This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3279723

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Parameters Sensitivity Analysis of Ant Colony based Clustering: Application for Student Grouping in Collaborative Learning Environment

ABIR ABID[1], (GS Member, IEEE), ILHEM KALLEL[1,2], JAVIER J. SANCHEZ-MEDINA[3], AND MOUNIR BEN AYED[1,4] (Senior Members, IEEE)

[1]ReGIM-Lab: Research Groups in Intelligent Machines, University of Sfax, ENIS, BP 1173, 3038, Sfax, Tunisia (e-mail: abir.aid@ieee.org)
[2]ISIMS: Higher Institute of Computer Science and Multimedia of Sfax, University of Sfax, Route de Tunis Km 10 B.P. 242, 3021, Sfax, Tunisia (e-mail: ilhem.kallel@ieee.org)
[3]CICEI, IUCES, University of Las Palmas de Gran Canaria. 35017 Las Palmas de Gran Canaria, Spain (email: javier.sanchez@ulpgc.es)
[4]Faculty of Sciences of Sfax, Computer science and Communications Department, University of Sfax, Route Soukra Km 3.5 BP 1171, 3000, Sfax, Tunisia (e-mail: mounir.benayed@ieee.org)

Corresponding author: Abir Abid (e-mail: abir.abid@ieee.org).

This work was supported by the university of Sfax through the alternate scholarship granted to spend a 3-month internship at the University of Las Palmas de Gran Canaria and by grants from Tunisian General Direction of Scientific Research (DGRST)

ABSTRACT Clustering analysis is one of the data analysis techniques that organizes items into clusters according to their degrees of similarities. In this context, bio-inspired algorithms have found success in solving clustering problems.Inspired by nature, Ant Colony based Clustering arises from ant colony behavior in organizing nests and clustering ants corpses. Accordingly, several researchers proposed different clustering algorithms that mimic the real ants behavior in forming cemeteries. However, the performance of a given algorithm depends strongly on its parameters settings. Indeed, it holds a large number of adjustable parameters that need to be instantiated by suitable values. In this paper, we study the parameters influence, more precisely the parameter α which is responsible for adjusting similarity between objects. In fact, we analyze the impact of α values on the performance of some well known Ant Colony based Clustering Algorithms applied to constructing team-works in a collaborative learning environment. After various bench tests, the choice of α value is determined based on the best algorithm accuracy for each learning data-set. The experimental results prove that Ant Colony algorithms performance strongly depends on α, especially when applied to high-dimensional data-sets. However, α has a negligible influence on the algorithm's accuracy when applied to low-dimensional data-sets. Obviously, the feature selection step could be ignored since it has a negligible influence on the algorithm performance even with different values of α.

INDEX TERMS Ant Colony Algorithms Parameters, Ant Colony Clustering (ACC), Collaborative Learning Environment, Alpha parameter. Alpha similarity, Parameters analysis, ACC Parameters sensitivity Ant colony optimization (ACO)

I. INTRODUCTION

THE continuous process of digitization with the growing technological development has generated an enormous bulk of data. In this context, clustering analysis is a methodological area significantly contributing to many domains applications. It intends to identify representative clusters to explore meaningful conclusions from the original data. In fact, clustering is a kind of classification that aims at discovering groups of items in a way that similar ones are gathered in the same cluster while the dissimilar ones are grouped in separate clusters [1].

Nature-inspired meta-heuristic algorithms have been developed on the principles and inspiration of the biological evolution of Nature. They are well known in machine learning to address optimal solutions of complex problems. Nowadays, there exist more than 150 bio-inspired algorithms in the literature [2] such as Genetic Algorithms (GA) [3], Particle Swarm Optimization (PSO) [4] [5], Ant Colony Op-

timization (ACO) [6], Bee Colony Optimization (BCO) [7], etc. These algorithms have been adapted and found success in solving clustering and NP complete problems. However, these algorithms involve different parameters which impact their performance. Parameters' tuning (or setting) has attracted several researchers in the past decade [8] as even though the algorithm is efficient, setting inappropriate values of parameters may lead to a low quality solution. As an example, in [9] authors stated that integration among parameters such as mutation, crossover rates and population is vital for successful GA search. Similarly, the searching capability of PSO algorithm is directly influenced by its three main control parameters (inertia weight, cognitive acceleration coefficient and social acceleration coefficient). In fact, as defined in [10], [11] and [12], a priory tuning of the PSO control parameters may lead to improving its performance to be rather sensitive to these parameters. Likewise, inspired by larval sorting activities and corpses clustering observed in real ant colonies, the ant based clustering has emerged as a new method for solving clustering problems. Just like other bio-inspired algorithms, ant colony algorithms require an initial setting of parameters before starting. Like other researchers inspired by this issue, we hereby, in this paper, present a comparative study of ant colony clustering parameters effect on small and large educational data-sets.
The remaining parts of this article are organized as follows: Section II looks over the influence of parameters on ants behavior based algorithm. Then, a review on ant colony algorithms for data clustering is conducted in Section III. Next, sections IV and V present a set of experiments of clustering students and results discussions, respectively. Finally, section VI concludes the paper and opens the door to further research.

II. PARAMETERS INFLUENCE ON ANTS BEHAVIOR BASED ALGORITHM

Parameter tuning is to find appropriate parameter settings of algorithms in order to optimize their performance. It has a strong impact on the accuracy of ant colony based algorithms since it controls their behavior. Ant colony algorithms have several parameters controlling different aspects. In this section, we analyze the most important parameters as well as their influence on the algorithm performance. As illustrated in Table 1, we categorise these parameters depending on their related factors such as Pheromone, Ants' movement and similarity related parameters.

A. PHEROMONE RELATED PARAMETERS

Artificial ants communicate by laying synthetic pheromone along the edges on their path through a decision graph. This attracts following ants likely to search in the same region of the search space. In addition, pheromone values are used and updated by the Ant Colony algorithm during the search.
Both [13] and [14], studied the impact of the pheromone related parameters α_p, p, Q and Δ_τ on Ant colony optimization algorithm and Ant colony system. Authors in [14] proposed an hybrid algorithm called Harris's hawk optimizer ant colony system (HHO-ACS) in order to improve the path finding behavior of the traditional ACS algorithm (get the optimal path). The aim of their work is to tune ACS parameters using Harris's hawk optimization in solving the traveling salesman problem. Their proposed algorithm was compared to other well known meta-heuristics such as PSO, dragonfly, GA and ACO algorithms. Similarly, [13] proposed a hybrid clustering method named BACOK (Basic ACO improved by k-means algorithm). After extensive parameter fitting, their experiments show that their method performs the compared algorithm (BACO and K-means) in reasonable time on five real life data-sets. In addition, they revealed that the ACO algorithm is very sensitive to p, α_p and β parameters. For instance, high values of the pheromone evaporation rate p make the pheromones evaporate faster and the past solutions are easily forgotten and the exploration of new solutions (better or worse) is encouraged. Whereas, low values make past solutions more important on the construction of new ones and helps a good solution to converge, either to a local or global optimum. On the other hand, even though they stated that the Pheromone amplification constant Q has negligible influence, it affects the convergence speed of the ACO to a certain extent. As a matter of fact, if it is large, the pheromone concentration will be highly concentrated, making the algorithm fall into a local optimum. In the same context, [15] proposed an algorithm for tuning ACO parameters (p and Δ_τ) and they proved how they affect the performance of ACO which affects in its turn the performance of grid environment when applied for scheduling. Actually, the pheromone update quantity Δ_τ aims to enhance the diversity of algorithm search and to avoid getting into local optimal. It determines the search efficiency of artificial ants, and then affects the optimization performance and evolution speed of the whole algorithm.

B. ANTS' MOVEMENT RELATED PARAMETERS

As we mentioned above, ACO algorithm is sensitive to some parameters such as the key parameter α_p. It is a weight assigned to pheromone concentration deposited by ants and it aims at controlling the relative importance of the pheromone trail concentrations. In other word, it reflects the strength of stochastic factors in the path search of ant colony. The greater the value, the greater the likelihood of re-choosing the path which weakened the randomness of search. Therefore, when α_p value is too large, this can also make ant colony search prematurely trapped in local optima. In fact, α_p is a complement for the parameter β and they are closely related. Their combinations are used to discuss their impact on the performance of ant colony algorithm. Indeed, β is a weight assigned to ant visibility; it reflects the strength of ant colony in priory of path search and uncertainty factors. The greater the β value, the greater the likelihood for ants to select local shortest path on a local point, although the search convergence rate speed up. However, ant colony in search of the optimal path weaken randomness to fall into local optimum easily. Actually, ant colony parameters, in literature, could have different labels. For example, [14] and [15] tuned the same parameter q_0 and p_0 respectively which control the probability of ants movements between objects. Thus, they possess the same role. As suggested in the literature, good values of this parameter tend to be close to 1 and perform best for a short run-time. In extreme cases, a value of 1 quickly leads to search stagnation. Low values of 0 generally result in better final performance and values smaller than 0.75 produce very slow convergence towards good solutions. Therefore, the strategies that decrease the speed of this parameter result in faster convergence to good solutions.

C. SIMILARITY RELATED PARAMETERS

Ant colony optimization and Ant colony Clustering (ACC) are two algorithms inspired from ants' behavior. The difference between both of them is described in the following: On one hand, ACO algorithm is inspired by ant's shortest food path search, ants Lay down synthetic pheromones along the edges on their path through a decision graph. This attracts following ants likely to search in the same region of the search space. Therefore, α_p parameter presents the weight assigned to pheromone concentration deposited by ants. On the other hand, ACC is arising from the nest organization of ants: building cemeteries or brood pits, clustering their corpses, sorting their larvae, etc. into several piles. Each ant starts by walking randomly around the space, thus, based on the similarity and density of the data items within the ants' local neighborhood, ants are likely to pick up items that are surrounded by dissimilar ones and tend to drop them in the cluster of similar ones. Ants distinguish their living nest-mates from dead ones through their specific corpse odor. Therefore, α parameter in the local density function presents the scaling dissimilarity parameters that allows the decision to have or not to have two items next to each other. In this context, [16] adapt ant based clustering behavior and they propose a new Abstraction Ant Colony Clustering (AACC) algorithm using data combination mechanism to improve the computational efficiency and accuracy of the AACC algorithm. Their results show that the AACC algorithm can solve the clustering problem with a high degree of accuracy and speed while providing a very good computing stability compared with ACC and K-means algorithm. They also presented a study of the main parameters that significantly affects both AACC and ACC such as the threshold for picking up object "K_p", the constant C for picking up/dropping probability parameter which can speed up the algorithm convergence if increased. During the clustering, some objects (called outliers) with high dissimilarity to all other data elements. The outliers prevent ants from dropping them, which slows down the algorithm convergence. Therefore, a larger parameter C forces the ant to drop the outliers at the later stage of the algorithm-knowing that the role of α and $\alpha 1$ is to adjust the similarity between objects and the similarity between data reactors respectively. They affect the number of clusters and the algorithm convergence rate. Authors in [16] stated that these parameters for both AACC and ACC can be determined by trials based on sensitivity analysis results for each data-set. In addition, the influence laws of parameters for different data-sets are similar. According to influence laws from the sensitivity analysis results shown in their study, the suitable values of parameters can be determined by trials: Selecting initial parameters values according to previous experiences or studies, changes them by trials and finally leads to the suitable values through some trials. As a result, the values found for the main parameters were different in each data-set. Other parameters that barely affect these algorithms can be determined through testing and experimenting and these parameters can be fixed for different data-sets as for the number of objects in the data reactor visited by the current ant.

To conclude, we cannot deny the fact that almost all Ant Colony parameters influence its performance and accuracy. However, α has always been a field of studies, we can notice that almost all cited research papers have studied the value of α due to its importance and influence on the algorithm performance. For this reason, in this paper we present a study of the key parameter α and its sensitivity when applying ACC algorithm for constructing collaborative learning teamwork.

D. ALPHA (α) PARAMETER SENSITIVITY ANALYSIS

As stated earlier, bio-inspired algorithms are sensitive to their parameters' settings, as they influence on the clustering robustness and performance. In this research work, we pinpoint the key parameter α since it plays an important role in Ant Colony Clustering algorithms. In fact, α presents the scaling dissimilarity parameters that determine when two items should, or should not, be located next to each other. As to say, α adjusts the similarity between objects and determines the percentage of items on the grid that are classified as similar. Therefore, a too large choice of α leads to the fusion of individual clusters, and in some cases, all items could be gathered within one cluster. Nontheless, a too small choice of α could prevent the formation of clusters on the grid [17].

Accordingly, α affects the number of clusters and the algorithm convergence rate. Objects with greater degrees of similarity have greater values of α and tend to cluster. Thus, the number of clusters decreases, and the algorithm becomes faster. On the contrary, if α is smaller, the objects have smaller degrees of similarity, and the larger group will split into smaller groups. Thus, the number of clusters will increase, and the algorithm will become slower [16]. Therefore, during the experimental studies, we notice that α affects clearly the clustering results. This paper studies the effect of α parameter on different dimensional Educational data-sets.

III. REVIEW ON ANT COLONY ALGORITHMS FOR DATA CLUSTERING

Several researchers referred to some entomologists [18] and [1] who had studied the ant colonies to recognize the behavior of real ants in forming cemeteries. In this section, we give a brief presentation of the selected ant colony algorithms that

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3279723

TABLE 1. Review summary on ant based clustering algorithms parameters

Taxonomy	Parameters	Ants Behavior Inspired Algorithms				Description	Parameters Range	Ref
		ACO	ACS	ACC	AACC			
Pheromone related parameters	α_p	X	X			Weight assigned to pheromone concentration deposited by ants	[0, 0.25, 0.5, .., 6]	[13] [14]
							{0.8848, 0.8929, 0.9419, 0.9822, 1.0327, 1.0678, 1.1288, 1.1919}	
	p	X	X			Pheromone evaporation rate	[0.1, 0.2, .. , 0.9]	
							{0.0993, 0.0994, 0.0995, 0.997, 0.0999}	
	Q	X				Pheromone amplification constant	[50, 100, 150, .., 500]	[13]
	Δ_τ	X				Deposited pheromone of ants	[0.00155 .. 0.00161]	[15]
Ant's movement related parameters	β	X	X			Weight assigned to ant visibility	[0, 0.25, 0.5, .., 6]	[13]
							{ 1.9904, 1.9912, 1.9914, 1.9926, 1.9942, 1.9957, 1.9983, 1.9995}	[14]
	q_0	X	X			Control the probability of ant's movements between objects	{0.8907, 0.8909, 0.8938, 0.8941, 0.8942, 0.8944, 0.8957, 0.8970}	
	p_0						[0.1467 .. 0.1473]	[15]
Similarity related parameters	α			X	X	Adjust the similarity between objects	{0.45, 0.5, 0.6, 0.7, 1.5}	[16]
	$\alpha 1$				X	Adjust the similarity between data reactors	{0.25, 0.3, 0.4, 0.5}	
	k_c				X	Threshold for similar data reactor combination	{0.05, 0.15, 0.25}	
	C			X		Constant for picking up/dropping probability	{3, 5, 6, 8}	
	k_p				X	Threshold for picking up one data object.	{0.05, 0.1, 0.2}	

were applied in this research study.
These algorithms arise from ant colony behavior and how they achieve the goal of clustering data: At the beginning, the dead ants are scattered in the nest. Hence, in order to clean the nest from corpses and cluster them, each ant starts by walking randomly around the space. If it hits one of corpses, the ant has two choices: to pick up the corpse if it is not in a dense location of dead ants, then moves it and drops it where they are surrounded by other similar dead ones. The second choice is to let it there if it is already surrounded by other similar ones. The ant distinguishes its living nest-mates from dead ones through their specific corpses odor.

A. THE L&F ALGORITHM

L&F (for Lumer & Faieta) algorithm proposed by [19] operates on a fixed regular low-dimensional grid where ants are generated in a simulation environment. Its main idea is to define the measure of similarity and dissimilarity among the different objects/items, then to cluster them into well defined groups. L&F algorithm contains three major phases which are: the initialization phase, the simulation phase and the cluster phase which are illustrated as following:
Phase 1: Called the initialization phase where the ants and items o_i are randomly settled on the grid.
Phase 2: Is the activity simulation phase which evolves in discrete time t steps. Each ant is randomly selected and moved along the grid; if there is a pattern on its current location, it can pick it up; otherwise, if the object is already carried, the ant can drop it. The probabilities of picking up p_{pick} or depositing $p_{deposit}$ a pattern are defined by the following equations:

$$p_{pick} = (\frac{\gamma_{pick}}{\gamma_{pick} + F})^2 \tag{1}$$

$$p_{deposit} = \begin{cases} 2\,F(o_i), when\ F(o_i)\ <\ \gamma_{deposit} \\ 1, when\ F(o_i)\ \geqslant\ \gamma_{deposit} \end{cases} \tag{2}$$

with F is the perceived fraction of local density of neighboring sites occupied by data points of the same type, γ_{pick} and $\gamma_{deposit}$ are two threshold constants. Generally, the probability of picking up an object increases if it is surrounded by dissimilar data, or when there is no data in its neighborhood. The local density with respect to object o_i is given as follows:

$$F(o_i) = \begin{cases} \frac{1}{\sigma^2} \sum_{o_j \in neigh(s*s)(r)} (1 - \frac{d(i,j)}{\alpha}), \\ if\ F \geqslant 0 \\ 0\ , otherwise \end{cases} \tag{3}$$

Where α is the scaling dissimilarity parameters that allows the decision to have or not to have two items next to each

other.
The σ represents the neighbor size, $\sigma \in$ [9,25]. And *r* is the ant located side and *s* is the neighborhood scaling parameters.
phase 3 : Called the clustering phase of objects occurring in order to determine the boundaries between the groups.

B. THE ACA ALGORITHM

Stated in [1], the Ant Clustering Algorithm (ACA) and the Ant Clustering Algorithm Modified version (ACAM) exploit the short-term memory of each ant. It is the number *n* of items that come across during the last time *t*. Each artificial ant uses its memory depending on the following assumption: If an ant is located at a specific cell on the grid, and carrying an object o_i, it employs its own memory to proceed to all remembered placements, one after the other. Each of them is evaluated using the neighborhood/ density function $F^*(o_i)$ to find the fittest site to deposit the current carried object.

These proposals itemized by The ACA algorithm have also three main phases:
Phase 1: In the initialization phase, items are randomly dispersed on the grid. Thereafter, each ant starts by selecting the object o_i at a random way, then pick it and move it to another empty grid location.
Phase 2: The simulation phase is evolved too in a discrete time *t*, the ants are randomly selected and moved to other new locations. Then, the ants carry items and measure the similarity between them and the probability of depositing an object. They are computed according to the following equation:

$$F^*(o_i) = \begin{cases} \frac{1}{\sigma^2}\sum_j(1-\frac{d(i,j)}{\alpha}),\ if\ F^* \geqslant 0 \\ and\ \sum_j(1-\frac{d(i,j)}{\alpha}) \geqslant 0 \\ 0 \quad ,otherwise \end{cases} \tag{4}$$

where σ is the neighbor size. For dropping decision, a threshold is expressed as follows:

$$p^*_{deposit}(i) = \begin{cases} 1, \quad if\ F^*(i) \geqslant 1 \\ \frac{1}{F^*(i)^4}, \quad else \end{cases} \tag{5}$$

Subsequently, if the operation of dropping the item is done; the ant continues selecting the object o_i, picks it up, moves it and compares it with other surrounded ones or move it to another empty location until they find the suitable data sites.

The picking probability is calculated by:

$$p^*_{pick}(i) = \begin{cases} 1, \quad if\ F^*(i) > 1 \\ \frac{1}{F^*(i)^2}, \quad else \end{cases} \tag{6}$$

Phase 3: Called the cluster phase in which clusters are finally defined.

C. THE ACAM ALGORITHM

Similarly, authors in [1] proposed another modified version of ACA called ACAM based on the following new modification using a new neighborhood scaling parameter $\frac{S_0^2}{S^2}$ that depends on an adaptive perception range *S*.

The density/neighborhood function in the new version ACAM is computed by:

$$F^*(o_i) = \begin{cases} \frac{S_0^2}{S^2}\sum_j 1-\frac{d(i,j)}{\alpha} \qquad \forall o_j(1-\frac{d(i,j)}{\alpha}) > 0 \\ 0 \ ,otherwise \end{cases} \tag{7}$$

where S^2 represents the perception range of each ant. $\frac{S_0^2}{S^2}$ is the relation between the initial size of perception S_0 and the current size of perception S which defines the new quarter scaling parameter. Consequently, the ACAM solution is proposed in order to overcome the difficulty to determine and distinguish the differences between clusters of different sizes. Based on the previous studies of L&F and ACA, they present a stable values of the neighborhood scaling parameter $\frac{1}{\sigma}$ which may lead to inappropriate results of clusters.

D. THE IMPROVED ACA ALGORITHM

Based on the ACA solution, authors in [16] proposed a new Improved ACA and Abstraction Ant Clustering Algorithm (AACA) which present more efficiency than the previous ones. So, we choose to work on the Improved ACA proposal because it requires fewer parameters than AACA and it does not require applying the combination mechanism which can increase the time complexity.

The process in phases of the Improved ACA is similar to the previous ones with a major modification on the similarity function and the probability measures.
Phase 1: is the initialization phase.

Initialize parameters *n*, *m*, *s*, α, c and v_{max}; initialize a random scattered data items on the grid; ants are randomly scattered on the grid and they start walking along the grid to pick up and compare objects.
Phase 2: The similarity function is defined by equation:

$$F^*(o_i) = \begin{cases} 0, \ \frac{1}{s^2} \\ \sum_{o_j \in Neigh_{sxs}(r)}[1-\frac{d(o_i,o_j)}{\alpha(1+(v-1)/v_{max})}] \end{cases} \tag{8}$$

With s^2 is the total number of sites in the local area of ant, $v \in [1, v_{max}]$ is the speed characterized by each ant.
Hence, The process behind the picking up and dropping probabilities in this algorithm is as defined in [16], [20]:

- The function of picking up probability is calculated by the equation:

$$P_{pick} = 1 - sigmoid(f(o_i)) \tag{9}$$

- The function of deposit probability is calculated by the equation:

$$P_{drop} = sigmoid(f(o_i)) \tag{10}$$

Where the sigmoid function is expressed by the equation:

$$sigmoid(x) = \frac{1-e^{-cx}}{1+e^{-cx}} \tag{11}$$

content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3279723

E. THE SELECTED ANT-CLUSTERING ALGORITHMS COMPARED

Considering the previous descriptions of algorithms, we can deduce that these algorithms differ in the following points:

- In **L&F**, each ant is characterized by an area called a neighbor size. When it selects an item when moving randomly and coming in crowds with similar surrounding items, the ant chooses to drop the selected object.
- In **ACA** and **ACAM**, each ant is characterized by a memory to reduce the random trends. When a new item is picked up by an ant, a comparison process is made with the items in memory. So, the ant moves automatically towards the location of the memorized items most similar to the picked one.
- In **Improved ACA**, the algorithm keeps the same functionality as L&F, but each ant has a specific moving speed v distributed randomly in the range of $[1, V_{max}]$.

The most common thread characterizing these algorithms is that they automatically determine clusters without any prior knowledge about the possible number of clusters, as introduced by [21], [1], [16].

IV. ANT COLONY BASED CLUSTERING FOR CONSTRUCTING LEARNERS' TEAM-WORKS

Our research work process is illustrated in FIGURE 1. First, we start with data preparation and selection of relevant attributes [22]. Then, we select one of the above mentioned ACC algorithms (L&F, ACA, ACAM or Improved ACA). After setting their parameters, we run each algorithm for every 0.01 step between 0,5 and 0,6. All F-measure results are recorded and saved in order to be studied and discussed.

```
PseudoCode: α parameter selection process
    Input   : K Educational Datasets;
              inf← 0.5;
              sup← 0.6
    Output  : α with the best F-measure
Foreach (dataset ∈ K ) do
    Data preparation and selection of relevant attributes;
    Foreach ACC Algorithms ∈ {L&F; ACA; ACAM; Improved ACA} do
        Initializations of parameters;
        while (inf <=α<=sup) do
            Ants are randomly initialized to data;
            Compute the local density function f (data_element) by eq.3;
            Ants move data to the suitable cluster or create a new one;
            Output of the clustering results and recording F-measure;
            α← α + 0.01
        end
        Comparing results and selecting α with the best F-measure;
    end
end
```

FIGURE 1. Pseudo Code of our α parameter selection process

A. EDUCATIONAL DATA-SETS DESCRIPTION

Our experimental benchmark is represented by 14 concrete educational data-sets from the literature. These data-sets are detailed as follows: Two data-sets constructed by Cortez et al. [23], each one has 33 attributes presenting students' academic and social data for a specific core classes: Mathematics and Portuguese language; a package of other data-set called xAPI-Edu-Data collected by Amrieh et al. [24] and [25]. This latter data-set consists of 480 students' records over 12 different subjects. Since we are grouping learners into teams to study together the same subject, it is required to arrange this data-set into subsets according to the topic of the course. Accordingly, we obtain 12 different data-sets which are: Arabic, Spanish, IT (Information Technology), Biology, Chemistry, English, French, Geology, History, Math, Quran and Science.

As depicted by FIGURE 1, the first step of our process is to prepare and improve the quality of the data. So, by checking our data-sets, we noticed that these data were imbalanced. This problem is recognized as one of the major issue in the field of data mining and machine learning. In fact, most related algorithms assume that data is equally distributed. In order to handle this disproportion in the 14 data-sets, we apply a re-sampling method available in the data mining tool WEKA [30]. Furthermore, and for data reduction step, we select the most relevant attributes from this data. In fact, a feature selection action presents an important step in the knowledge-discovery process since it can provide more efficient analysises. This is why it is still an ongoing research topic [26] [27]. Thus, we opt for applying relief feature algorithm (RF) [28] on our data. This part of our research work is already presented and detailed in a previous publication [22].

As related results, TABLE 2 presents the number of the selected attributes for the 14 data-sets after applying the Relief (RF) algorithm detailed in [28]. For instance, after applying the RF algorithm on the Arabic subject, the selected attributes were limited from{ StudentAbsenceDays, GradeID, StageID, Relation, PlaceofBirth, Raisedhands, ParentschoolSatisfaction, ParentAnsweringSurvey, NationalITy, AnnouncementsView, gender, VisITedResources, Discussion, SectionID, Semester} to {GradeID, StageID,ParentschoolSatisfaction, ParentAnsweringSurvey, VisITedResources, AnnouncementsView, StudentAbsenceDays, raisedhands, Relation, Discussion}.

Finally, as Lumer and Faieta [19] modified and extended the basic model proposed by Deneubourg et al. [29] for clustering using robotic ants, to a numerical data analysis model, we apply, accordingly, pre-processing mechanisms to prepare our data in order to be ready for the ACC Algorithms. First, we transform our data-sets into numerical data. Therefore, all attributes should take a value in [0..1]. For example, the attribute "ParentschoolSatisfaction" has two possible values "Good" or "Bad", so, we categorize it into two cells (the value "Good" takes "1" and "Bad" takes 0). Then, we normalize all numerical values in order to have a common scale for all numerical attributes values in the same range [0,1].. For instance, the attribute "raisedhand" is how many times students raise their hand in class during their e-learning course. Accordingly, we use the Min-Max normalization defined by the following equation:

$$z_i = \frac{x_i - min(x)}{max(x) - min(x)} \tag{12}$$

where x=$x_1, x_2, ..., x_n$, m is the number of instances and z_i is the i^th normalized data.

TABLE 2. The obtained number of the most relevant selected attributes, after applying Relief Algorithm, for each data-set [22]

Data-sets	#of instances	#of Attributes before Rf Algo	#of Attributes after Rf Algo
xAPI-IT	95		13
xAPI-Spanish	25		9
xAPI-Arabic	59		10
xAPI-Biology	30		10
xAPI-Chemistry	24		11
xAPI-English	45	15	14
xAPI-French	65		13
xAPI-Geology	24		10
xAPI-History	19		8
xAPI-Math	21		11
xAPI-Quran	22		12
xAPI-Science	51		11
Mathematics Course	395	33	24
Portuguese language Course	649		

B. ANT CLUSTERING ALGORITHMS PARAMETERS SETTINGS

To be well executed, some parameters and conditions of the selected algorithms has to be initialized. Hence, TABLE 3 presents the initialization parameters of the four Ant Clustering algorithms.

As mentioned previously, the value of α is data dependent and it is determined experimentally by repetitive executions of each ACC algorithm. In fact, the key parameter α has attracted several researchers and has always been under study because of its strong influence on Ant Colony Algorithms' performance. After several simulation experiments, we notice that the best accuracy value in most cases is when α is between 0.5 and 0.6. So, we investigate the influence of α for every 0.01 in the range of [0.5 .. 0.6]. Thus, TABLE 4 presents the α values elected for each algorithm per data-set. As for xAPI-French data-set, the selected α values in regards to L&F, ACA, ACAM and Improved ACA are 0.54, 0.55, 0.57 and 0.54 respectively; since they show the best F-measure value for each clustering algorithm.

C. ANT CLUSTERING ALGORITHMS FOR LEARNERS' GROUPING

As an instance of our clustering evolution, L&F algorithm simulation is illustrated in FIGURE 2, where data-sets items are randomly scattered on the grid. By running the algorithm, 10 artificial ants start exploring the environment with random movement along the grid. Then, each one select data point and move it toward its new group based on grade similarities during 20,000 iterations. For performance reasons, the number of ants should be kept small. Since ants walk randomly on the grid, too many ants should not have any effect, i.e., two ants coincide many times, over and over again, but they follow different walk [1]. FIGURE 4 illustrates three states of L&F algorithm: At the start point, at the 10,000th iteration and the last point of the 20,000th iteration.

TABLE 3. The algorithms parameters used during the simulation. These parameters are randomly generated according to [1] and [16] studies

Parameters	Algorithms			
	L&F	ACA	ACAM	Improved ACA
γ_{pick}	0.1	-	-	-
γ_{drop}	0.15	-	-	-
σ	5	5	-	5
Memory	-	8	8	-
Max speed	-	-	-	20
Sigmoid(f(o_i))	-	-	-	c=5
Number of ants	10			
Number of Iterations	20,000			
Neighborhood radius	9			
α	See TABLE 4			

TABLE 4. α values derived after multiple explorations per data-set

Data-Sets	α Value			
	L&F	ACA	ACAM	Improved ACA
xAPI-IT	0.5	0.5	0.57	0.57
xAPI-Spanish	0.55	0.52	0.5	0.54
xAPI-Arabic	0.55	0.51	0.54	0.51
xAPI-Biology	0.5	0.51	0.55	0.59
xAPI-Chemistry	0.5	0.58	0.51	0.6
xAPI-English	0.53	0.5	0.52	0.53
xAPI-French	0.54	0.55	0.57	0.54
xAPI-Geology	0.55	0.56	0.5	0.56
xAPI-History	0.59	0.51	0.5	0.55
xAPI-Math	0.54	0.51	0.5	0.51
xAPI-Quran	0.5	0.56	0.5	0.59
xAPI-Science	0.57	0.55	0.52	0.56
Mathematics Course	0.52	0.5	0.57	0.57
Portuguese language Course	0.57	0.56	0.59	0.59

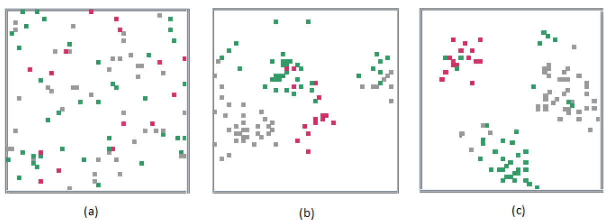

FIGURE 2. Simulation of L&F clustering algorithm at (a) start, (b) iteration 10,000 and (c) iteration 20,000

D. PERFORMANCE METRIC

We evaluate the clustering accuracy of the four selected ACC algorithms based on F-measure, which is bound by the interval [0..1]. It represents the harmonic mean between the precision and recall of the clustering for all classes presented as following:

$$F_{combined} = \sum_i \frac{n_i}{n} max_j F(i,j) \tag{13}$$

Where:

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3279723

- n: the total number of objects being clustered.
- n_j : the number of objects in class i.
- $maxF(i,j)$: the maximum F-measure that each class sees over all clusters given by:

$$F(i,j) = \frac{2Precision(i,j)Recall(i,j)}{Precision(i,j) + Recall(i,j)} \quad (14)$$

Where:

- $Precision(i,j)$: the proportion of items in cluster j that are of class i given by (14).
- $Recall(i,j)$: the proportion of class i that belongs to that cluster j given by (15).

$$Precision(i,j) = \frac{n_{ij}}{n_j} \quad (15)$$

$$Recall(i,j) = \frac{n_{ij}}{n_i} \quad (16)$$

Where:

- n_{ij}: the number of objects of class i within cluster j.

V. RESULTS DISCUSSION

This section presents in details the conclusions draw from the experimental results after applying the ant algorithms to a benchmark consisting of 14 Educational data-sets provided by [23], [24] and [25].

A. EVALUATION OF THE ANT BASED CLUSTERING ALGORITHMS PERFORMANCE BASED ON F-MEASURE

In this section we investigate the performance of the four selected ant colony based clustering algorithms in educational fields. As depicted by FIGURE 3, the reported results can give us a rough estimation about the effectiveness of the ant algorithms. The obtained radar diagram visualises the results presented in TABLE 5, obtained at the end-run of 20.000 iterations. This big number of iterations is a common characteristic of different ant-based clustering algorithms, and even more for any heuristic method.

Therefore, we can conclude that ACAM algorithm outperform the others in most of the tested data-sets. Although, we can notice that it gives good results especially using data-sets with reduced number of instances (between 19 and 95 instances), while Improved ACA algorithm proves that it can handle data-sets with much more instances (395 and 649 instances).

B. SENSITIVITY ANALYSIS OF α PARAMETER FOR ACAM AND IMPROVED ACA ALGORITHMS

Based on the above results, we choose to deepen our analysis about the sensitivity of α parameter for ACAM and Improved ACA algorithms since they outperform L&F and ACA algorithms. In fact, we analyze the impact of α value on the F-measure of ACAM with xAPI data-sets and Improved ACA with Mathematics/Portuguese course data-sets.

FIGURE 4 and 5, illustrate a potential correlation, nearly to a linear relationship, between α values and the performances of the ACAM and Improved ACA algorithms. On one hand, FIGURE 4 presents the variation of F-measures according to the growing values of α from 0.5 to 0.6 for xAPI data-set. We can notice that there is a negligible influence of α on the ACAM algorithm performance. On the other hand, as illustrated by 5, linear function of Portuguese course data-set, which is almost double sized than the Mathematics one, shows that when α increase the F-measure decrease. While, for the Mathematics data-set, we notice that as long as α increase, F-measure increase too. Thus, these findings prove that as long as the data-sets dimension increase, the influence of α parameter on the performance of ACC algorithms, increase too.

TABLE 5. Comparative F-measure results of L&F, ACA, ACAM and Improved ACA algorithms

Data-sets	L&F	ACA	ACAM	Improved ACA
xAPI-IT	0.79	0.9	**0.92**	0.8
xAPI-Spanish	0.61	0.55	**0.97**	0.58
xAPI-Arabic	0.71	0.94	**0.98**	0.49
xAPI-Biology	0.52	0.6	**0.98**	0.53
xAPI-Chemistry	0.57	0.62	**0.97**	0.5
xAPI-English	0.72	0.75	**0.98**	0.53
xAPI-French	0.68	0.89	**0.96**	0.49
xAPI-Geology	0.8	0.74	**0.95**	0.74
xAPI-History	0.57	0.56	**0.99**	0.58
xAPI-Math	0.57	0.65	**0.93**	0.51
xAPI-Quran	0.56	0.54	**0.97**	0.59
xAPI-Science	0.72	0.77	**0.92**	0.57
Mathematics Course	0.71	0.47	0.59	**0.79**
Portuguese language Course	0.49	0.37	0.59	**0.94**

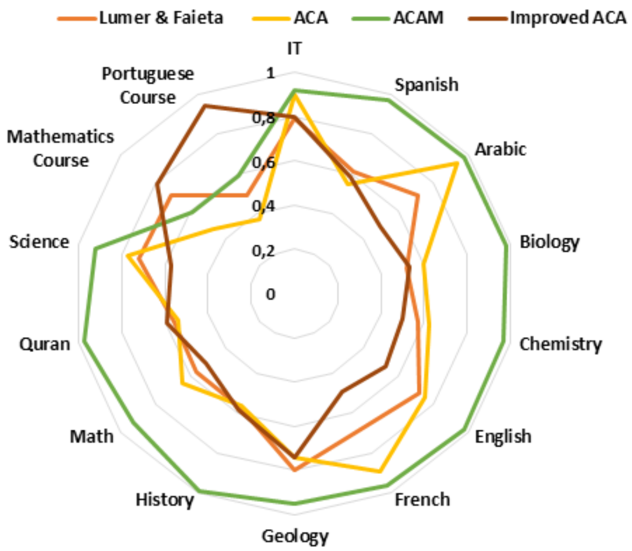

FIGURE 3. Radar analysis performances of L&F, ACA, ACAM and Improved ACA algorithms applied to 14 educational Data-Sets

FIGURE 4. α sensitivity for ACAM algorithm applied on:
a: History, Math and Quran xAPI-Data-sets
b: Chemistry, Geology and Spanish xAPI-Data-sets
c: Biology, English and Science xAPI-Data-sets
d: Arabic, French and IT xAPI-Data-sets

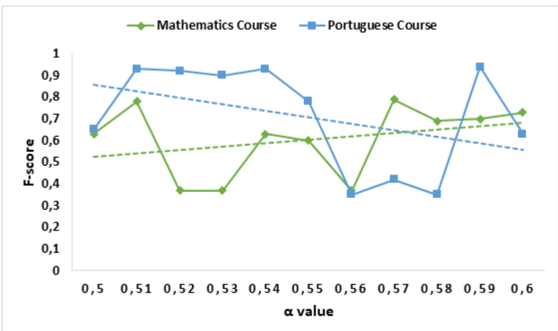

FIGURE 5. α sensitivity for Improved ACA algorithm applied on Mathematics and Portuguese Data-sets

C. RELATIONSHIP BETWEEN α VALUE AND ATTRIBUTE SELECTION

As stated in [22], selecting relevant attributes and reducing redundant/irrelevant features improve the performance of the algorithms. Accordingly, in this section, we investigate the relationship between α value and attribute selection technique. We carried out a series of Improved ACA algorithm running on both Mathematics and Portuguese course data-sets before and after attribute selection for each value of α between 0,5 and 0,6. As illustrated in FIGURE 6, the performance of the algorithm does not show any difference whether we applied RF algorithm for selecting relevant attributes or not. In fact, ACC algorithms accuracy is barely influenced by α value when applying on small data-sets (i.e. xAPI-Data-sets). As results, feature selection step could be ignored since it has a negligible influence on the algorithm performance even with a different value of α. However, data-sets with large number of features lead to high computational complexity for ACC algorithms [32]. Thus, further analyses are needed to investigate this issue especially to find the best trade-off between run-time complexity and clustering performance. Whereas, the findings presented in FIGURE 7 show that selecting appropriate features do affect the performance of the Improved ACA algorithm when applying to the Portuguese course data-sets. In addition, when applying the algorithm on the subset, the best accuracy is when α equals 0.59. However, using the full-set data-set the value of α for the best F-measure is 0.53. Therefore, the fact that stands out from these findings is that α parameter has a remarkable and a strong influence on Ant based Clustering algorithm

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3279723

performance especially for high dimensional data-sets.

FIGURE 6. Improved ACA performance with and without feature selection applied on Mathematics Data-sets for different α values

FIGURE 7. Improved ACA performance with and without feature selection applied on Portuguese Data-sets for different α values

VI. CONCLUSION

Many research works study the influence of Ant Colony algorithms parameters. Distinctively, this paper focuses on the α parameter sensitivity of ant colony based clustering algorithms performance, when applied to educational data-sets for constructing teamworks, in a collaborative learning environment [31]. The goal of this research study is to analyze the influence of this parameter on low and high dimensional data-sets. The obtained findings attest that Ant Colony algorithms performance intensely depend on α, but also, they depend on the data-sets dimension used. For instance, Improved ACA algorithm outperformed other algorithms when applied to high-dimensional data-sets. However, it is not efficient for low-dimensional data-sets and vice versa for ACAM algorithm. Therefore, according to our study, we find out that Ant Colony based clustering algorithms performance is related to many factors (i.e. Algorithm parameters, Data-set dimensionality , etc.). Thus, tuning parameters of ant based clustering algorithms is still considered as a challenging task [32].

As perspectives, it will be interesting to work on automatic selection of α values by employing learning techniques such as those based on genetic fuzzy systems for example [33]. In addition, it should be possible to extend the deep exploration for a wider range α, and study the influence of the other ant colony algorithms parameters, such as the number of ant and similarity between data reactors since they can impact the algorithm accuracy.

FUNDING DETAILS

This work was supported by the university of Sfax through the alternate scholarship granted to spend a 3-month internship at the University of Las Palmas de Gran Canaria, and by grants from Tunisian General Direction of Scientific Research (DGRST) under the ARUB program.

DISCLOSURE STATEMENT

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

DATA DEPOSITION

The data-sets analysed during the current study are owned and cited by the authors of the papers cited in [23], [24] and [25].

REFERENCES

[1] Boryczka, U. (2009). Finding groups in data: Cluster analysis with ants. Applied Soft Computing, 9(1), 61-70.
[2] Bonyadi, M. R., and Michalewicz, Z. (2017). Particle swarm optimization for single objective continuous space problems: a review. Evolutionary computation, 25(1), 1-54.
[3] Forrest, S. (1996). Genetic algorithms. ACM Computing Surveys (CSUR), 28(1), 77-80.
[4] Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In Proceedings of ICNN'95- international conference on neural networks, volume 4, pages 1942–1948. IEEE.
[5] Shami, T. M., El-Saleh, A. A., Alswaitti, M., Al-Tashi, Q., Summakieh, M. A., and Mirjalili, S. (2022). Particle swarm optimization: A comprehensive survey. IEEE Access.
[6] Dorigo, M., and Di Caro, G. (1999, July). Ant colony optimization: a new meta-heuristic. In Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406) (Vol. 2, pp. 1470-1477). IEEE.
[7] Teodorovic, D., and Dell'Orco, M. (2005). Bee colony optimization–a cooperative learning approach to complex transportation problems. Advanced OR and AI methods in transportation, 51, 60.
[8] Nannen, V. and Eiben, A. E. (2006). A method for parameter calibration and relevance estimation in evolutionary algorithms. In Proceedings of the 8th annual conference on Genetic and evolutionary computation, pages 183–190.
[9] Hassanat, A., Almohammadi, K., Alkafaween, E., Abunawas, E., Hammouri, A., and Prasath, V. S. (2019). Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. Information, 10(12):390.
[10] Carlisle, A. and Dozier, G. (2001). An off-the-shelf pso. proceedings of the workshop on particle swarm optimization. Purdue school of engineering and technology, Indianapolis, IN.
[11] Trelea, I. C. (2003). The particle swarm optimization algorithm: convergence analysis and parameter selection. Information processing letters, 85(6), 317-325.
[12] Van den Bergh, F., and Engelbrecht, A. P. (2006). A study of particle swarm optimization particle trajectories. Information sciences, 176(8), 937-971.
[13] Chavarría-Molina, J., Fallas-Monge, J. J., and Trejos-Zelaya, J. (2020). Clustering via ant colonies: Parameter analysis and improvement of the algorithm. In Advanced Studies in Behaviormetrics and Data Science (pp. 265-282). Springer, Singapore.
[14] Yasear, S. A. and Ku-Mahamud, K. R. (2021). Fine-tuning the ant colony system algorithm through harris's hawk optimizer for travelling salesman problem. International Journal of Intel- ligent Engineering and Systems, 14(4):136–145. 18 Evolutionary Computation Volume x, Number x
[15] Sahana, S. K. (2019). An automated parameter tuning method for ant colony optimization for scheduling jobs in grid environment. International Journal of Intelligent Systems and Applications, 11(3), 11.

[16] Gao, W. (2016). Improved ant colony clustering algorithm and its performance study. Computational Intelligence and Neuroscience, 2016.

[17] Boryczka, U. (2008). Ant clustering algorithm. Intelligent Information Systems, 1998:455–458.

[18] Anari B., Torkestani J. A., and Rahmani A. M. (2018, July). A learning automata-based clustering algorithm using ant swarm intelligence. Expert systems, vol. 35, no 6, p. e12310.

[19] Lumer, E. D., and Faieta, B. (1994, July). Diversity and adaptation in populations of clustering ants. In Proceedings of the third international conference on Simulation of adaptive behavior: from animals to animats 3: from animals to animats 3 (pp. 501-508).

[20] Chniter, M., Abid, A., and Kallel, I. (2018). Towards a Bio-inspired ACO Approach for Building Collaborative Learning Teams. In 2018 17th International Conference on Information Technology Based Higher Education and Training (ITHET), pages 1–8, Olhao. IEEE.

[21] Kao, Y. and Li, Y. L. (2008). Ant colony recognition systems for part clustering problems. Inter- national Journal of Production Research, 46(15):4237–4258

[22] Abid, A., Kallel, I., Blanco, I. J., and Benayed, M. (2017, December). Selecting relevant educational attributes for predicting students' academic performance. In International Conference on Intelligent Systems Design and Applications (pp. 650-660). Springer, Cham.

[23] Cortez, P. and Silva, A. M. G. (2008). Using data mining to predict secondary school student performance.

[24] Amrieh, E. A., Hamtini, T., and Aljarah, I. (2015, November). Preprocessing and analyzing educational data set using X-API for improving student's performance. In 2015 IEEE Jordan conference on applied electrical engineering and computing technologies (AEECT) (pp. 1-5). IEEE.

[25] Amrieh, E. A., Hamtini, T., and Aljarah, I. (2016). Mining educational data to predict student's academic performance using ensemble methods. International Journal of Database Theory and Application, 9(8), 119-136.

[26] Peng, H., Ying, C., Tan, S., Hu, B. and Sun, Z. (2018, November). An improved feature selection algorithm based on ant colony optimization. IEEE Access., vol.6, pp. 69203–69209.

[27] Abd E., Diaa S., Ibrahim, S.A., Houssein E.H., and Elsayed, S. M (2022, January). An efficient chaotic gradient-based optimizer for feature selection. IEEE Access., vol.10, pp. 9271–9286.

[28] Kononenko, I. (1994). Estimating attributes: analysis and extensions of RELIEF. In European conference on machine learning, pages 171–182. Springer.

[29] Deneubourg, J. (1991). The dynamics of collective sorting robot-like ants and ant-like tobots. In Proceedings of the first international conference on simulation of adaptive behavior. The MIT Press.

[30] Smith, T. C., and Frank, E. (2016). Introducing machine learning concepts with WEKA. In Statistical genomics (pp. 353-378). Humana Press, New York, NY.

[31] Abid, A., Kallel, I., and Ayed, M. B. (2016, September). Teamwork construction in E-learning system: A systematic literature review. In 2016 15th International Conference on Information Technology Based Higher Education and Training (ITHET) (pp. 1-7). IEEE.

[32] Chniter, M., Abid, A., Kallel, I., and Kanoun, S. (2022, October). Computational Complexity Analysis of Ant Colony Clustering Algorithms: Application to Student's Grouping Problem. In 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 736-741). IEEE.

[33] Kallel, I. Alimi, A. M. (2006). MAGAD-BFS: A learning method for Beta fuzzy systems based on a multi-agent genetic algorithm. Soft Computing, 2006, vol. 10, p. 757-772.

ABIR ABID is a researcher and PhD student in computer systems Engineering at the National Engineering School of Sfax, Tunisia (ENIS) and a member of the Research Groups in Intelligent Machines laboratory (ReGIM-Lab) since 2015. She received the bachelor degree in computer science in 2012 and the Research Master degree in 2015 from the Higher Institute of Computer Science and Multimedia (ISIMS) Sfax, Tunisia. Abir joined IEEE in 2015 as a graduate student member within IEEE ISIMS Student Branch. She is actually holding the position of IEEE Education Society Tunisia Chapter secretary (2021 - 2022) She is an active IEEE member with various and appreciated volunteering. She started her IEEE student membership in 2015. She served as Local Arrangement of IEEE ENIS Student Branch for 2016-2017. She served also as Vice chair of IEEE ISIMS Student Branch for 2019-2020 and within IEEE Education Society Tunisia Chapter she served as Secretary/Webmaster for (2017-2018) and Vice chair for (2019-2020). She is also a member in different other technical societies like ES, CS, and WIE. She participated in various IEEE conferences as an organizing/technical committee member such as 2022 IEEE Global Engineering Education Conference(EDUCON)

ILHEM KALLEL is an A. Professor with the Department of Computer Science Engineering in ISIMS, and a researcher with ReGIM-Lab in ENIS (university of Sfax-Tunisia). Since 1988, Ilhem had many leadership positions and a rich professional experience dealing with developing and managing enterprise information systems within an industrial company, teaching fields of computer science and training of teachers. Ilhem is participating in the supervision of many engineering research projects and contributing as author and reviewer for many international conferences and journals. Her research interests are mainly in collective and hybrid intelligence, education engineering, intelligent multirobot systems, fuzzy modeling, swarm intelligence, multiagent systems, and systems' quality assessment. Ilhem Kallel is an IEEE senior member. Currently, she serves as a chair of the Tunisia section (2021-2022) and the chair of the Education Tunisia chapter for 2017-2022. On 2009, she was the founder and the past chairwoman of IEEE Tunisia Women In Engineering; she was one past chair of SMC and CIS Tunisia Chapters, the chapters coordinator in Tunisia Section for 2015-2016. Ilhem is a member of the DIS TC in SMC Society. She was the winner of the 2011 WIE Clementina Award for Region 8 for her professional and personal achievements; this award attests that she can serve as a role model for combining professional achievements and family life. She was also recognised as the 2018 R8 Outstanding Women In Engineering Section Volunteer Award.

JAVIER J. SANCHEZ-MEDINA earned his Engineering Master Degree at the Telecommunications Faculty on 2002, and his PhD at the Computer Science Department on 2008. His PhD dissertation versed on the use of Genetic Algorithms, Parallel Computing and Cellular Automata based Traffic Microsimulation to optimize the Traffic Lights Programming within an Urban Traffic Network. His research interests include mainly the application of Evolutionary Computation, Data mining and Parallel Computing to Intelligent Transportation Systems. He has a wide experience on the development of traffic models and simulation platforms. Javier Sanchez-Medina has been volunteering for several years at many international conferences related to Intelligent Transportation, Computer Science, Evolutionary Computation, etc. He is reviewer for some Transportation related journals.. He is also very active as a volunteer of the IEEE ITS Society. Since 2010, we has served for the IEEE ITS Society organizing the TBMO 2010 Workshop at ITSC2010, co-organizing the "Travel Behavior Research: Bounded Rationality and Behavioral Response" Special Session at ITSC2011, being Publications Chair at the IEEE FISTS2011, Registration Chair at the IEEE ITSC2012, Workshops and Tutorials Chair for IEEE ITSC 2013, Panels Chair at IEEE VTC2013-Fall, Program co-Chair at IEEE ITSC2014, program co-Chair at IEEE ITSC2016, publicity chair at IEEE IV2016, program chair at IEEE ICVES 2017, program chair at IEEE ITSC2018 and General Chair at IEEE ITSC2015. Currently he also is EiC of the ITS Podcast, the ITS Newsletter and Vice-president of the IEEE ITSS's Spanish chapter. He has widely published his research with more than 25 international conference articles and more than 15 international journal articles.

MOUNIR BEN AYED is a Professor within the Department of Computer Sciences and Communication in the Faculty of Sciences of Sfax. He has obtained his HdR in Computer System Engineering in 2013 from the National Engineering School (ENIS), University of Sfax, Tunisia, and his PhD in Biomedical Engineering in 1989 from the Val De Marne University, Paris XII. He currently teaches Software Engineering, Data Base Management Systems, Decision Support Systems and data warehouse. He is member of the ReGIM-Lab laboratory. most of his research revolves around data analysis and visualization in the medical field. He was the Chair, Organization Committee Member, and Technical Committee Member of many national and international conferences. He is an IEEE senior member. He founded the IEEE-EMBS Tunisia Chapter in 2009 and chaired/co-chaired this chapter for many years.