating the clusters's separation, not considering their compactness.

So, as the ocopt-aiNet algorithm and its extensions can maintain solutions with diversity, their structure can be used as a basis for the development of new techniques for alternative clustering. Therefore, its details are shown in the next subsection.

2.2.1 The ocopt-aiNet algorithm

The algorithm starts with a randomly generated initial population, which is evolved through cloning, hypermutation, suppression, and memory maintenance strategies.

In ocopt-aiNet, each candidate solution is encoded as illustrated in **Figure 3**, for a hypothetical dataset with 15 objects: each individual is an integer-value array of size equal to the number of objects in the dataset, and the values in each position correspond to the label of the cluster to which that object must be assigned.

1	1	1	3	3	4	4	2	2	2	4	4	2	2	2	Individual 1
1	1	2	2	2	2	2	3	3	3	2	2	3	3	3	Individual 2
2	2	2	2	3	3	3	3	3	3	3	3	3	3	3	Individual 3
1	1	1	1	2	2	3	3	3	3	2	3	3	3	3	Individual 4

Figure 3. Example of 4 candidate solutions encoded in ocopt-aiNet's population.

Then, the algorithm assesses the individuals using an adaptation of the Silhouette Criterion. Commonly, the original Silhouette Criterion [Rousseeuw, 1987] and its adaptation use the Euclidean distance as a measure of dissimilarity between objects. In the ocopt-aiNet's adaptation, the cluster center is used, while the original Silhouette uses all the objects of a cluster. Equation 6 describes the formulation of the Silhouette Criterion adopted in ocopt-aiNet.

$$S_{\overrightarrow{x_j}} = \frac{b_{q,j} - a_{p,j}}{max(a_{p,j}, b_{q,j})} \tag{6}$$

where $S_{\overrightarrow{x_j}}$ is the Silhouette value for object $\overrightarrow{x_j}$ in cluster p, $a_{p,j}$ is the distance of object $\overrightarrow{x_j}$ to the center of cluster p, and $b_{q,j}$ is the distance between object $\overrightarrow{x_j}$ and the center of the closest cluster q where $q \neq p$. The average of $S_{\overrightarrow{x_j}}$ for all objects in the dataset represents the Silhouette Criterion of a specific partition.

ocopt-aiNet maximizes the average Silhouette Criterion during the search to get better partitions for a given dataset. Therefore, with a single criterion, it assesses intra- and inter-cluster distances.

After each individual is evaluated, they are cloned and undergo a mutation process. Each clone is mutated using a unique mutation operator, and the choice of operators to be used is made randomly, with probability of occurrence, 25, 25, and 50% for the following operators, respectively:

- **Operator 1:** increase the number of clusters of the individual by selecting one cluster, with more than two objects, and dividing it into two new clusters.
- **Operator 2:** cluster suppression, which eliminates one cluster from an individual with more than two clusters. In this case, the objects belonging to the chosen cluster are reallocated to the nearest cluster.
- **Operator 3:** object adjustment occurs in 10% of the objects, identifying the best cluster that the object should be assigned to. To do so, the algorithm chooses the cluster with the shortest distance between the object and the cluster's center.

After the application of the mutation operators, each clone is evaluated and only the best individuals are kept for the next iteration. This process is carried out until the average fitness stabilizes. After stabilization, a suppression and insertion process occurs.

The suppression is used when equal individuals in the population are eliminated. For insertion, the algorithm will use the 10% growth rate for the current population, and at least 1 new individual is created.

Using the structure of the aiNet framework as a basis, it is possible to develop new operators that are capable of exploring the search space to find new partitions, in addition to maintaining solutions with diversity. And, based on Information Theory concepts for cluster evaluation, it is possible to investigate new structures of the spatial distribution of datasets.

3 Proposed Approaches

The new algorithm called *Information-Theoretic Learning Artificial Immune Network* (ITL-aiNet) has a basic structure similar to ocopt-aiNet (Section 2.2.1) but modifies the mutation operators, evaluation of solutions, clustering criteria and maintenance of alternative solutions.

The representation of individuals in the population is similar to the one illustrated in **Figure 3**, and the overall steps of the new algorithm are given in **Algorithm 1**.

Algorithm 1 ITL-aiNet algorithm

```
Input X: Dataset
σ: Kernel size
P: Population size
K: Initial number of clusters
C: Number of clones
I: Number of internal iterations
Z: Number of global iterations.
Output Set of partitions.
Create P individuals with K random labels
Build a matrix M(i, j)_{N×N} using Equation 4 where i ≠ j
repeat
    repeat
        Fitness assessment using the ITL-based Silhouette Coefficient
        Creation of C clones for each of the P individuals
        Application of mutation operators
        Fitness assessment of clones through the ITL-based Silhouette Coefficient
        Selection of the best individuals
    until I is reached
    Evaluation of the affinity between individuals
    Suppression of individuals with high affinity
    Creation of new individuals to maintain population size P
until Z is reached
Return: Final set of individuals (each corresponding to a possible partition)
```

In **Algorithm 1**, the initialization of the population with P individuals occurs randomly with a maximum of K different clusters. Matrix $M_{N\times N}$ contains the quadratic information potential estimator V obtained by the evaluation of Equation 4 between all objects.

The fitness value corresponds to the internal quality of the partition and is given by the new index adapted from the Silhouette criterion (further described in Section 3.1), where σ represents the Parzen Window width. Then, C clones are created for each individual in the population and the mutation operators are applied.

All altered clones are then evaluated with the adapted Silhouette and only those individuals with the highest fitness are maintained in the population for the next iteration.

Once the local iteration is complete, the affinity between individuals is analyzed, and those with high affinity (similarity) are suppressed, so new randomly created individuals are inserted into the network, replacing the suppressed ones. The algorithm will terminate after a given number of global iterations.

The inclusion of new individuals increases the number of new candidate solutions up to P. This allows the control of the size of the population, differently from the ocopt-aiNet, which has no population size limits. Also, in ITL-aiNet, the number of iterations is defined by a parameter and no longer by the convergence of algorithms, which can be difficult to estimate using a threshold. Section 4.3 depicts a convergence study of the proposed algorithm.

3.1 ITL-based Silhouette Coefficient

For the adaptation of the Silhouette coefficient, we use the *quadratic information potential estimator* (Equation 4). Therefore, when two samples are compared, the information potential between them, provided by the symmetric Gaussian kernel, is obtained.

The proposed index is presented in Equation 7. As in the original Silhouette, higher values indicate better partitions of a given dataset.

$$S'(X) = \frac{1}{N}\left(\sum_{i=1}^{N}(V_{intra}(\overrightarrow{x}_i) - V_{inter}(\overrightarrow{x}_i))\right) \quad (7)$$

Equation 8 calculates the value of the information potential of the object $\overrightarrow{x}_i$ with all objects of cluster c. So, as we have the average of the information potential between object $\overrightarrow{x}_i$ and all objects of its cluster, it represents cluster c intra-cluster information potential (V_{intra}).

$$V_{intra}(\overrightarrow{x_i}) = \frac{1}{N_c - 1}\sum_{j=1}^{N_c} G_{2\sigma^2}(\overrightarrow{x}_i - \overrightarrow{x}_j) \quad (8)$$

where N_c is the number of objects in cluster c, $\overrightarrow{x}_i$ and $\overrightarrow{x}_j$ are objects allocated to c, and $i \neq j$.

Equation 9 represents the inter-cluster information potential (V_{inter}): the average of the information potential of object $\overrightarrow{x}_i$ and all objects of a cluster d (cluster d has the highest average information potential with object $\overrightarrow{x_i}$).

$$V_{inter}(\overrightarrow{x_i}) = \frac{1}{N_d} \sum_{l=1}^{N_d} G_{2\sigma^2}(\overrightarrow{x}_i - \overrightarrow{x}_l) \tag{9}$$

where N_d is the number of objects in cluster d, $\overrightarrow{x}_l \in d$, and $\overrightarrow{x}_i \notin d$.

3.2 Mutation operators

The mutation operators of ITL-aiNet are applied to the clones generated for each individual in the population, to create, suppress, or rearrange clusters.

3.2.1 Cluster Suppression

The first operator aims to delete one cluster from the candidate solution and it can only be applied to clones with more than 2 clusters. This operator randomly selects an existing cluster in the individual and suppresses it. Therefore, each object originally allocated to the suppressed cluster must be reallocated to an existing cluster.

The reallocation process is made according to the Differential Entropy Clustering given in Equation 5: each object to be reallocated is evaluated with Equation 5 and inserted into the cluster that presents the lowest entropy gain.

3.2.2 Cluster Division

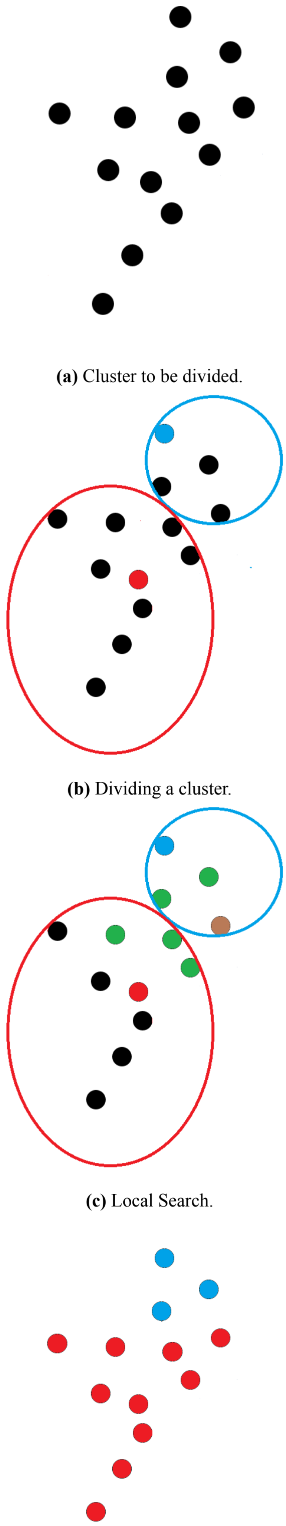
The second operator aims to create, through division, new clusters in a candidate solution, thus it can only be applied to clusters with more than 2 objects allocated to it. To illustrate how this operator works, consider the cluster depicted in **Figure 4a**.

Two objects are chosen to form two new clusters. The first object has the highest information potential among the cluster and the second object contains the least information potential to the first selected object. Thus, the algorithm first selects object A (red point, **Figure 4b**) that has the highest average quadratic information potential compared to the other objects in the cluster, and B (blue point), which has the lowest quadratic information potential when compared to A.

A and B are the initial objects of the two new clusters. For each of the remaining objects, the quadratic information potential is evaluated with A and B, and the highest value indicates to which cluster that object must be assigned. In **Figure 4b**, the two new clusters are marked in red and blue lines.

This operator also refines the new cluster generated by B (blue point). This refinement has a 10% probability of occurring. To implement this local search, we used the process known as *Neighborhood Analysis*, adapted from the work of Zhang *et al*. [2013].

The neighbors are selected independently of clusters, and even clusters that were not evaluated by this operator are analyzed. The neighbors of an object are selected based solely on the criterion of higher *Information Potential* to the analyzed object. Once the neighboring objects are found, the label of each neighbor is checked; if the majority of neighbors are in the same cluster as the analyzed object, no changes are made.



(a) Cluster to be divided.

(b) Dividing a cluster.

(c) Local Search.

(d) Final Local Search.

Figure 4. Cluster Division.

Otherwise, if the neighbors belong to a cluster different than the analyzed object, the cluster label of this object will be changed to this new cluster. Thus, it does not matter if the new cluster was one of those currently created or was already a pre-existing cluster in the partition. With this operation, the algorithm ensures that the clusters have greater cohesion.

Figure 4c depicts the objects allocated to the new cluster generated by the base object B (blue line). This cluster will be reassessed by the local search process using this *Neighborhood Analysis* process.

For the *ITL-aiNet* algorithm, 5 neighbors with the highest *Information Potential* are selected. Therefore, in **Figure 4c**, the objects in green represent those that are considered neighbors for the analysis of the selected object (brown point).

It is possible to verify that there are more neighbors belonging to the cluster delineated by the red line; thus, the selected object should have its label changed to this new cluster. **Figure 4d** illustrates the final configuration of the clusters, where the blue points indicate one cluster and the red points represent the other cluster formed.

Finally, in case of a tie, the cluster with the highest average value of information potential with the analyzed object is selected. To exemplify this case: consider that the object selected for evaluation has 2 neighboring objects that are in cluster C, 2 other neighboring objects that are in cluster D, and 1 neighboring object in cluster E; in the first two cases, there was a tie between C and D, so the cluster in which the analyzed object has the highest average information potential between C and D will be selected.

3.2.3 Cluster Refinement

The third operator aims to refine the clusters presented in a candidate solution. When this operator is selected, a new step is started: each object of the dataset has a 10% probability of being evaluated, regardless of their original clusters, and, through Differential Entropy Clustering (Equation 5), their possible reallocation to all other clusters of the solution is evaluated.

3.3 ITL-aiNet complexity

The evaluation process is dependent on the number N of objects in the dataset and the population size P. In this process, sums are used in most calculations. There is no comparison between individuals in this step, so there is $Z \times I \times P \times N^2$ evaluations.

The step with a longer processing time is the application of the mutation operators in each of the C clones, which is dependent on the population size P. In this case, there are $P \times C$ individuals to be mutated and each operator has different complexities:

- In Cluster Suppression, when a cluster c is suppressed, there are N_c comparisons to be done, with K clusters still existing in this individual, so the number of comparisons is $N_c \times (\sum_{i=0}^{K}(N_i)) > N$, where $(\sum_{i=0}^{K}(N_i)) + N_c = N$.
- In Cluster Division, there are N_c^2 comparisons. In the local search the number of comparisons is $N_c' \times N$, where N is the number of objects in the dataset and N_c' is the number of objects in the new cluster generated through mutation, which is less than N_c.
- In Cluster Refinement, only 10% of the objects are evaluated (N_a) and compared to all clusters K in the individual. The number of comparisons is $N_a \times (\sum_{i=0}^{K}(N_i))$.

The algorithm must compare each individual to its clones. This step uses the fitness value of each clone, so it performs $P \times C$ comparisons. In the suppression, the algorithm compares each individual and each object, so it performs $Z \times P \times N^2$ comparisons.

The number of individuals is maintained at each iteration, thus worst-case complexity becomes smaller than ocopt-aiNet's, which does not have population control.

In summary, the computational complexity of ITL-aiNet is shown in Expression 10. Regardless of the mutation operator chosen, its complexity is dominated by the quadratic cost of computing the evaluation function or comparing individuals in the suppression step.

$$O(Z \times I \times P \times N^2 + P \times C + Z \times P \times N^2) \qquad (10)$$

4 Materials and Methods

This section presents the datasets used in the experiments, the algorithms compared to the ITL-aiNet, and the assessment indices. The proposed algorithm was implemented in Java (JDK 1.8.0) and all the experiments were executed in a PC with an Intel Core i7-8565U processor (Quad-core CPU), 8 GB of RAM, and Windows 10.

4.1 Datasets

Experiments were performed on 5 datasets with different characteristics. The first dataset, presented in **Figure 5a**, is a synthetic dataset with 1024 two-attribute objects, divided into four clusters. For this dataset, 4 clusters are expected. Some alternative solutions must be possible using different initial parameters, to analyze the maintenance of compactness of known clusters.

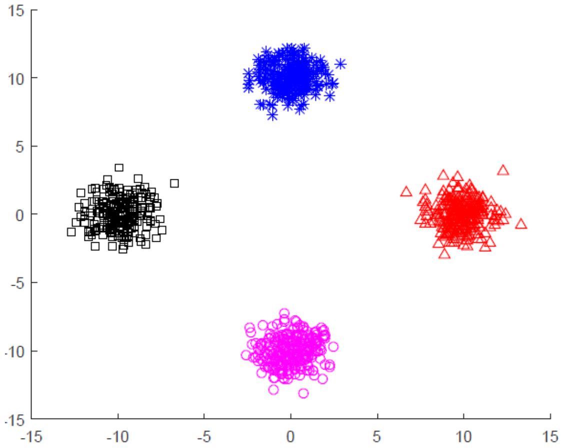
The second dataset, named “Flower”, is the set of pixels of a single image, given in **Figure 5b** [Wu *et al.*, 2018], in which the algorithm must be able to cluster the pixels to create a sub-image. The RGB values of each pixel are taken as a single object and converted into black and white pixels. This results in a dataset of 256 samples and 3 attributes and only partitions with 2 clusters were analyzed.

This experiment analyzes how algorithms behave in image segmentation. In this case, there are no labels available, and one possibility would be to verify the segmentation into background and foreground [Niu *et al.*, 2014].

The third dataset, CMUFace [Dua and Graff, 2017], contains 640 32 $\times$ 30 pixel images of 32 different faces, containing 20 images of each face. Three different faces were manually selected (illustrated in **Figure 5c**) randomly and without any prior analysis. These faces feature multiple images from different positions (straight, left, right, up), with distinct expressions (neutral, happy, sad, angry), and with or without sunglasses.

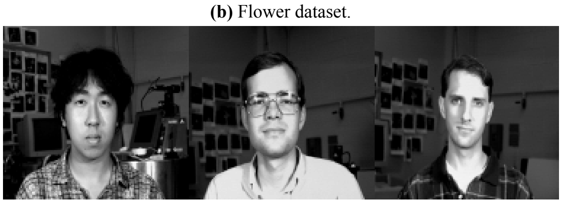
For this dataset, the possible solutions, besides the clustering of images of the same person, would be the variations of positions or the differentiation by the existence or not of sunglasses.

The fourth dataset is the “Amsterdam Library of Object Images” (ALOI) [Geusebroek *et al.*, 2005]. The original



(a) Synthetic dataset.

(b) Flower dataset.



(c) Example of the CMUFace dataset.

Figure 5. Illustrations of three datasets of the experiments.

dataset has 1000 objects, with 110,250 images but, here, 9 objects that have similar features between shape and color were chosen, with 12 images containing different illumination variations of each object. The 9 objects are presented in **Figure 1**. These objects can be clustered either by shape or color.

Finally, the last dataset is the *WebKb* [1] dataset. This dataset corresponds to a collection of 1041 HTML documents from four universities. These documents were preprocessed and 500 words were maintained, then PCA was applied, keeping 87% of variance, resulting in a dataset with 1041 objects and 100 features. The documents can be clustered based on topics or university names.

In the experiments performed here, the attributes of all datasets were normalized into [-1, 1].

4.2 Algorithms

To compare the results of the ITL-aiNet with the literature, we chose well-established algorithms that perform clustering in different ways, namely the classic K-Means [Jain, 2010], KDAC [Wu *et al.*, 2018], MISC [Wang *et al.*, 2019], the bioinspired algorithms copt-Bees-MO [Cunha *et al.*, 2017] and ocopt-aiNet [Borges *et al.*, 2012], and, finally, MinCEntropy [Vinh and Epps, 2010].

The implementation of KDAC, MISC, and MinCEntropy algorithms, were obtained from their official websites, while the implementations of ocopt-aiNet and copt-Bees-MO were provided by their authors.

The copt-Bees-MO and ocopt-aiNet are based on the Euclidean distance to evaluate the dissimilarities between data objects, and ocopt-aiNet uses the Silhouette criterion, adapted with cluster center distance, to assess the quality of each proposed partition. MinCEntropy, on the other hand, is based on Information Theory metrics for clustering quality evaluation.

Table 1 shows the parameter settings adopted in the comparisons for the algorithms that have dataset-dependent values. The configuration of ITL-aiNet was defined through an initial set of experiments with all datasets.

Compared to other algorithms, ITL-aiNet has a larger number of input parameters, but only σ is difficult to adjust. A sensitivity analysis is carried out on this parameter in Section 4.3.

Table 1. Parameter settings.

	Synthetic	Flower	Face	ALOI	WebKb
ITL-aiNet	K=10 P=5 C=5 I=50 Z=2 σ=1.4	K=10 P=5 C=5 I=50 Z=2 σ=1	K=10 P=5 C=5 I=50 Z=2 σ=7	K=10 P=5 C=5 I=50 Z=2 σ=10	K=10 P=5 C=5 I=50 Z=2 σ=10
K-Means	K={4,2}	K=2	K={3,4}	K=3	K=4
KDAC	K_1=4 K_a=2	K_1=2 K_a=2	K_1=3 K_a=4	K_1=3 K_a=3	K_1=4 K_a=4
MinCEntropy	K_1=4 K_a=2	K_1=2 K_a=2	K_1=3 K_a=4	K_1=3 K_a=3	K_1=4 K_a=4

K-Means has a single solution as output. In cases of alternative partitions with different numbers of clusters, two configurations are used. KDAC and MinCEntropy have as input the number of clusters of the initial and alternative partitions to be generated. As ITL-aiNet, MinCEntropy requires the value σ as input, and for these experiments, we adopted the recommendation by Vinh and Epps [2010].

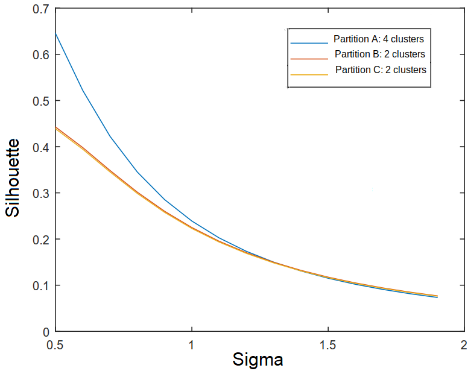
The copt-Bees-MO, ocopt-aiNet, and MISC algorithms were also adjusted as in their original papers [Cunha *et al.*, 2017; Borges *et al.*, 2012; Wang *et al.*, 2019] for all datasets studied in this work.

Finally, all experiments were repeated 10 times for all algorithms and datasets, and for diversity and quality analyses, they were considered as independent experiments. It is important to highlight that, for each independent experiment, K-Means was run with its two configurations, so alternative solutions could be found.

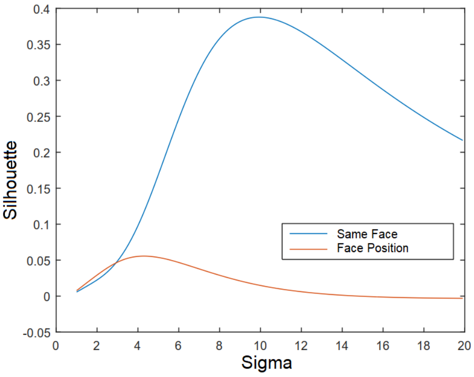
4.3 Sensitivity and convergence analysis

There is one parameter, σ, in ITL-aiNet, which may significantly influence clustering. An analysis of the impact of the variation of this parameter in the proposed Silhouette criterion is shown, considering the configuration of **Algorithm**

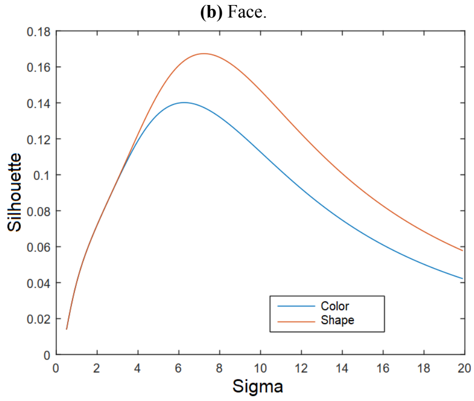
[1] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/



(a) Synthetic (the three partitions are shown in Figure 7).

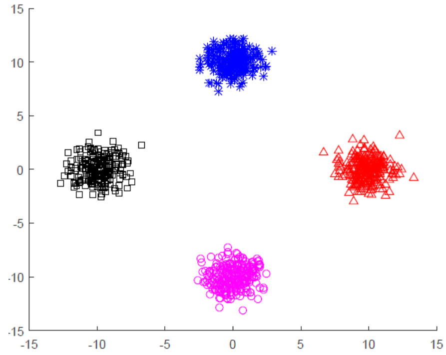


(b) Face.

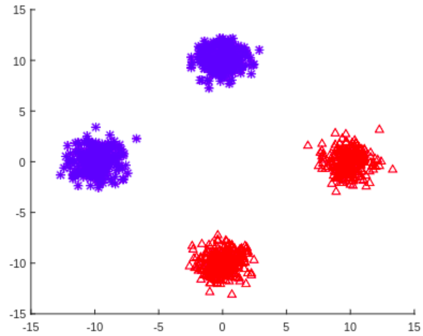


(c) ALOI.

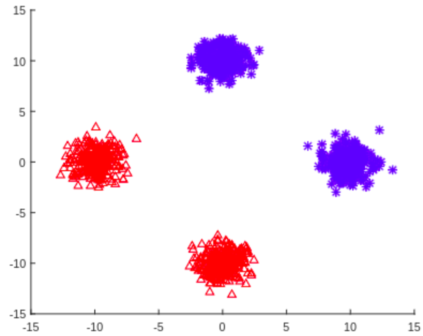
Figure 6. Silhouette behavior with sigma variation.



(a) Partition A.



(b) Partition B.



(c) Partition C.

Figure 7. Partitions found by ITL-aiNet.

1. For these analyses, three datasets that have known partitions were chosen: Synthetic, Face, and ALOI.

Figure 6 shows the behaviors of changing the value of σ for the Synthetic dataset (**Figure 6a**), Face dataset (**Figure 6b**), and ALOI dataset (**Figure 6c**).

For the Synthetic dataset, three possible partition configurations were found by ITL-aiNet: one with 4 clusters and two with 2 clusters (**Figure 7**). Considering these three different partitions, it is possible to analyze, through **Figure 6a**, that smaller σ values result in more distant curves for different clusterings, and as this value increases, the difference decreases.

This is due to the increase in the size of the Parzen Window, which means that objects that had less information potential for smaller σ end up increasing their information potential when σ increases.

Consequently, objects that should be in separate clusters end up being merged into one cluster. As for the other two datasets, the curves for different clustering configurations show greater distances. This occurs due to the different shapes of object clusters in these datasets.

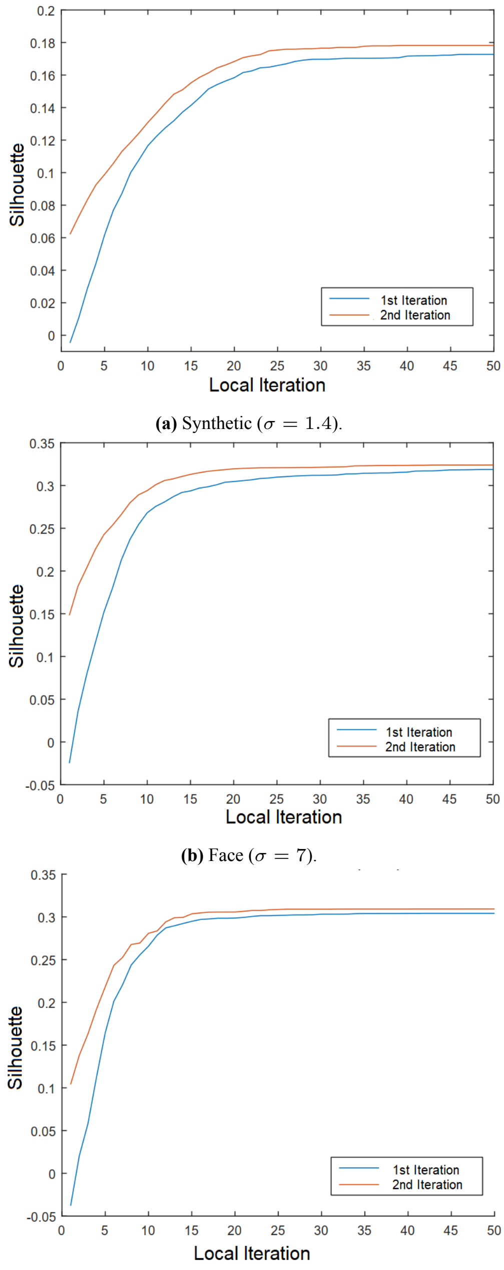
The experiments have the parameter σ chosen after the execution of the ITL-aiNet algorithm with different values, within the range shown in the graphics of **Figures 6**.

When the choice of σ led to results close to the maximum value of Silhouette, the algorithm tended to converge to a similar solution in all alternative solutions, thus intermediate values were selected, which increased the search for diverse solutions.

Figure 8 shows the Silhouette stability (fitness) of ITL-aiNet in both the 1st global iteration and 2nd global iteration. The stability of the average fitness value is reached with 40 iterations in all datasets. Thus, the choice of 50 local iterations is consistent with the results found in the Silhouette stability analysis.

4.4 Performance metrics

Four indices were used to analyze the obtained results: two of them for quality analysis of the partitions and two for diversity evaluation. Quality assessment must take into account the compactness of the clusters (or cohesion) and their sepa-



(a) Synthetic ($\sigma = 1.4$).

(b) Face ($\sigma = 7$).

(c) ALOI ($\sigma = 10$).

Figure 8. Fitness Stability.

ration from the other clusters in the partition.

With this in mind, the Xie-Beni [Xie and Beni, 1991] and the Generalized Dunn's [Bezdek and Pal, 1998] indices were chosen here. For diversity assessment, the Adjusted Rand Index (ARI) [Hubert and Arabie, 1985] and the Normalized Mutual Information (NMI) [Strehl and Ghosh, 2002] were used.

4.5 Statistical tests

Statistical tests were employed to offer more accurate assessments of the obtained results. These results undergo an analysis to show that differences found in the evaluated models are statistically significant.

To determine the statistical significance of variations in the algorithms' performance, the Friedman test [Demšar, 2006] with the corresponding post-hoc tests, was utilized with a significance level of $\alpha = 0.05$.

5 Experimental Results

The results are organized by dataset, and the mean and standard deviation values for each metric are presented for all the algorithms. The results are shown in tables and (↓) indicates that lower values are better and (↑) that higher values are better for each index.

Results in bold highlight the best values. Finally, results in *italic* are statistically different when compared to ITL-aiNet.

5.1 Synthetic dataset

Table 2 presents the diversity and quality indices of solutions found by the considered algorithms when compared with the known partition (ground truth). In this sense, GD, ARI, and NMI must be maximum.

Table 2. Synthetic Dataset: mean and standard deviation of diversity and quality metrics, compared to the ground truth.

	Ground Thruth			
	NMI ↑	ARI ↑	XB ↓	GD ↑
ITL-aiNet	**1.00±0.00**	**1.00±0.00**	**0.01±0.00**	**40.65±0.00**
cOptBees.	0.95±0.07	0.90±0.14	0.06±0.06	24.92±18.48
KDAC	**1.00±0.00**	**1.00±0.00**	**0.01±0.00**	**40.65±0.00**
K-Means	0.61±0.21	0.48±0.42	0.96±3.02	36.60±12.79
MinCEnt.	**1.00±0.00**	**1.00±0.00**	**0.01±0.00**	**40.65±0.00**
MISC	-	-	-	-
ocopt-aiN.	**1.00±0.00**	**1.00±0.00**	**0.01±0.00**	**40.65±0.00**

Comparing the best results of the algorithms, ITL-aiNet, KDAC, MinCEntropy, and ocopt-aiNet found the best solutions in all executions. MISC was not able to find, at any time, a solution that is similar to the ground truth, as it works in order to find subspaces.

As the dimension of this dataset is small, MISC could not obtain the correct partition, so no evaluation was possible. K-Means is highly dependent on the initialization of its centroids and, as it can get stuck in a local optima, which can be noticed in the presented results, ARI and NMI values were high, but not maximum for all runs.

The cOptBees-MO found the ground truth, but not for all executions: this algorithm has a slow convergence when the dataset has a large number of samples, which affects its search for the best solutions, considering a stopping criterion equal to a maximum number of iterations or convergence.

Table 3 presents the results of the alternative partitions, considering different solutions. In this case, greater diversity and high quality are expected. Thus, smaller values of ARI, NMI, and XB are expected, whereas, for the GD index, higher values are better.

ITL-aiNet exhibits greater diversity than the other algorithms indicating that, as a population-based algorithm, it can maintain solutions with high diversity for this type of dataset. The quality is lower compared to algorithms with lower diversity due to the larger number of solutions found.

Since the MISC algorithm did not generate more than one possible solution within each experiment, it was not possible to analyze diversity per experiment; thus, it does not present diversity values.

In the statistical test, ITL-aiNet showed significant differences, in most indices, when compared to the coptBees-MO and KDAC algorithms.

Table 3. Synthetic Dataset: mean and standard deviation of diversity and quality metrics, considering alternative clustering solutions. The best results are in bold and those statistically different from ITL-aiNet ($\alpha = 0.05$) are in italic.

	Alternative Clustering			
	NMI ↓	ARI ↓	XB ↓	GD ↑
ITL-aiNet	**0.57±0.00**	**0.17±0.00**	0.18±0.00	14.71±0.00
cOptBees-MO	*1.00±0.00*	*0.74±0.23*	***0.12±0.04***	1.69±0.19
KDAC	*0.71±0.00*	*0.50±0.00*	*0.13±0.00*	***21.19±0.00***
K-Means	*0.66±0.14*	*0.45±0.15*	0.62±1.53	19.13±6.53
MinCEntropy	0.67±0.01	0.49±0.00	0.14±0.00	21.18±0.00
MISC	-	-	0.26±0.00	1.74±0.00
ocopt-aiNet	*0.75±0.12*	*0.47±0.23*	0.24±0.43	17.91±9.08

Figure 9 depicts the best alternative clustering obtained by the compared algorithms. The choice of the best partition uses the value of the objective function adopted by each algorithm or, in the case of K-Means, the alternative clustering most frequently found in all executions.

coptBees-MO and MinCEntropy do not maintain the compactness between the known clusters, while the ITL-aiNet, MISC, K-Means, KDAC, and ocopt-aiNet do not separate objects of the same cluster, merging different clusters.

5.2 Flower

Table 4 presents the diversity and quality results for the Flower dataset. In this case, there is no ground truth, so greater diversity implies better results. It is worth noting that the ITL-aiNet obtained the highest diversity, considering both diversity indices (NMI and ARI).

ITL-aiNet presented statistically better performance, in terms of diversity, than cOpt-Bees-MO and K-Means. Some algorithms showed low diversity when dealing with this problem or ended up with low-quality solutions.

Table 4. Flower Dataset: Mean and standard deviation of diversity and quality metrics. The best results are in bold and those statistically different from ITL-aiNet ($\alpha = 0.05$) are in italic.

	NMI ↓	ARI ↓	XB ↓	GD ↑
ITL-aiNet	**0.11±0.04**	**0.11±0.33**	0.48±0.66	1.07±0.44
cOptBees-MO	*0.87±0.29*	*0.93±0.14*	0.38±0.06	1.02±0.08
KDAC	*0.90±0.00*	*0.94±0.00*	***0.20±0.00***	**1.53±0.00**
K-Means	*0.63±0.37*	0.56±0.48	0.23±0.02	1.48±0.23
MinCEntropy	0.18±0.24	0.19±0.24	2.12±0.71	0.96±0.19
MISC	0.24±0.04	0.21±0.07	1.62±0.76	0.72±0.26
ocopt-aiNet	0.14±0.02	0.46±0.48	0.26±0.03	1.37±0.22

Figure 10 shows the two best partitions obtained by each algorithm according to their internal evaluation indices. It can be seen that ITL-aiNet and ocopt-aiNet separate the flowers from the leaves. cOptBees-MO has two very close clusters, with practically null diversity, as shown in **Table 4**.

KDAC has a high-quality result, but both the most commonly found images, **Figures 10e** and **10f**, show that only the flowers were found. As K-Means, KDAC, and MinCEntropy found the flowers only. For MISC, **Figure 10k**, shows the flowers, while in its alternative clustering, **Figure 10l**, it shows the borders of the flowers.

5.3 Face

For the Face dataset, there are different ways of clustering, two of which were analyzed here, following the experiments of Vinh and Epps [2010], Wu *et al*. [2018], and Niu *et al*. [2014]. **Table 5** presents the diversity and quality results for the Face dataset compared to the ground truth, and **Table 6** considers alternative solutions.

Table 5. Face dataset: Mean and standard deviation of diversity and quality metrics, compared to ground-truth.

	Ground Truth			
	NMI ↑	ARI ↑	XB ↓	GD ↑
ITL-aiNet	**1.00±0.00**	**1.00±0.00**	**0.27±0.00**	**0.06±0.00**
cOptBees-MO	**1.00±0.00**	**1.00±0.00**	0.30±0.07	**0.06±0.00**
KDAC	**1.00±0.00**	**1.00±0.00**	**0.27±0.00**	**0.06±0.00**
K-Means	**1.00±0.00**	**1.00±0.00**	**0.27±0.00**	**0.06±0.00**
MinCEntropy	**1.00±0.00**	**1.00±0.00**	**0.27±0.00**	**0.06±0.00**
MISC	-	-	-	-
ocopt-aiNet	0.96±0.11	0.94±0.18	0.31±0.12	**0.06±0.01**

In the comparison with the ground truth (**Table 5**), it is possible to observe that, besides the lower quality results of ocopt-aiNet, all the algorithms found the correct partition.

MISC performs matrix operations to find subspaces and then runs the K-Means algorithm to find partitions in each subspace. Given the high dimensionality of this dataset, the algorithm was not able to finish its iterations, due to insufficient memory in the computer where the experiments were made.

For alternative clusterings, the results are different, as shown in **Table 6**. The ITL-aiNet achieves less diversity than most algorithms. The algorithms that obtained greater diversity are MinCEntropy, and KDAC. For quality results, ITL-ainet performs better than MinCEntropy and KDAC, with a statistically significant difference compared to these two algorithms.

For this dataset, using a semi-supervised method, i.e., obtaining an initial solution and using this solution as a comparison to generate an alternative solution, as MinCEntropy and KDAC do, brings greater diversity possibilities. But there are issues in converging to solutions with high quality.

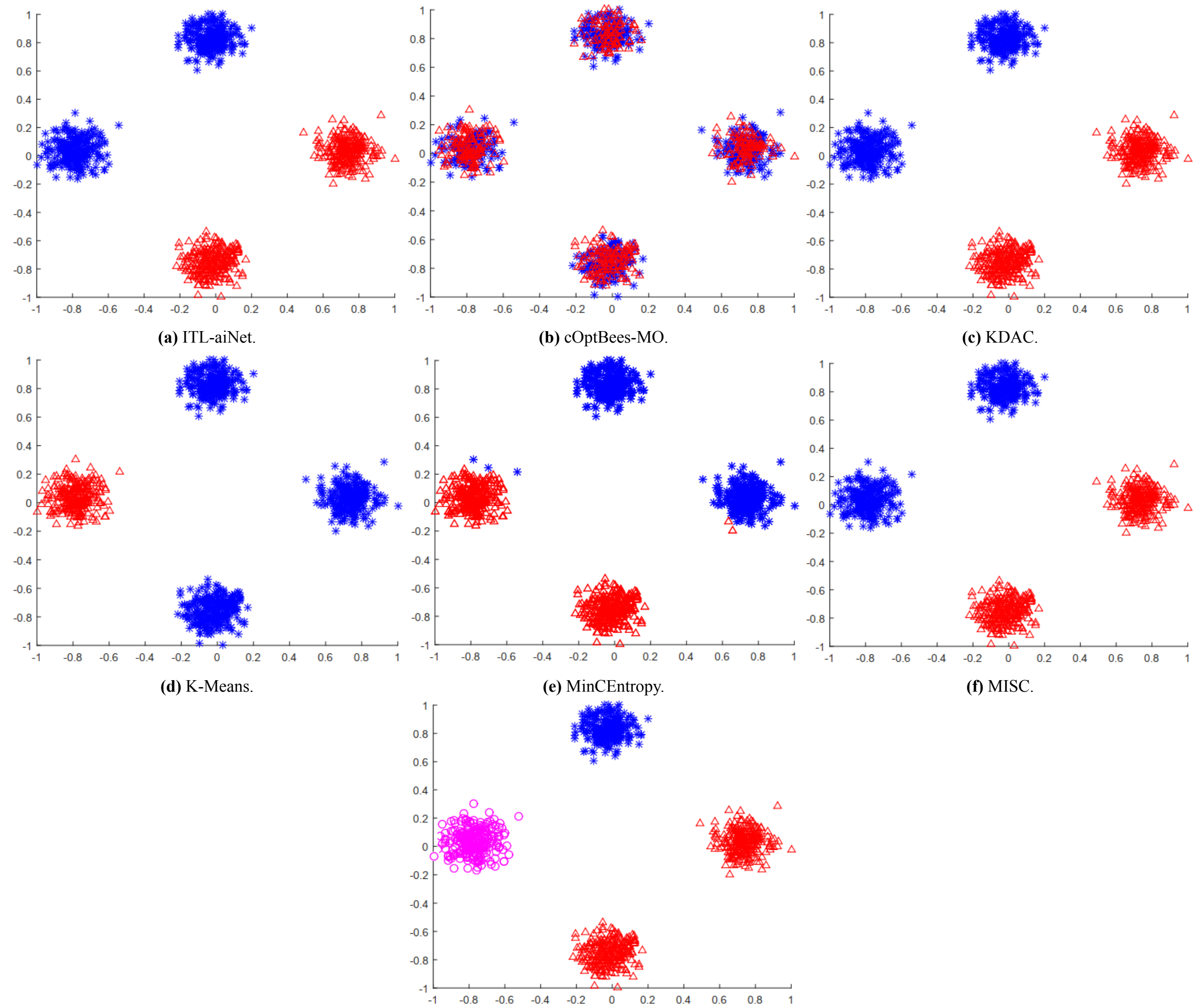
Table 6. Face dataset: Mean and standard deviation of diversity and quality metrics for alternative solutions. The best results are in bold and those statistically different from ITL-aiNet ($\alpha = 0.05$) are in italic.

	Alternative Clustering			
	NMI ↓	ARI ↓	XB ↓	GD ↑
ITL-aiNet	0.70±0.16	0.73±0.10	**0.46±0.04**	**0.05±0.00**
cOptBees-MO	0.88±0.22	0.73±0.18	0.62±0.62	0.05±0.01
KDAC	0.17±0.03	*0.11±0.03*	*1.54±0.07*	*0.04±0.00*
K-Means	0.92±0.01	0.89±0.03	0.52±0.18	0.05±0.01
MinCEntropy	***0.04±0.02***	***0.01±0.02***	*5.34±3.17*	*0.04±0.00*
MISC	-	-	-	-
ocopt-aiNet	0.55±0.11	0.57±0.12	0.56±0.12	0.04±0.00

5.4 ALOI

Table 7 presents the diversity results for the ALOI dataset, considering shape and color clustering. The quality of all solutions found is presented in **Table 8**.

Table 7 shows the comparison between the known color and shape clusterings with the results found by the algorithms. In this case, the results must be as close as possible to the ground truth, so they must have ARI and NMI close to or equal to 1.



(a) ITL-aiNet. **(b)** cOptBees-MO. **(c)** KDAC. **(d)** K-Means. **(e)** MinCEntropy. **(f)** MISC. **(g)** ocopt-aiNet.

Figure 9. Alternative clustering for the Synthetic dataset.

Table 7. ALOI dataset: Mean and standard deviation of diversity metrics to shape and color clustering. The best results are in bold and those statistically different from ITL-aiNet ($\alpha = 0.05$) are in italic.

	Color		Shape	
	NMI ↑	ARI ↑	NMI ↑	ARI ↑
ITL-aiNet	0.32±0.06	0.13±0.04	0.29±0.07	0.09±0.05
cOptBees-MO	***0.58±0.05***	***0.43±0.06***	0.25±0.04	0.12±0.04
KDAC	0.38±0.00	0.22±0.00	0.35±0.00	***0.24±0.00***
K-Means	*0.51±0.18*	*0.34±0.16*	0.30±0.12	0.15±0.13
MinCEntropy	0.39±0.09	*0.31±0.10*	0.25±0.09	0.19±0.09
MISC	-	-	-	-
ocopt-aiNet	*0.56±0.06*	*0.29±0.05*	***0.47±0.10***	*0.22±0.06*

It is possible to verify that the best algorithm was cOptBees-MO for color, while for shape, on average across both indices, the best was ocopt-aiNet. This shows that this dataset may have an elliptical cluster distribution, as both algorithms use the Euclidean distance.

Considering the quality results shown in **Table 8**, ITL-aiNet demonstrated that the new evaluation index can effectively assess the partition's quality, with a high quality result. Statistically, in terms of quality using the GD index, only KDAC did not show a significant difference compared to ITL-aiNet.

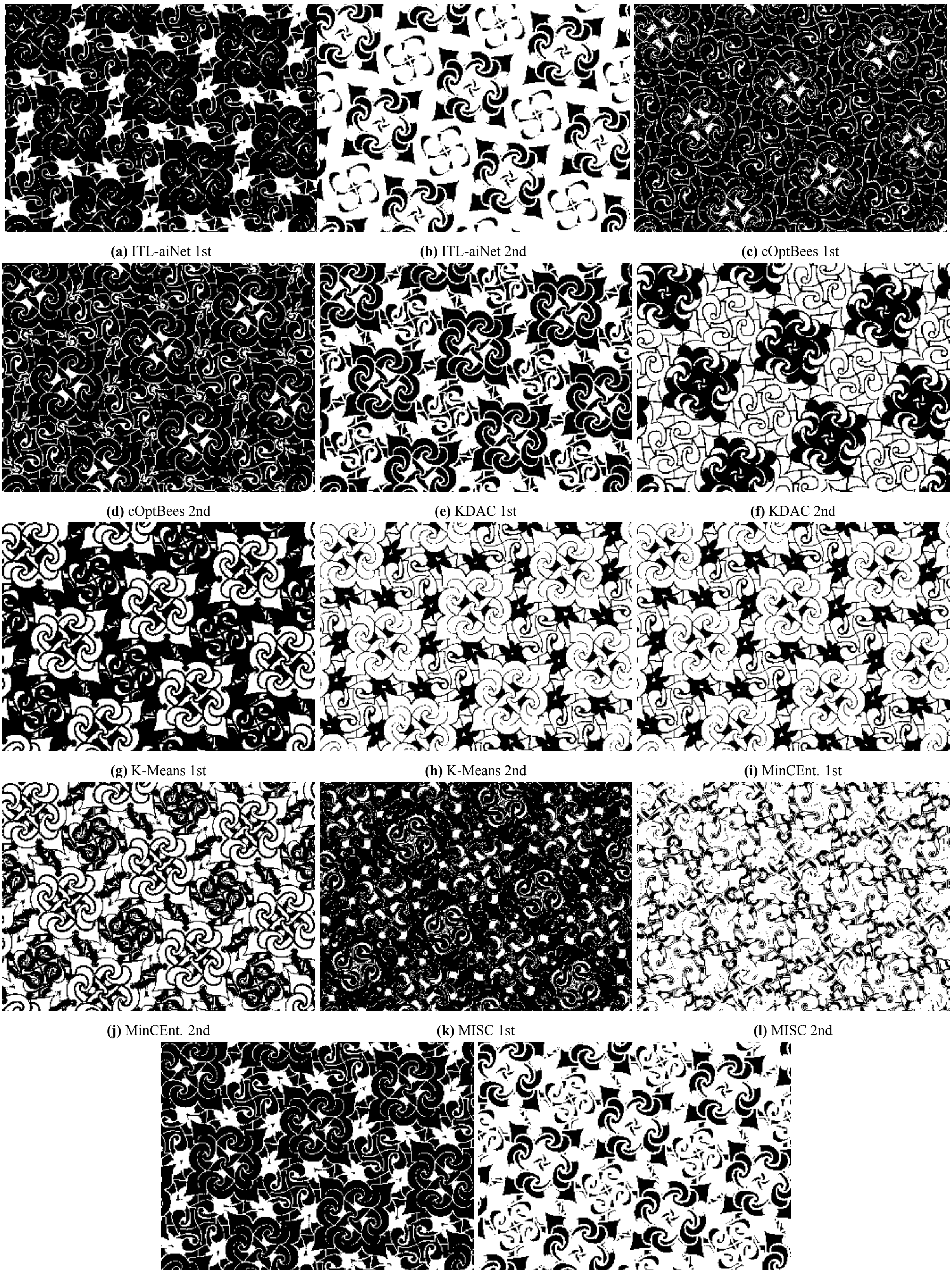
Table 8. ALOI dataset: Mean and standard deviation of quality metrics to shape and color clustering. The best results are in bold and those statistically different from ITL-aiNet ($\alpha = 0.05$) are in italic.

	XB ↓	GD ↑
ITL-aiNet	**0.34±0.01**	**0.055±0.001**
cOptBees-MO	0.79±0.13	*0.031±0.003*
KDAC	0.83±0.01	0.034±0.000
K-Means	0.87±0.20	*0.032±0.006*
MinCEntropy	*1.61±0.31*	*0.020±0.003*
MISC	-	-
ocopt-aiNet	*1.70±1.20*	*0.030±0.004*

5.5 WebKb

Table 9 presents the diversity results for the WebKb dataset, considering clustering by topic and university. The quality measured is presented in **Table 10**.

Table 9 shows the comparison between the known topic and university labels with the results found by the algorithms.



(a) ITL-aiNet 1st **(b)** ITL-aiNet 2nd **(c)** cOptBees 1st
(d) cOptBees 2nd **(e)** KDAC 1st **(f)** KDAC 2nd
(g) K-Means 1st **(h)** K-Means 2nd **(i)** MinCEnt. 1st
(j) MinCEnt. 2nd **(k)** MISC 1st **(l)** MISC 2nd
(m) ocopt 1st **(n)** ocopt 2nd

Figure 10. Results for the Flower dataset.

In this case, the results must be as close as possible to the ground truth. Since all the algorithms have a worse performance compared with the other datasets, it seems that it is more challenging to find the correct clustering.

KDAC has better results with NMI and ARI indices considering the topic labeling but has a worse performance when the university labeling is considered. For university labeling, MinCEntropy got better results than the other algorithms, indicating that partitions with this labeling are easily found by the ITL concepts. ITL-aiNet corroborates that ITL concepts

Table 9. WebKb dataset: Mean and standard deviation of diversity metrics to topic and university clustering. The best results are in bold and those statistically different from ITL-aiNet ($\alpha = 0.05$) are in italic.

	Topic		University	
	NMI ↑	ARI ↑	NMI ↑	ARI ↑
ITL-aiNet	0.02±0.01	0.03±0.01	0.05±0.02	0.03±0.01
cOptBees-MO	0.01±0.01	0.03±0.04	*0.01±0.01*	0.01±0.00
KDAC	***0.07±0.00***	***0.09±0.00***	0.02±0.00	*0.00±0.00*
K-Means	0.04±0.01	0.08±0.02	*0.01±0.00*	*0.00±0.00*
MinCEntropy	***0.07±0.00***	*0.08±0.01*	**0.06±0.04**	**0.04±0.04**
MISC	0.02±0.01	0.04±0.01	*0.02±0.02*	0.00±0.01
ocopt-aiNet	0.02±0.00	0.01±0.01	*0.01±0.00*	*0.00±0.00*

can work with the University label.

For the quality results (shown in **Table 10**) ocopt-aiNet, considering both indices, achieved the best result. It indicates that, in this dataset, there is a clear separation between the clusters, as this algorithm utilizes an evaluation index, the adapted Silhouette, that analyze the objects and the cluster centers.

Table 10. WebKb dataset: Mean and standard deviation of quality metrics. The best results are in bold and those statistically different from ITL-aiNet ($\alpha = 0.05$) are in italic.

	XB ↓	GD ↑
ITL-aiNet	59.05±14.42	0.03±0.01
cOptBees-MO	33.49±13.11	0.05±0.01
KDAC	14.89±0.02	0.05±0.00
K-Means	*7.77±5.53*	0.03±0.01
MinCEntropy	*13.12±11.96*	0.00±0.00
MISC	21.34±23.60	0.01±0.01
ocopt-aiNet	***1.30±1.92***	***0.85±0.14***

5.6 Discussion

Chao *et al*. [2021] indicate that algorithms dealing with Alternative Clustering remain an open area. Thus, in this work it was proposed the implementation of an ITL-based Artificial Immune Network algorithm for alternative clustering: ITL-aiNet.

Considering the Synthetic dataset, which has linearly separable clusters, it obtained the ideal solution as one output. This shows that it can find the correct cluster configuration. ITL-aiNet also obtained the most diverse results in alternative solutions.

For the Flower dataset, ITL-aiNet presented good results for diversity, since it had the best results when compared to the other algorithms. Considering the CMUFace dataset, ITL-aiNet found the correct solution in all executions of the experiments.

For alternative solutions though, ITL-aiNet did not obtain the most diverse results. For the ALOI dataset, ITL-aiNet was better in terms of color than shape, but its performance was far from the best ones. Finally, for the WebKb dataset, which has proved to be the most complex dataset, ITL-aiNet presented better results for the second category (University), with the runner up performance.

Table 11 presents the average elapsed time and the number of solutions (together with the standard deviation) obtained by the proposed algorithm for each dataset.

It is possible to notice that the *WebKb* dataset had a significant impact on the number of solutions, since it has high dimensionality and the largest number of objects, which increases the number of possible solutions and comparisons that must be done.

The number of solutions has low variation for all datasets, which is due to the population control of ITL-aiNet. The highest elapsed time was observed for the datasets that have the largest number of objects, Synthetic and WebKb, showing that the greatest impact on execution times is the number of objects and not the dimensionality of the samples.

Table 11. Number of solutions and Elapsed Time.

	Number of solutions	Elapsed time [mm:ss]
Synthetic	2.50±0.71	08:37±00:20
Flower	2.00±0.67	00:13±00:01
ALOI	2.20±0.92	00:01±00:00
CMUFace	2.50±0.53	00:01±00:00
WebKb	4.90±0.32	09:49±00:44

In terms of algorithm complexities:

- MISC = $O(D \times N(D^2 + D \times N + I \times K + N))$
- KDAC = $O(N^3) + O(I \times N^2 \times D^2)$
- K-Means = $O(N \times K \times D \times I)$
- ITL-aiNet = $O(Z \times X \times P \times N^2 + P \times C + Z \times P \times N^2)$
- ocopt-aiNet = $O(Z^* \times X \times P \times N^2 + P \times C + Z^* \times P \times N^2)$
- MinCEntropy = $O(N^2 \times D) + O(I \times N \times (K + N))$

where D is the dimension of the dataset, N is the number of objects, I is the number of iterations of the algorithm, Z is the number of global iterations, X is the number of internal iterations, P is the size of the population of candidate solutions, and K is the number of clusters.

The computation complexity of the algorithms used in the experiments were acquired using data extracted from the references of each algorithm. The coptBees-MO algorithm is the only one that has no reference regarding its complexity.

It can be noticed that KDAC and MISC algorithms are dependent on the data dimensions since they work to create different feature matrices (or subspaces) and then calculate the cluster distributions. Therefore, the larger number of attributes in a dataset, the higher the complexity for these algorithms. The K-Means algorithm has the lowest complexity, as it has the fewest internal operations.

The MinCEntropy algorithm initially computes the matrix of information potentials among the objects in the dataset. After that, it randomly initializes the first candidate solution and, for each new iteration, updates the cluster labels of the objects to find the best configuration among them.

The ocopt-aiNet and ITL-aiNet algorithms are very close in terms of computational complexity. After calculating the distance matrix (ocopt-aiNet) and the Information Potentials (ITL-aiNet) among objects, only summations are required.

However, the ocopt-aiNet algorithm depends on the convergence of the average fitness values of the population to finish (Z^* and X^*) , which increases its complexity.

Since the number of individuals in ocopt-aiNet's population P is variable, it can increase without restrictions, while in ITL-aiNet, the number of iterations is predefined and the number of individuals is not altered during iterations.

Therefore, considering the complexity of the algorithms, both KDAC and MISC have the highest complexity for data with larger dimensions. MinCEntropy and K-Means, on the other hand, have the lowest complexity due to the fewer number of operations they perform, and as they progress, the cluster label changes for the objects decreases.

Meanwhile, ITL-aiNet and ocopt-aiNet tend to be more computationally expensive when the number of objects is higher.

6 Conclusion

Clustering itself can be considered a complex task, and clustering objects of a dataset in different ways is an actual demand to find new patterns within the same samples. This work proposed a new bioinspired algorithm based on Jerne's immune network theory, on the aiNet framework, and on ITL concepts: ITL-aiNet.

ITL-aiNet aims to find and maintain high-quality and alternative clustering solutions. Besides, a new clustering validation index, based on a combination of the Silhouette criterion with Information Theory aspects, was used as the ITL-aiNet inner criterion to find new partitions for a dataset.

ITL-aiNet includes adapted mutation operators to use Information Theory concepts, thus being compatible with the modified Silhouette criterion.

Using different evaluation indices, it was possible to notice that ITL-aiNet maintains high-quality clusters for image datasets. For those datasets with a smaller number of attributes, the diversity found was better compared to the algorithms used in the experiments.

Therefore, it is possible to conclude that the proposed algorithm can find a good partition configuration regarding diversity in datasets with a smaller number of attributes and achieves better quality as the number of attributes increases.

As for future work, the mutation operators and the new evaluation criterion proposed here can be adapted to different clustering tasks, such as clustering real-time unbounded data streams. Furthermore, seeking ways to adapt parameter configurations automatically could definitely improve the overall performance of ITL-aiNet.

Declarations

Acknowledgements

The authors thank Espaço da Escrita – Pró-Reitoria de Pesquisa – UNICAMP - for the language services provided.

Funding

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Authors' Contributions

The first draft of the manuscript was written by the first author and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The datasets generated and/or analysed during the current study and the source codes of ITL-aiNet are available in here.

References

Aggarwal, C. C. and Reddy, C. K., editors (2014). *Data Clustering*. Chapman and Hall/CRC, Boca Raton, 1 edition. DOI: 10.1201/9781315373515.

Araújo, D., Neto, A. D., and Martins, A. (2013). Information-theoretic clustering: A representative and evolutionary approach. *Expert Systems with Applications*, 40(10):4190–4205. DOI: 10.1016/j.eswa.2013.01.027.

Bae, E. and Bailey, J. (2006). COALA: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 53–62. DOI: 10.1109/ICDM.2006.37.

Bezdek, J. C. and Pal, N. R. (1998). Some new indexes of cluster validity. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 28(3):301–315. DOI: 10.1109/3477.678624.

Borges, E. and Coelho, G. P. (2018). Cip-ainet: An entropy-based immune network for multiple clustering. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. DOI: 10.1109/CEC.2018.8477744.

Borges, E., Ferrari, D. G., and de Castro, L. N. (2012). Silhouette-based clustering using an immune network. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–9. IEEE. DOI: 10.1109/CEC.2012.6252945.

Chao, G., Sun, S., and Bi, J. (2021). A Survey on Multiview Clustering. *IEEE Transactions on Artificial Intelligence*, 2(2):146–168. DOI: 10.1109/TAI.2021.3065894.

Chikhi, N. F. (2016). Multi-view clustering via spectral partitioning and local refinement. *Information Processing and Management*, 52(4):618–627. DOI: 10.1016/j.ipm.2015.12.007.

Cover, T. M. and Thomas, J. A. (2005). *Elements of Information Theory*. John Wiley & Sons, Inc., 1 edition. DOI: 10.1002/047174882X.

Cunha, D., Cruz, D., Politi, A., de Castro, L. N., and Maia, R. D. (2017). Bio-inspired multiobjective clustering optimization: A survey and a proposal. *Artificial Intelligence Research*, 6(2):10. DOI: 10.5430/air.v6n2p10.

Dang, X. H. and Bailey, J. (2010). Generation of alternative clusterings using the CAMI approach. *Proceedings of the 10th SIAM International Conference on Data Mining, SDM 2010*, pages 118–129. DOI: 10.1137/1.9781611972801.11.

de Castro, L. N. and Von Zuben, F. J. (2002). aiNet: An Artificial Immune Network for Data Analysis. In Abbass, H. A., Sarker, R., and Newton, C. S., editors, *Data Mining: A Heuristic Approach*, pages 231–260. IGI Global. DOI: 10.4018/978-1-930708-25-9.ch012.

de França, F. O., Coelho, G. P., Castro, P. A., and Von Zuben, F. J. (2010). Conceptual and Practical Aspects of the aiNet Family of Algorithms. *International Journal of Natural Computing Research*, 1(1):1–35. DOI: 10.4018/jncr.2010010101.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1):1–30. Available at:`https://www.jmlr.org/papers/volume7/demsar06a/demsar06a.pdf`.

Dua, D. and Graff, C. (2017). UCI machine learning repository. Available at: `http://archive.ics.uci.edu/ml`.

Ferraria, M. A., Ferraria, V. A., and de Castro, L. N. (2023). An Investigation Into Different Text Representations to Train an Artificial Immune Network for Clustering Texts. *International Journal of Interactive Multimedia and Artificial Intelligence*, 8(3):55. DOI: 10.9781/ijimai.2023.08.006.

Fu, L., Lin, P., Vasilakos, A. V., and Wang, S. (2020). An overview of recent multi-view clustering. *Neurocomputing*, 402:148–161. DOI: 10.1016/j.neucom.2020.02.104.

Geusebroek, J.-M., Burghouts, G. J., and Smeulders, A. W. M. (2005). The amsterdam library of object images. *Int. J. Comput. Vision*, 61(1):103–112. DOI: 10.1023/B:VISI.0000042993.50813.60.

Havrda, J. and Charvát, F. (1967). Quantification Method of Classification Processes. *Kybernetika*, 3(1):30–35. Available at:`https://dml.cz/bitstream/handle/10338.dmlcz/125526/Kybernetika_03-1967-1_3.pdf`.

Hossain, M. S., Ramakrishnan, N., Davidson, I., and Watson, L. T. (2013). How to ”alternatize” a clustering algorithm. *Data Mining and Knowledge Discovery*, 27(2):193–224. DOI: 10.1007/s10618-012-0288-4.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1):193–218. DOI: 10.1007/BF01908075.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651 – 666. DOI: 10.1016/j.patrec.2009.09.011.

Jenssen, R. (2010). Kernel entropy component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):847–860. DOI: 10.1109/TPAMI.2009.100.

Jerne, N. K. (1974). Towards a network theory of the immune system. *Annales d'immunologie*, 125C(1-2):373–89. Available at: `https://pubmed.ncbi.nlm.nih.gov/4142565/`.

Kontonasios, K.-N. and De Bie, T. (2015). Subjectively interesting alternative clusterings. *Machine Learning*, 98(1-2):31–56. DOI: 10.1007/s10994-013-5333-z.

Niu, D., Dy, J. G., and Jordan, M. I. (2014). Iterative discovery of multiple alternative clustering views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1340–1353. DOI: 10.1109/TPAMI.2013.180.

Orouskhani, M., Shi, D., and Orouskhani, Y. (2020). Multi-objective evolutionary clustering with complex networks. *Expert Systems with Applications*, 165:113916. DOI: 10.1016/j.eswa.2020.113916.

Principe, J. C. (2010). *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*. Springer New York, NY, 1 edition. DOI: 10.1007/978-1-4419-1570-2.

Rényi, A. (1961). On Measures of Entropy And Information. In *Proceedings of 4th Berkeley Symposium*, volume 1, page 457. Available at:`http://l.academicdirect.org/Horticulture/GAs/Refs/Renyi_1961.pdf`.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(C):53–65. DOI: 10.1016/0377-0427(87)90125-7.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(July 1928):379–423. DOI: 10.1145/584091.584093.

Silva, D. G., Fantinato, D. G., Canuto, J., Duarte, L. T., Neves, A., Suyama, R., Montalvão, J., and Attux, R. (2015). An Introduction to Information Theoretic Learning, Part I: Foundations. *Journal of Communication and Information Systems*, 31(April):68–79. DOI: 10.14209/jcis.2016.6.

Strehl, A. and Ghosh, J. (2002). Cluster ensembles - A knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3(3):583–617. Available at:`https://www.jmlr.org/papers/volume3/strehl02a/strehl02a.pdf`.

Vinh, N. X. and Epps, J. (2010). MinCEntropy: A novel information theoretic approach for the generation of alternative clusterings. *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 521–530. DOI: 10.1109/ICDM.2010.24.

Wan, X., Liu, X., Liu, J., Wang, S., Wen, Y., Liang, W., Zhu, E., Liu, Z., and Zhou, L. (2023). Auto-Weighted Multi-View Clustering for Large-Scale Data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8):10078–10086. DOI: 10.1609/aaai.v37i8.26201.

Wang, X., Wang, J., Domeniconi, C., Yu, G., Xiao, G., and Guo, M. (2019). Multiple independent subspace clusterings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5353–5360. DOI: 10.1609/aaai.v33i01.33015353.

Wu, C., Ioannidis, S., Sznaier, M., Li, X., Kaeli, D., and Dy, J. (2018). Iterative spectral method for alternative clustering. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 115–123. PMLR. Available at:`https://proceedings.mlr.press/v84/wu18a.html`.

Xie, X. L. and Beni, G. (1991). A Validity Measure for Fuzzy Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):841–847. DOI: 10.1109/34.85677.

Zhang, L., Cao, Q., and Lee, J. (2013). A novel ant-based clustering algorithm using Renyi entropy. *Applied Soft Computing Journal*. DOI: 10.1016/j.asoc.2012.11.022.