A Novel Adaptive Density-based ACO Algorithm with Minimal Encoding Redundancy for Clustering Problems

Esther Villar-Rodriguez*, Antonio Gonzalez-Pardo†, Javier Del Ser*,‡,
Miren Nekane Bilbao‡ and Sancho Salcedo-Sanz§
*TECNALIA. OPTIMA Unit, E-48160 Derio, Spain.
Email: {esther.villar, javier.delser}@tecnalia.com
†Basque Center for Applied Mathematics (BCAM), 48009, Bilbao, Spain.
Email: agonzalezp@bcamath.org
‡University of the Basque Country UPV EHU, 48013 Bilbao, Spain.
Email:{javier.delser, nekane.bilbao}@ehu.eus
§University of Alcalá, 28871 Alcalá de Henares, Madrid, Spain.
Email: sancho.salcedo@uah.es

***Abstract*—In the so-called Big Data paradigm descriptive analytics are widely conceived as techniques and models aimed at discovering knowledge within unlabeled datasets (e.g. patterns, similarities, etc) of utmost help for subsequent predictive and prescriptive methods. One of these techniques is clustering, which hinges on different multi-dimensional measures of similarity between unsupervised data instances so as to blindly collect them in groups of clusters. Among the myriad of clustering approaches reported in the literature this manuscript focuses on those relying on bio-inspired meta-heuristics, which have been lately shown to outperform traditional clustering schemes in terms of convergence, adaptability and parallelization. Specifically this work presents a new clustering approach based on the processing fundamentals of the Ant Colony Optimization (ACO) algorithm, i.e. stigmergy via pheromone trails and progressive construction of solutions through a graph. The novelty of the proposed scheme beyond previous research on ACO-based clustering lies on a significantly pruned graph that not only minimizes the representation redundancy of the problem at hand, but also allows for an embedded estimation of the number of clusters within the data. However, this approach imposes a modified ant behavior so as to account for the optimality of entire paths rather than that of single steps within the graph. Simulation results over conventional datasets will evince the promising performance of our approach and motivate further research aimed at its applicability to real scenarios.**

I. Introduction

In the last few years the generation of digital data has increased sharply in almost all knowledge and application domains as a result of the universal digitalization of their underlying systems and processes [1]. Unfortunately, our ability to infer hidden information therein has not evolved accordingly. Even though human limitations to understand highly dimensional flows of information have traditionally found an effective solution in advanced knowledge discovery models, the scales, heterogeneity and generational speed of data have coined the so-called *Big Data* concept. This global paradigm, which impacts on almost all disciplines (e.g. from Health to Telecommunications, Energy, Manufacturing, Social Sciences and Transport), aims at the research and development of technological tools and platforms to manage, store, retrieve, analyze and visualize data characterized by unprecedented scales in terms of their volume, variety, velocity or veracity [2]. In this work we narrow the scope on the analysis of data, around which the research community is witnessing a real upsurge of novel methods springing from Statistics, Mathematics, Physics and Artificial Intelligence.

Among those techniques that fall in the broad category of Artificial Intelligence, Machine Learning emerges as one of the most popular research areas. The goal of any given Machine Learning model is to extract patterns or similarity-based relationships from data. Depending on the implemented learning mechanism, the Machine Learning research field is divided into manifold subfields, such as *Supervised Learning*, *Unsupervised Learning* or *Reinforcement Learning* [3]. The first comprises the construction of predictive models based on labeled training data, i.e. every instance of the training data is tagged with its corresponding category or output, which is in turn the variable that the model must infer for any new unlabeled instance. Accordingly, *Unsupervised Learning* techniques are applied whenever no label for the training instances is available; in this case, the goal is to find patterns or similarities among the input instances based on a measure of similarity rather than on a variable to be predicted. To end with, *Reinforcement Learning* refers to those algorithms that learn from interactions with their environment by producing actions or events and receiving a positive or negative stimulus.

This paper elaborates on Unsupervised Learning, particularly on clustering as one of the most used family of models for discovering hidden structures or patterns within data. Generally speaking, a clustering problem aims at grouping unlabeled data instances in clusters considering a measure of similarity. The literature has been so far specially

profitable in terms of clustering models, with contributions focused on either the similarity metric or the learning process itself. As to mention, *distance*-based clustering algorithms minimize the distortion defined as the sum of the squared distances (e.g. Euclidean, Minkowski, Mahalanobis or any specifically devised ad-hoc metric) between each observation and its designated centroid. By contrast, *density*-based algorithms find clusters within data based on a notion of density-based compactness among neighboring instances.

Regarding the clustering procedure an upsurge of research works have gravitated on the use of Computational Intelligence, which have been shown to efficiently overcome different shortcomings of conventional methods, such as a strong dependence on initialization, a slow convergence speed and/or the lack of optimality of the produced solutions. In this context a number of clustering schemes have relied on different population-based meta-heuristic solvers from Evolutionary Computation, such as Genetic Algorithms (GA, [4], [5]), Harmony Search (HS, [6], [7]) or Estimation of Distribution Algorithms (EDA, [8], [9]), among others. Another subset of clustering heuristics hinges on Swarm Intelligence, which is inspired by the behavior of social organisms and their interactions to achieve a global form of collective intelligence. Examples of Swarm Intelligence algorithms applied to clustering problems abound, from Particle Swarm Optimization (PSO, [10]), or Ant Colony Optimization (ACO, [11]), the latter grasping the algorithmic focus of this paper. Specifically, the present work outlines the design of a novel density-based ACO clustering scheme that resorts to a tailored encoding of the solution space (i.e. the graph explored by the ant colony) that embeds the number of clusters underlying the dataset under analysis. This is a radically new approach to the partitioning of unsupervised datasets that outperforms previous schemes in the literature, as argued in the following survey and subsequently proven by simulations over illustrative datasets.

II. ACO Algorithms for Clustering Problems

ACO algorithms are based on the foraging behavior of ants, which walk through the environment arbitrarily at random, initially guided by their sole intuition. Once the any given ant reaches the destination (and then, the completion of a solution), it retraces up to the starting point placing a certain quantity of pheromones (proportional to the quality of the achieved solution) along the path that will guide subsequent ants in their foraging process. In order to avoid that non-optimal solutions attract the majority of ants, pheromones are forced to decrease according to a evaporation rate ρ.

Different application domains have leveraged this meta-heuristic algorithm when undertaking optimization problems modeled by graphs, such as scheduling, planning and routing. Indeed supervised and unsupervised learning lie among those scenarios where ACO approaches have been put to practice: for instance, the authors in [12] proposed a rule-based ACO algorithm for classification tasks; in [13] a neural network prediction model was trained by means of an ACO scheme; and naïve Bayes classifiers have been recently hybridized with ACO algorithms in [14]. When it comes to clustering the most popular ACO approach is the one in [11] (coined as ACOC), where the optimization problem is formulated as to find the mapping from instances $\{X_i\}_{i=1}^{m}$ to cluster centers $\{C_k\}_{k=1}^{g}$ (represented as $w_{i,k} \in \{0,1\}$ such that $w_{i,k} = 1$ if instance i belongs to cluster j) that:

$$\{\{w_{i,j}^{*}\}_{i=1}^{m}\}_{j=1}^{g} \doteq \underset{w_{i,j}\in\{0,1\}}{\arg\min} \sum_{i=1}^{m}\sum_{j=1}^{g} w_{i,j} \parallel X_i - C_j \parallel, \quad (1)$$

subject to $\sum_{j=1}^{g} w_{i,j} = 1 \ \forall i \in \{1,\ldots,m\}$ and $\sum_{j=1}^{g} w_{i,j} \geq 1 \ \forall j \in \{1,\ldots,g\}$. In the above expression, $\parallel \cdot \parallel$ denotes the Frobenius norm. From a practical perspective the aforementioned ACOC approach maps the problem onto a decision graph $G \doteq (\mathcal{V}, \mathcal{E})$, over which the standard ACO heuristic is executed. The set of nodes of the graph $\mathcal{V}$ contains all possible pairs $\langle m, g \rangle$. When visited by any given ant, node $N(i,j)$ in the graph G establishes that instance i is assigned to cluster j.

When analyzed in depth, several drawbacks of the ACOC are worth being mentioned: to begin with, ACOC requires the a priori definition of the number of clusters g to be found within the data so as to arrange the $m \times g$ decision graph over which ants are deployed. Since every node in the graph represents by itself the mapping from instances to clusters, the exploratory behavior of ants simply reduces to a probabilistically driven movement from $N(i,j)$ to any node within the set $\{N(i+1,j)\}_{j=1}^{g}$. This movement is influenced by both past experience and a personal judgment of the ant symbolized by a heuristic value measuring the expected efficiency of such a choice. For the ACOC approach, the probability to move along the path $(n1, n2)$ between nodes $n1 \doteq N(i,j)$ and $n2 \doteq N(i+1,j')$ for ant k and step t will be given by

$$p_{n1,n2}^{k,t} = \frac{(\tau_{n1,n2}(t))^{\alpha} \cdot (\eta_{n1,n2}(t))^{\beta}}{\sum_{n=1}^{g} (\tau_{n1,n}(t))^{\alpha} \cdot (\eta_{n1,n}(t))^{\beta}}, \quad (2)$$

where $\tau_{n1,n2}(t)$ denotes the quantity of pheromones deposited in the path $(n1, n2)$ at iteration t; $\eta_{n1,n2}(t)$ defines the heuristic value of the problem; and α and β are two exponential parameters that balance between the influence of the pheromones and the heuristic function. This stepwise movement policy for ants is indeed simple to implement; the clustering solution (i.e. the path followed by ants) can be decomposed in locally decided movements without any loose of optimality. However, as mentioned before this imposes that g (i.e. the number of clusters) must be known beforehand, which can be a time consuming task (especially if undertaken with greedy schemes or exhaustive search procedures). Furthermore, in most cases there is no gold standard that unveils the cardinality of the cluster space that the algorithm should infer, fact that jeopardizes the assessment of the clustering arrangement produced by the algorithm.

Another drawback can be found underneath the decision graph proposed by the ACOC model: two different ants

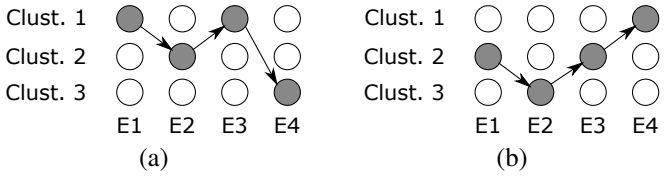

Fig. 1. Two different solutions of the ACOC algorithm for a dataset composed by 4 elements that must be gathered in 3 clusters. Clustering arrangements represented by both paths are equal to each other.

may represent the same structural arrangement in the cluster space even though traversing different paths in the decision graph of the algorithm. As shown in Figure 1, this noted fact is due to the encoding redundancy of the decision graph: both solutions plotted in this figure are absolutely equivalent representations of the same 3-cluster arrangement: (1,3), (2) and (4). This redundancy must be avoided for the sake of a good convergence of the algorithm: otherwise ants may become *confused* when traversing the graph by finding two distant paths with similar fitness. In this context our work adopts the so-called Linear Linkage Encoding (LLE, [15]) representation that removes the encoding redundancy by representing each cluster arrangement as a list of numerical pointers to the very next item belonging to the same cluster in numerical order. Furthermore, this encoding strategy allows for the blind estimation of the number of clusters within the data, i.e. in their exploration through the graph ants do not only allocate instances to clusters but also estimate the cardinality of the cluster arrangement. Unfortunately this comes along with more complex behavioral operators for the pheromone deposit and the ants' movement, as described in what follows.

III. Description of the Proposed ACO Algorithm

As explained in the previous section, the clustering algorithm proposed in this work hinges on a specially designed solution graph over which ants are deployed to seek a solution. This graph results from the adoption of the so-called Linear Linkage Encoding (LLE) approach proposed in [15] to numerically represent clustering solutions. Similarly to conventional number encoding schemes, in LLE each clustering solution is represented by a m-sized list of integers, with m denoting the total amount of instances in the dataset. In traditional number encoding the allocation of an instance i to a cluster j is represented by the numerical index of the given cluster (i.e. j) located at the i-th position of the solution vector, which implicitly implies setting the number of considered clusters beforehand. However, in LLE each entry of the clustering vector is a link to its subsequent neighbor within the same cluster; consequently each group is identified by its linked member instances rather than by a cluster identifier nominatively assigned to each instance. Following this notation the clustering solution exemplified in Figures 1.a and 1.b would be uniquely encoded as $(3, 2, 3, 4)$, whereas their number encoding representation would be $(1, 2, 1, 3)$ (Figure 1.a) and $(2, 3, 2, 1)$ (Figure 1.b), phenotypically different yet genotypically equivalent.

Generally speaking, a LLE representation must satisfy the following requirements:

1) The integer value stored in each position of the solution is greater than or equal to its index but less than or equal to the number of instances m.
2) Two different indexes within a given solution can not hold the same value, unless the value is equal to any of both indexes (which stands for the closure of the cluster).

When adopting the above set of rules to the proposed algorithm, its search complexity is alleviated by virtue of a significantly pruned solution graph, which constitutes a non-redundant phenotype of the clustering problem and potentially makes the deployed colony of ants converge faster to global solutions. Such a graph is initially based on a $m \times m$ grid of states. If we denote as $\mathcal{N}_i^k(t)$ the set of reachable states by ant k from position $i \in \{1, \ldots, m\}$ at iteration t, and $j_z^k(t)$ is the past state visited by this ant at previous step $z \in \{1, \ldots, i-1\}$ within the same iteration, it can be shown that the above rule set gives rise to a dynamically[1] pruned grid of states defined by

$$\mathcal{N}_i^k(t) \doteq \left\{ j : i \leq j \leq m,\ j \notin \bigcup_{z=1}^{i-1} j_z^k(t) \cdot \mathbb{I}\left(j_z^k(t) \neq i\right) \right\}, \quad (3)$$

where $\bigcup$ denotes union of elements, and $\mathbb{I}(\cdot)$ is an auxiliary indicator function taking value 1 if the argument is true (0 otherwise). At this point it is also important to note that the dynamically pruned solution graph embeds the estimation of the number of cluster, as its initial layout is $m \times m$ disregarding the specific dataset under analysis. Figures 2.a and 2.b illustrates this pruning procedure.

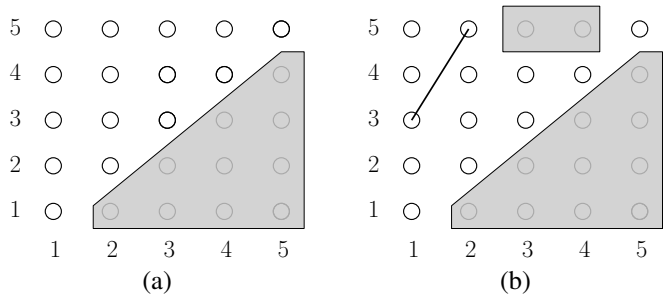

Fig. 2. (a) Initially pruned solution graph of the ACO due to the application of the LLE encoding rules to a clustering problem with $m = 5$ data instances; (b) dynamic pruning due to the partial path $(3, 5)$ traversed by an ant. At $i = 3$ the ant is allowed to visit nodes $(4, 3)$ or $(4, 4)$. Gray-shaded regions of the graph are unreachable.

A. Modified ACO Operators

The pruning of the decision graph explained above does not modify significantly the behavior of ants with respect to its original definition in the traditional ACO algorithm: the movement of ants are still driven probabilistically as a function of the heuristic of the problem and the experience of deployed ants in previous iterations. As such, an ant placed in a given

[1]Here, *dynamically* refers to the fact that the set of pruned states at position $i + 1$ depends on the visited states by the ant at $\{1, \ldots, i\}$.

state at step i selects any of a list of possible destination states at step $i+1$. Then it computes the heuristic value in charge of measuring the quality or merit of moving from the current state to any of the possible next states, and retrieves the pheromone information from the graph in order to infer knowledge in terms of the quality of past solutions. With this information the ant at hand selects the state to visit at step $i+1$ using the heuristic and pheromone information by using probabilities computed as in Expression (2).

B. Modified Pheromone Computation

Despite its high-level similarity with naïve ACO solvers, there are strong underlying differences in regards to the computation of the heuristic value and the retrieval of pheromones that underpin the novelty of the proposed ACO model. To begin with, in general pheromones contain localized path-wise information about the history of past executions. The use of LLE for representing the clustering solutions (and the subsequent graph pruning) imposes that a pheromone in our approach is not related to how proper an isolated movement from step i to $i+1$ proves to be, but rather indicates how efficient a partial clustering solution is by considering the knowledge inferred by ants until step i. This particularity changes the way pheromones are computed in our approach, for which the algorithm incorporates a *memory* or pool of P entries, each storing a path traversed through the graph by the deployed pheromones and its corresponding fitness value. If ant k is assessing whether to move from node $n1 = N(i,j)$ to node $n2 = N(i+1,j')$ (with $j' \in \mathcal{N}_i^k(t)$), the corresponding pheromone value $\eta_{n1,n2}^k(t)$ required for computing the heuristic value of this transition at iteration t should be determined by 1) the fitness of those previous ants stored in the pool that contain this transition in their path; and 2) the similarity between the partial cluster arrangements of ant k with respect to those previous ants that traversed the same transition from node $n1$ to $n2$. In other words,

$$\eta_{n1,n2}^k(t) \propto \sum_{\substack{p=1 \\ j_i^{p,\flat}(t\text{-}1)=j \\ j_{i+1}^{p,\flat}(t\text{-}1)=j'}}^{P} \theta(\boldsymbol{J}_i^k(t-1), \boldsymbol{J}_i^{p,\flat}(t-1)) \cdot f\left(\boldsymbol{J}_m^{p,\flat}(t)\right),$$

where $\boldsymbol{J}_i^k(t) \doteq \{j_z^k(t)\}_{z=1}^i$ and $\boldsymbol{J}_i^{p,\flat}(t) \doteq \{j_z^{p,\flat}(t)\}_{z=1}^i$ stand for the partial paths traversed by ant k and pooled ant p until position i at iteration t, respectively. In the above expression $\theta(\boldsymbol{J}, \boldsymbol{J}')$ is a measure of the similarity between two data clusterings represented by $\boldsymbol{J}$ and $\boldsymbol{J}'$, which can be implemented in practice by resorting to any of the distance metrics published in the literature (e.g. Adjusted Rand Index or Normalized Mutual Information [16]). Finally, $f(\boldsymbol{J})$ is the fitness of the clustering arrangement represented by $\boldsymbol{J}$. the design of the pheromone expression as stated above accommodates any of the plethora of structural clustering metrics reported to date, from the well-known Silhouette score [17] to more elaborated counterparts such as the Davies-Bouldin [18] or the Dunn index [19]. In the preliminary results discussed later in this article the so-called Calinski-Harabasz metric [20] has been selected for its sensitiveness to small-sized clusters.

This type of pheromones and the number of different combinations between pair of elements in any dataset may result in a combinatorial explosion in the number of pheromones. In order to solve this problem, we have introduced an *Oblivion Rate* heuristic [21] in charge of controlling the number of pheromones created in the system. Specifically we have implemented this pheromone control as a reinforcement strategy in the aforementioned pool of past solutions. Given a new solution, if it has been already discovered (i.e. the solution exists in the pool) the fitness associated to this solution is increased by a factor α. The size of the pool is kept fixed to P by removing solutions depending on their *obsolescence* (i.e. the probability that a given solution is deleted from the pool increases as the number of ants having followed this path decreases).

C. Density-based Heuristic Value

The probability driving the movement of ants along the solution graph is also controlled by the heuristic value $\tau_{n1,n2}^k(t)$, which should help ants converge towards solutions maximizing the aforementioned clustering metric jointly with the learning capability enabled by the deposit of pheromones. We have opted for making isolated decisions related to the state of the current node and the pertinent cluster by using a density-based heuristic similar to the one in the DBSCAN algorithm [22]. This metric grounds on a parameter ϵ that quantifies the maximum normalized distance $d(i,i')$ between nodes i and i' belonging to the same cluster; the lower ϵ is, the more compact the clusters embedded in the solution of the algorithm will be. This heuristic value will be given by

$$\tau_{n1,n2}^k(t) = \frac{d(i,i+1) \cdot \mathbb{I}(d(i,i+1) \leq \epsilon)}{\sum\limits_{i' \in \mathcal{N}_i^{k,\bowtie}(t)} d(i,i') \cdot \mathbb{I}(d(i,i') \leq \epsilon)}, \qquad (4)$$

where $\mathcal{N}_i^{k,\bowtie}(t)$ denotes the search space of ant k at step i and iteration t. This search space will be given by the union of 1) the set of reachable states due to the LLE approach utilized for encoding the problem space defined in Expression (3); and 2) those past instances laying at a distance less than ϵ. The need for exploring past instances finds its rationale in the incremental assignment performed by ants along their paths, which clashes with the linkage between samples imposed by the encoding strategy. By also considering past yet sufficiently close data instances in the search space of the heuristic the algorithm is able to merge new data instances into existing clusters disregarding whether they are closed by the LLE representation. This extended search space is expressed as

$$\mathcal{N}_i^{k,\bowtie}(t) \doteq \mathcal{N}_i^k(t) \cup \left\{ \bigcup_{c \in \mathcal{C}_i^k(t)} \arg\min_{l \in c} d(l,i) \right\}, \qquad (5)$$

with $\mathcal{C}_i^k(t)$ denoting the set of already existing clusters solved by ant k at step i and iteration t. That is, the right-hand union

of sets in the above expression and the indicator functions in Expression (4) follow the core principle of density-based clustering by which a cluster is reachable by point i if any instance within the cluster is ϵ-reachable from i. Also note that this cluster reconsideration comes along with a side repair procedure of the solution vector so as not to violate the encoding rules of the LLE strategy. In addition, the selection of the ϵ value is independent of the dataset used thanks to the normalization of $d(i, i')$ with respect to the maximum distance within the dataset which, as a matter of fact, can be set to any arbitrary multidimensional measure of distance.

Summarizing, the proposed ACO algorithm features several advantages with respect to the state of the art:

1) It does not depend on the specification of the number of clusters to be found in the dataset.
2) The joint adoption of a density-based heuristic and a Calinski-Harabasz measure of structural clustering fitness allows identifying outliers within the dataset, as opposed to recent ACO-based clustering alternatives reported in the literature; and
3) The algorithm is flexible enough to arbitrarily tune its parameters so as to meet clustering paradigms of very diverse nature: the fitness function $f(\cdot)$, the metric of similarity $\theta(\boldsymbol{J}, \boldsymbol{J'})$ among partial clusters or the heuristic $\tau^k_{n1,n2}(t)$ itself. Interestingly the latter unveils a sought independence between the heuristic approach used for building the clustering solution and the metric adopted for evaluating the structural quality of the complete solution built by the deployed ants.

IV. Results and Discussion

In order to shed light on the performance of the proposed ACO-based clustering approach, several computer experiments have been carried out over datasets with simple yet diverse structural characteristics. This preliminary set of experiments aim at answering the following research questions:

Q1 Is the proposed algorithm able to identify the correct number of clusters without providing any a priori information about this parameter?
Q2 How does the algorithm perform when the dataset contains outliers? (i.e. points or small clusters with structural looseness or low connectivity to other well-defined clusters of higher cardinality)
Q3 How does the proposed clustering technique perform with respect to the popular ACOC approach proposed in [11]?

To answer these questions from an empirical approach three different datasets have been selected: the first one is a synthetic dataset composed by 9 instances, each with 2 features or characteristics. A simple visual inspection evinces that these elements are easily grouped in 4 different clusters, one of them composed by a single element (i.e. an outlier). This dataset will be useful to understand whether the combination of the density-based heuristic and the Calinski-Harabasz measure excels at identifying not only the clear underlying cluster structure, but also on isolating the outlier within the dataset.

The other two selected datasets are well-known in the literature and have been extensively utilized for preliminarily validating new clustering schemes. One of them is the Iris flower dataset [23], which comprises 150 instances with 4 features and 50 samples. Such elements are drawn from three different classes or categories, each representing a type of Iris plant: *Iris-setosa*, *Iris-versicolor* and *Iris-virginica*. Features correspond to the sepal length and width (in cm) and the petal length and width (in cm). Each class contains 50 different elements within the dataset. Interestingly, one class (*Iris-setosa*) is linearly separable from the other 2, whereas the latter are not linearly separable from each other. Therefore it is easy to differentiate between *Iris-setosa* and the other two classes, but it is extremely difficult to separate the elements belonging to *Iris-versicolor* and *Iris-virginica* without any prior knowledge about the number of clusters. As for clustering purposes, it is unrealistic (and controversial as discussed in related studies) to make any proposed algorithm discriminate among 3 clusters within this dataset; this parameter cannot be easily inferred a priori, thus it should be the structural measure of quality and the utilized heuristic what should drive the algorithm on this purpose.

Finally, the third dataset is a classical problem in spectral clustering: the *Moons* dataset, composed by 400 bi-dimensional elements that belongs to 2 different clusters with structural connectivity (i.e. their shape is a half circle). The position of the different elements of this dataset jeopardizes the correct identification of the clusters if the clustering algorithm uses a centroid-based metric of structural fitness. For solving all these three datasets, we have executed the classical ACOC algorithm [11] and the proposed algorithm described in Section III using the same configuration for the sake of fairness in the comparison of their results. This configuration is shown in Table I. For the proposed algorithm, the parameters ϵ and P – i.e. the maximum normalized distance $d(i, i')$ between nodes i and i' belonging to the same cluster and the size of the pool – have been fixed to 0.2 and 5 in all the considered experiments.

Parameter	Value
Colony	5
Iterations	10
α	1
β	2
Evaporation rate ρ	0.1
Repetitions	10

TABLE I
CONFIGURATION OF THE ACOC ALGORITHM AND THE PROPOSED ACO FOR CLUSTERING THE DIFFERENT DATASETS.

Figure 3 depicts the best solutions obtained for the different datasets: the first column (Figures 3.a, 3.c and 3.e) is comprised by plots of the best solutions achieved by the proposed algorithm, whereas in the second column (correspondingly, 3.b, 3.d and 3.f) the best solutions found by the ACOC algorithm are shown. Instances belonging to the same cluster are marked with the same symbol. In the case of the Iris flower dataset dimensionality has been reduced

down to 2 components via Principal Component Analysis (PCA). To begin with, Figure 3.a shows that the proposed algorithm finds correctly the number of different clusters that compose the dataset. In this case, our approach provides the same solution than the ACOC algorithm (see Figure 3.b), the difference being that no prior knowledge about the number of clusters has been assumed for the former. Another aspect to highlight is that the clustering algorithm proposed in this paper is sensible to outliers: in this first dataset the outlying sample is perfectly identified as an isolated cluster. This capability is critical for any clustering algorithm because treating outliers as an element inside any other cluster could degrade the solution and mask relevant structural information for the application at hand (as in e.g. the unsupervised detection of failures in predictive maintenance of industrial machinery).

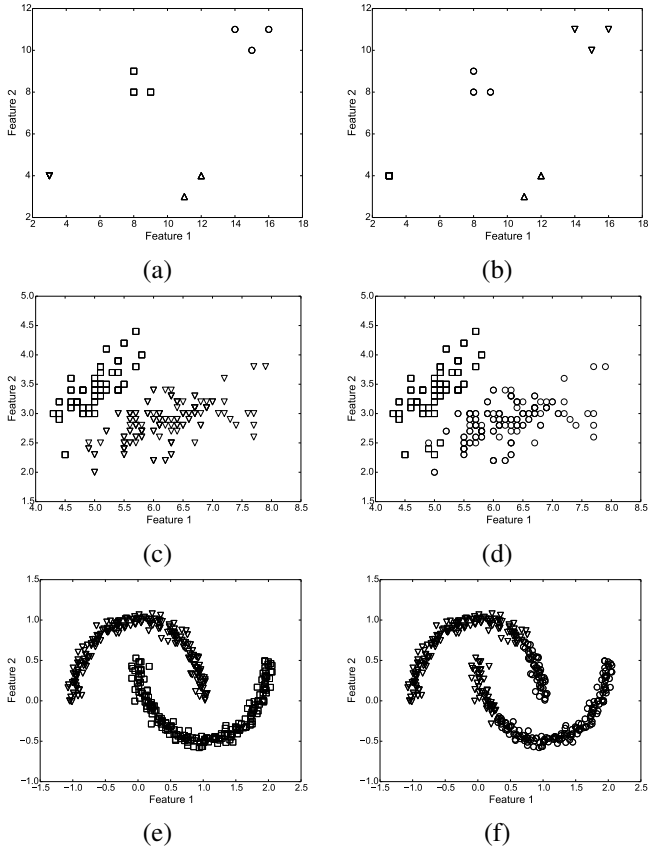

Fig. 3. Best solution found by the proposed algorithm (first column) and the well-known ACOC algorithm (second column) for the synthetic dataset (first row), the Iris flower dataset (second row) and the Moons dataset (third row).

Regarding the Iris flower dataset, the proposed algorithm is able to identify the *Iris-setosa* class, whereas the elements belonging to *Iris-versicolor* and *Iris-virginica* are grouped in the same cluster (Figure 3.c). This arrangement, however, must not be conceived as a bad solution for its mismatch with respect to the ground of truth because, as it was previously stated, elements belonging to *Iris-versicolor* and *Iris-virginica* are not linearly separable without any prior knowledge about the number of classes. In the case of ACOC, if we set the number of classes to 2 (i.e. $g = 2$) the best result is similar to the one by the proposed algorithm (Figure 3.d), with small differences in the frontier between the discerned classes.

The discussion follows by analyzing the performance for the Moons dataset shown in Figures 3.e and 3.f: the proposed clustering scheme groups together all elements belonging to the same semi-circled cluster by virtue of its density-based heuristic. By contrast, the ACOC algorithm fails to discover this clustering structure within the dataset as a consequence of its centroid-based search procedure (similar to the well-known K-Means algorithm). Furthermore, once again our algorithm is able to correctly map elements to clusters without providing any information about the number of clusters. This result reveals that the density-based heuristic and the Calinski-Harabasz measure are good options for constructing ACO-based approaches suitable to deal with clustering paradigms of diverse structural characteristics.

V. Conclusion and Future Research Lines

Theoretically clusters should correspond to data patterns that reflect a latent yet not evident structure of the data at hand. Critical to the task of inferring such patterns is to determine the number of clusters g to be discovered, which is often approached via a multidimensional similarity metric that is meant to be optimized so as to reduce the intra-cluster distortion. Both the value of g and the similarity metric depend on the dataset at hand and the ultimate purpose of the clustering problem, and their proper selection is decisive to achieve optimality. Unfortunately, in most cases g may not be easily estimable; in fact the estimation of g is deemed one of the most challenging paradigms in data science, which usually ends up by resorting to computationally-expensive enumerative strategies and a priori knowledge. Consequently, partitioning algorithms such as K-Means, K-medoids or the one rooted on Expectation-Maximization schemes undergo this noted shortcoming and hence prune their search space by a previously-specified albeit potentially unrealistic value of g.

In this context this work has elaborated on the design of a novel Ant Colony Optimization algorithm capable of dodging two of the most important downsides of its major predecessors: the assumption that the discovered clusters are spherically shaped, and the a priori knowledge of the number of clusters to be found. In order to overcome these disadvantages, our proposed algorithm adopts a density-based heuristic method able of discovering clusters of arbitrary structures and shapes by resorting to connectivity and density notions. Furthermore, the decision graph representing the solution space of the clustering problem is pruned as a result of the adoption of Linear Linkage Encoding, which permits denoting the neighboring relationships between data instances and the number of resulting groups or clusters in a single, minimally redundant numerical representation. The devised ACO-based clustering method leverages the above density-based heuristic, the LLE encoding approach and a modified pheromone update method that hinges on a metric of partial path similarity between paths through the graph, and an arbitrary measure of structural quality of the clustering arrangement inferred by

past ants. The flexibility in terms of the utilized structural clustering allows directing the convergence of ants towards cluster arrangements with different structural sensitivity levels (e.g. for detecting outliers).

In the experimental phase 3 different examples have been analyzed and discussed, aimed at evincing the performance of the proposed algorithm when facing dataset with diverse structural shape. The first one is a synthetic dataset that contains outliers, which has been helpful to clarify the sensitivity of the algorithm to infer clusters at different resolutions. Results in the remaining datasets (i.e. Iris and Moons) have been insightful to compare the performance of the proposed method to that of the ACOC in [11]: the proposed algorithm is able to identify correctly the number of clusters in the dataset as told by the utilized metric of structural clustering quality, no matter whether the dataset is characterized by a partitional (Iris) or spectral (Moons) separability. This good performance across heterogeneous datasets is promising and motivates future efforts towards extending the applicability of the proposed algorithm to other clustering paradigms (e.g. on-line clustering).

ACKNOWLEDGEMENTS

This work has been supported by the Spanish Ministry of Science and Education (project ref. TIN2014-56494-C4-4-P), by the Comunidad Autonoma de Madrid under project CIBERDINE S2013/ICE-3095, Savier an Airbus Defense & Space project (FUAM-076914 and FUAM-076915), by the Basque Government through the BERC 2014-2017 program and the BID3A project, and by Spanish Ministry of Economy and Competitiveness MINECO: BCAM Severo Ochoa excellence accreditation SEV-2013-0323.

REFERENCES

[1] A. Fernández, S. del Río, V. López, A. Bawakid, M. del Jesus, J. Benítez, F. Herrera, "Big Data with Cloud Computing: an Insight on the Computing Environment, Mapreduce, and Programming Frameworks," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Vol. 4, N. 5, pp. 380–409, 2014.

[2] D. Laney, "3D Data Management: Controlling Data Volume, Velocity and Variety," *META Group Research Note*, Vol. 6, p. 70, 2001.

[3] O. Bousquet, U. von Luxburg, G. Rätsch, *Advanced Lectures on Machine Learning*, Lecture Notes in Computer Science, N. 3176, 2004.

[4] L. Coletta, E. Hruschka, A. Acharya, J. Ghosh, "A Differential Evolution Algorithm to Optimise the Combination of Classifier and Cluster Ensembles," *International Journal of Bio-Inspired Computation*, Vol. 7, N 2, pp. 111–124, 2015.

[5] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, "A Survey of Multiobjective Evolutionary Clustering," *ACM Computing Surveys*, Vol. 47, N. 4, pp. 1–46, 2015.

[6] A. George, G. Gopakumar, M. Pradhan, K. Abdul Nazeer, M. Palakal, "A Self Organizing Map - Harmony Search Hybrid Algorithm for Clustering Biological Data," *IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems* (SPICES), pp. 1–5, 2015.

[7] R. Forsati, M. Meybodi, M. Mahdavi, A. Neiat, "Hybridization of K-means and Harmony Search Methods for Web Page Clustering," *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Vol. 01, pp. 329–335, 2008.

[8] H. Mühlenbein, G. Paass, "From Recombination of Genes to the Estimation of Distributions I. Binary Parameters," *Parallel Problem Solving from Nature – PPSN IV*, Lecture Notes in Computer Science, vol. 1141, pp. 178–187, 1996.

[9] C. Echegoyen, A. Mendiburu, R. Santana, J. Lozano, "Toward Understanding EDAs based on Bayesian Networks through a Quantitative Analysis," *IEEE Transactions on Evolutionary Computation*, Vol. 16, N. 2, pp. 173–189, 2012.

[10] S. Alam, G. Dobbie, Y. Koh, P. Riddle, S. Ur Rehman, "Research on Particle Swarm Optimization based Clustering: A Systematic Review of Literature and Techniques," *Swarm and Evolutionary Computation*, Vol. 17, pp. 1–13, 2014.

[11] Y. Kao, K. Cheng, "An ACO-based Clustering Algorithm," *Ant Colony Optimization and Swarm Intelligence*, Lecture Notes in Computer Science, Vol. 4150, pp. 340–347, 2006.

[12] F. Otero, A. Freitas, C. Johnson, "Cant-Miner: An Ant Colony Classification Algorithm to Cope with Continuous Attributes," *Ant Colony Optimization and Swarm Intelligence*, Lecture Notes in Computer Science, Vol. 5217, pp. 48–59, 2008.

[13] C. Blum, K. Socha, "Training Feed-Forward Neural Networks with Ant Colony Optimization: an Application to Pattern Classification," *International Conference on Hybrid Intelligent Systems*, p. 6, 2005.

[14] M. Borrotti, I. Poli, "Naïve Bayes Ant Colony Optimization for Experimental Design," *Synergies of Soft Computing and Statistics for Intelligent Data Analysis*, Advances in Intelligent Systems and Computing, Vol. 190, pp. 489–497, 2013.

[15] O. Ülker, E. Özcan, E. Korkmaz, "Linear Linkage Encoding in Grouping Problems: Applications on Graph Coloring and Timetabling," *International Conference on Practice and Theory of Automated Timetabling VI*, pp. 347–363, 2007.

[16] B. Duran, P. Odell, *Cluster Analysis: a Survey*, Springer Science & Business Media, vol. 100, 2013.

[17] P. Rousseeuw, "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis," *Journal of Computational and Applied Mathematics*, Vol. 20, pp. 53–65, 1987.

[18] D. Davies, D. Bouldin, "A Cluster Separation Measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-1, N. 2, pp. 224–227, 1979.

[19] J. Dunn, "A Fuzzy Relative of the Isodata Process and its Use in Detecting Compact Well-Separated Clusters," *Journal of Cybernetics*, Vol. 3, N. 3, pp. 32–57, 1973.

[20] T. Calinski, J. Harabasz, "A Dendrite Method for Cluster Analysis," *Communications in Statistics*, Vol. 3, N. 1, pp. 1–27, 1974.

[21] A. Gonzalez-Pardo and D. Camacho, "A New CSP Graph-based Representation to Resource-Constrained Project Scheduling Problem," *IEEE Conference on Evolutionary Computation*, pp. 344 – 351, 2014.

[22] M. Ester, H. Kriegel, J. Sander, X. Xu, "A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, 1996.

[23] K. Bache and M. Lichman. "UCI machine learning repository", 2013