BUILDING MULTI-CLASSIFFIER SYSTEMS WITH ANT COLONY OPTIMIZATION

Leidys Cabrera Hernández[1] *, Gonzalo Nápoles Ruiz*, Lester Rene Santos *, Alejandro Morales Hernández*, Gladys M. Casas Cardoso*, María Matilde García Lorenzo *, Yailen Martínez Jiménez *
*Departamento de Computación, Facultad Matemática, Física y Computación, Universidad Central "Marta Abreu" de Las Villas, Cuba.

ABSTRACT
In recent years, the development of multi-classifier systems has become an active research field. A multi-classifier system is an ensemble of classification algorithms whose individual outputs are fused together for better accuracy and interpretability. An important aspect when designing such systems is related to the heterogeneity of the building blocks (classifiers) that make up the ensemble, since previous studies have uncovered that a more diverse ensemble often boosts up the overall classification power. Some statistical measures can be used to estimate how diverse the classifier ensembles are; they are called *diversity measures*. Another issue to be considered is the number of individual classifiers included in the model: the lower the number of classifiers, the simpler the resulting system. In general terms, the parsimony principle is highly desired in such ensembles, since a bulky ensemble will also be a very time-consuming model. Finding the minimal subset of individual classifiers that brings about the best system performance can be posed as a combinatorial optimization problem. In this paper, we address the problem of building multi-classifiers systems from the perspective of Ant Colony Optimization (ACO), a widely popular and effective metaheuristic optimization algorithm. The main reason behind the use of ACO lies on its strong ability to solve entangled combinatorial optimization problems. An empirical analysis is included to statistically validate the benefits of our proposal.

KEYWORDS: Ant Colony optimization, Diversity Measures, Classifier ensemble, Multi-Classifiers systems.

MSC: 68T20.

RESUMEN
En los últimos años el desarrollo de sistemas de multi-clasificadores se ha convertido en un campo de investigación activo. Un sistema multi-clasificador es un conjunto de algoritmos de clasificación cuyas salidas individuales se funden para una mayor precisión e interpretabilidad. Un aspecto importante en el diseño de tales sistemas está relacionado con la heterogeneidad de los bloques de construcción (clasificadores) que componen el conjunto, desde que estudios anteriores han descubierto que un conjunto más diverso a menudo aumenta la potencia de la clasificación general. Algunas medidas estadísticas se pueden utilizar para estimar cuán diversos los conjuntos de clasificadores son, ellas se llaman *medidas de diversidad*.
Otra cuestión a considerar es el número de clasificadores individuales incluidos en el modelo: cuanto menor sea el número de clasificadores, más simple es el sistema resultante. En términos generales, el principio de parsimonia es muy deseado en tales conjuntos, desde que un conjunto voluminoso también será un modelo que consume mucho tiempo. Encontrar el subconjunto mínimo de los clasificadores individuales que produce el mejor rendimiento del sistema se puede plantear como un problema de optimización combinatoria. En este trabajo se aborda el problema de la construcción de sistemas multi-clasificadores desde el punto de vista de la Optimización de Colonia de Hormigas (ACO), un algoritmo de optimización meta heurístico ampliamente popular y eficaz. La razón principal detrás del uso de ACO radica en su fuerte capacidad para resolver problemas de optimización combinatoria entrelazados. Un análisis empírico es incluido para validar estadísticamente las ventajas de nuestra propuesta.

1. INTRODUCTION

Classification problems are among the most widely studied subjects in data mining and machine learning. Despite the plethora of technical papers devoted to the topic, classification techniques continue winning the persistent support among the enthusiasts and researchers in the above disciplines. Choosing the best classifier largely depends on the characteristics of the problem at hand and the nature of the decision boundaries discovered by each technique to separate the different decision classes. In the quest for better classification schemes, the combination of several classifiers aimed at tackling the same problem is a popular trend. This is essential when a multi-classifier system (MCS) is built. An MCS relies on a number of individual classifiers

[1] leidysc@uclv.edu.cu, gnapoles@uclv.edu.cu, alejandromorales@uclv.edu.cu, gcasas@uclv.edu.cu, mmgarcia@uclv.edu.cu

yailenm@uclv.edu.cu

and fusion their outputs through some criteria with the intent of achieving a superior result [35] . In principle, we must expect better results by using an ensemble of classifiers, even in complex pattern classification problems.

Dietteric [11] suggests three reasons why a multi classifier system can be better than a single classifier. The first reason is statistical: if each classifier has a hypothesis, then the idea of combining these hypotheses results in a hypothesis that cannot be the best, but at least could avoid selecting the worst one. The second justification is computational: standard algorithms execute queries that can lead to different local optima; however, in an MCS, each classifier starts the search from a different point. It means that the ensemble of classifiers often produces solutions that are closer to the global solution. The last point is figurative because it is possible that the hypothesis space contains the hypotheses considered as non-optimal, but the approximation of several decision boundaries can result in a new space outside the initial hypothesis that is closer to the optimum.

As witnessed by the diverse number of models available in the literature, there are multiple ways to develop an MCS. Some of them deal with more generic pattern classification problems (i.e., bagging and boosting) whereas other target specific purposes. Despite the application field intended for the model, the creation of accurate MCSs involves two main challenges. The first one is related to the proper choice of individual classifiers (i.e., building blocks) whereas the second one is focused on the combination of their individual classification outputs [6], [5].

The selection of the underlying classifiers is the first step in building an MCS; there are several classic models that have been proposed to this end. Bagging [7] is rooted on the principle of generating different training sets extracted from the initial training set by means of random sampling with replacement, which ensures diversity. This model requires the selection of a weak/unstable classification model, i.e., a classifier that varies its outputs in presence of, e.g., minor parametric changes. Bagging also assumes that all its weak classifiers will be of the same type/family and the merging of their individual outputs is accomplished through the majority vote technique. This algorithm can be applied in learning methods with a numerical decision attribute (e.g., regression problems), in which the individual outputs are real numbers and are hence averaged. Another strategy that Bagging employs to produce the final outcome is to estimate a probability for each output. These probabilities estimated by the models are averaged and the most likely class will be spit out as solution by the Bagging-based MCS [41].

Boosting [37] is similar to Bagging because it also leans upon the same method to create training sets (i.e., random sampling with replacement), from the original training data and the same type/family for all the base classifiers. However, this technique is carried out in a sequential fashion, i.e., a classifier is trained after all its predecessors ones have finished so it can benefit from the classification performance of their previous peers. Another difference is that Boosting gives a weight to a classification model according to its performance rather than equally weighing all models. The replacement is done strategically: the incorrectly classified instances have a higher chance of making their way to the training set of the next base classifier than those that were correctly classified by a previous base model. There are many variants that use the idea of Boosting; AdaBoost is one that enjoys widespread popularity [17].

To summarize: the two previous techniques confine themselves to the same type/family of their underlying classification model and are trained with subsets of their training data. The former approach (Bagging) selects random subsets whereas the second one selects the subsets iteratively based on the result of the previous iteration (classifier). Another scheme is termed Stacking [42], which is used in different classification models that are trained with the same initial set. Stacking differs from the two aforementioned strategies in that it guarantees the ensemble diversity by resorting to classification models of different families. It recruits multiple classifiers generated by different algorithms for the same training set in a first phase. Then, in the second phase, it blends their classification outputs not via majority vote but through a meta-classifier, that learns the relationship between the outputs of the base classifiers and the original class. This meta-classifier is trained on a new set of instances out of the initial training set of the base classifiers, where each instance in this new training set is described by a feature vector composed of the decision classes of each base classifier and as class, the original instance. Stacking can be applied to numerical prediction (i.e., regression) and classification problems [41].

It could be said that these three paradigms are the most general and used when building MCSs, although the best alternative is not easy to determine. Individual MCSs, like simple classifiers, are not intrinsically better than others, but they have to be selected on the basis of their performance against a particular type of problem [26].

Preserving a certain degree of diversity among the base classifiers in the ensemble is a pivotal issue since it

will likely impact the MCS' effectiveness. It makes little sense to combine identical classifiers because the ensemble's behavior may not exceed that of the individual classifiers. Each classifier is able to correctly label a certain percentage of the training set; if the results attained by these classifiers are more diverse, then the probability of having a greater percentage of correctly classified instances by amalgamating their outputs thereof is higher. For example, if a classifier misclassifies an instance and the other classifiers match between them, then the instance can never be correctly labeled. As explained above, does not occur when there are differences between the classes assigned by the classifiers to the instance, i.e., when there are diversity between them [40], [24].

Some MCSs ensure diversity using different training sets, but this only works for classifiers that are sensitive to changes, such as decision trees. Others use different sets of features and thus also vary the training sets. Others use a collection of dissimilar base classifiers. In the latter case, it is difficult to know when a good diversity is ensured, thus making it necessary to resort to some statistical measures that help determine how diverse they are. The diversity measures are described by Kuncheva and other authors in [27]. These measures can be categorized as pairwise and group (non-pairwise) measures.

On the other hand, another aspect to be considered is the number of base classifiers that are to be part of the MCS: the smaller the number of classifiers, the simpler the system. However, finding the minimal subset of classifiers that is also high performing on the problem at hand could be envisioned as a combinatorial optimization problem with an exponential search space; this is due to the fact that new classifiers are being invented at a fairly quick pace. Even with a small bunch of them, we could end up in a highly explosive number of possible combinations. We have therefore decided to pull the Ant Colony Optimization (ACO) metaheuristic algorithm, which has proved to be a very viable and robust alternative for coping with complex search and optimization problems thanks to its biologically inspired and highly parallelized nature.

There are in the literature other approaches that uses metaheuristics to build MCS, for example, some of them apply metaheuristics to realize the best features selection in the database, others use metaheuristics to find the best configuration of parameters in classifiers [9], [30], [5], [31]. However, not exist the same methodology proposed by us in this paper with ACO, and diversity measures to ensure diversity between the classifiers combined.

In light of the aforesaid facts, in this paper we introduce a new methodology for building an MCS by means of an ACO technique. In particular, ACO helps with selecting a suitable group of base classifiers out of a large pool of possible alternatives. This group possesses a good diversity among them and attains the highest classification accuracy for the problem under consideration.

The rest of the text is organized as follows: Section 2 elaborates on the theoretical background and related works regarding the formulation of some diversity measures in MCSs and expose the essential concepts based in formulation of ACO. Section 3 describe how build MCS with ACO taking into account diversity measures. Section 4 is concerned with the experimental analysis and the discussion of the empirical results. Conclusions are given in Section 5.

2. BACKGROUND AND RELATED WORKS

As mentioned before, it does not make sense that MCSs combine identical classifiers between them because a good performance would not be attained, so it is important to know how diverse a classifier ensemble is. A number of ensemble diversity measures have been proposed in the literature. They are divided into two categories: pairwise and non-pairwise.

Diversity Measures

The first set of measures is calculated for pairs of classifiers. Its outputs are binary (0, 1) indicating whether the instance was correctly classified or not. Table 1 shows the results of two classifiers (*Ci, Cj*) for a given instance, depending on whether or not it was correctly classified. If we consider all N instances between the pair of classifiers (*Ci, Cj*), the results summarized in the Table 2 are obtained. It should be observed that a set of L classifiers have associated L (L-1)/2 pairs of values, so to obtain a single result these values must be averaged. N is the total number of cases.

	C_j correct (1)	C_j incorrect (0)
C_i correct (1)	a	b

C_i incorrect (0)	c	d
a + b + c + d = 1		

Table 1: Binary matrix for one instance

	C_j correct (1)	C_j incorrect (0)
C_i correct (1)	A	B
C_i incorrect (0)	C	D
A +B + C + D = N		

Table 2: Binary matrix for N instances

Correlation coefficient ρ

The coefficient of correlation [26], is one of the measures for pairs of classifiers, it is calculated as:

$$\rho_{ci,cj} = \frac{A \times D - B \times C}{\sqrt{(A+B)\times(C+D)\times(A+C)\times(B+D)}} \quad (1)$$

A better diversity is obtained for smaller values ofρ. The values of ρ will be in the interval [-1, 1].

Q Statistics

The Q statistic is one of the measures for pairs of classifiers

$$\mathrm{Qci, cj} = \frac{A \times D - B \times C}{A \times D + B \times C} \quad (2)$$

It has been proved that ρ and Q have the same sign. Also, it can be demonstrated that $|\rho| \leq |Q|$[27].

The Measure of Differences

The measure of differences was introduced by Skalak [38], it is the most intuitive measure between a pair of classifiers, and it is equal to the probability that the two classifiers disagree in their predictions. The diversity increases when the value of D increases.

$$D_{ci,cj} = \frac{B+C}{N} \quad (3)$$

The Double-Fault Measure

Another measure to be analyzed is known as double fault measure, which was introduced by Giacinto and Roli [20]and considers the failure of two classifiers simultaneously. This measure is based on the concept that it is more important to know when simultaneous errors are committed, that when both have a correct classification. The diversity increases when the value of DF decreases.

$$D_{ci,cj} = \frac{D}{N} \quad (4)$$

On the other hand, the non-pairwise measures take into account the outputs of all classifiers at the same time and calculate a unique value of diversity for the whole ensemble.

Entropy

This measure was introduced by Cunningham and Carney [10]:

$$E = \frac{1}{N} \times \frac{2}{L-1} \times \sum_{j=1}^{N} min\{(\sum_{i=1}^{L} Yj,i), (L - \sum_{i=1}^{L} Yj,i)\} \ , Yj, i \in \{0,1\} \quad (5)$$

Where $Y_{j,i}$ will be 1 if the classifier *i* was correct in the case *j*, and 0 otherwise. If *E* is equal to zero then there is not a difference between the classifiers and if *E* is equal to 1 then there is the most diversity.

Kohavi-WolpertVariance

The Kohavi-Wolpert Variance was introduced by Kohavi and Wolpert [23], and then Kuncheva and Whitaker presented a modification in [27]. In this measure, the diversity is lower if the value of KW is higher.

$$KW = \frac{1}{N \times L^2} \times \sum_{j=1}^{N} Y(Zj) \times \big(L - Y(Zj)\big) donde Y(Zj) = \sum_{i=1}^{L} Yi,j \quad (6)$$

Measurement of Inter-rater Agreement

The Measurement of Inter-rater Agreement was presented in [16].In this measure the diversity is lower when the k value is higher. The k is calculated by:

$$K = 1 - \frac{\frac{1}{L} \times \sum_{j=1}^{N} Y(Zj) \times \left(L - Y(Zj)\right)}{N \times (L-1) \times p \times (1-p)} \quad (7)$$

Where the last term is the measure of Kendall concordance and p is the mean of the accurate in the individual classification, which has the following formula:

$$p = \frac{1}{N \times L} \times \sum_{j=1}^{N} \sum_{i=1}^{L} Yj, i \quad (8)$$

Coincident Failure Diversity

The Coincident Failure Diversity is enunciated by Partridge y Krzanowski [33], this measure takes into account the instances where all the classifiers coincide.

$$CFD = \begin{cases} 0 & sipo = 1 \\ \frac{1}{1-po} \times \sum_{i=1}^{L} \frac{L-i}{L-1} \times pi \ , sipo < 1 \end{cases} \quad (9)$$

This measure has a minimum value of zero when all the classifiers are corrector incorrect, at the same time. The maximum value is one when at least one classifier is incorrect in any random object. In the formula p_i is the probability that $Y=i/L$ and L is the number of classifiers.

Distintic Failure Diversity

The Distintic Failure Diversity was also enunciated by Partridge y Krzanowski [32], as an improvement of the previous measure.

$$DFD = \begin{cases} 0 & siti = 0 \\ \sum_{i=1}^{L} \frac{L-i}{L-1} \times tisiti < 0 \end{cases} \quad (10)$$

Where *ti is* the number of *i* fails divided by total distinct fails, and L is the number of classifiers.

As a general rule, we may notice that these measures are more computationally complex than pairwise measures; the latter are simpler and the results lend themselves to an easier interpretation given their mathematic formulation. In this paper, we have confined ourselves to the pairwise measures given the previously mentioned considerations; in particular, we will use the Double Fault Measure in our simulations because is one of the most simple and more easy to interpretation.

It is in the presence of a combinatorial optimization problem because there are multiple variants of classifiers bases and variants of combination of diversity measures and precisely the metaheuristics are used for this, among metaheuristics are GA (Genetic Algorithms), ACO, PSO (Particle Swarm Optimization), etc. The main reason behind the use of ACO is its strong ability to solve combinatorial optimization problems intertwined and the existence of a previous paper where it already was modeled our problem using the metaheuristic GA.[8]

Next are described the ACO metaheuristic and its most popular algorithmic variants.

Ant Colony Optimization

The ACO metaheuristic is a stochastic search method originally designed in the combinatorial optimization problems; this method was invented by Marco Dorigo and draws inspiration from a colony of agents (ants) [13]. Real ants in nature search for food in the random proximity of their nest. Once the ants have found a food source, they assess this source according to its quality and quantity. In the path back to the nest, they lay a chemical substance named pheromone on the ground in order to guide the rest of the colony to the food source [12]. Therefore, ACO is a fully constructive model where each ant incrementally builds a candidate solution to the problem by exploring a construction graph in a step-by-step fashion.

More specifically, each artificial ant goes from one state (graph vertex/node) to another during the search process. The solution is then a sequence of moves. The preference of movement depends on two values associated to the link (graph edge) between these two nodes:

- The artificial information τ_{ij} is directly based on the pheromone trails and the ants iteratively update it during the algorithm execution.

- The heuristic information η_{ij} denotes the preference of traversing that edge. This problem-specific piece of knowledge often remains unchanged throughout the algorithm execution, so it must be carefully estimated beforehand.

From the perspective of the Kemeny ranking problem, Equation (11) denotes the probability of accepting the $j-$th state (i.e., elements to be ordered) at the ith position of the candidate ranking. $\mathcal{N}_i^k$ is the set of unvisited states for the kth ant, while α and β are two parameters used for controlling the influence exercised by the pheromone trails and the heuristic information, respectively, over the transition probability.

$$P_{ij}^k(t+1) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{r \in \mathcal{N}_i^k} [\tau_{ir}(t)]^\alpha [\eta_{ir}]^\beta}, j \in \mathcal{N}_i^k \quad (11)$$

After the iterative solution construction process is completed, all pheromone trails are updated using the solutions built by the agents (ants). In the first stage, pheromone evaporation takes place, thus uniformly reducing the amount of pheromone laid on all trails by a certain quantity. Subsequently, one or more solutions are used to increase the pheromone value of the trails included in these solutions. The pheromone update scheme is a pivotal step in any ACO-based algorithm. Essentially, most ACO variants mainly differ in the strategy used for updating the pheromone trail at each iteration.

Ant Systems

The Ant System (AS) was the first proposed ACO algorithm [14]. In AS, the pheromone trails are updated once all ants have completed their tours. As a first step, all pheromone trails are uniformly evaporated using a constant factor $0 < \rho < 1$. After that, each ant deposits a quantity of pheromone $\Delta\tau_{ij}$ on those graph edges that are part of its solution. It should be mentioned that the value $\Delta\tau_{ij}$ is calculated according to the quality of the solution found by the k^{th} ant. The following equation summarizes both steps, where ρ denotes the evaporation rate, whereas P is the number of ants in the colony.

$$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t) + \sum_{k=1}^{P} \Delta\,\tau_{ij}^k \quad (12)$$

On the edges that are not regularly chosen by the ants, the associated pheromone levels will gradually dwindle with the number of iterations, whereas edges often selected by the ants will see their pheromone level reinforced, hence making them more likely to be picked in future iterations. However, more comprehensive simulations reported in [29] is described that better results could be attained if only the global-best solution was used for updating the pheromone trails instead of using all ants in the colony. Notice that the unlimited accumulation of pheromone on the most promising edges can to produce stagnation in the search.

Ant Colony Systems

Ant Colony System (ACS) was devised to improve the AS method by exploiting the global-best solutions found by the ants during the search stage [39]. As result, the algorithm enhances the exploitation features of the ants when they build a solution instead of exploring new areas of the solution space. This goal is achieved through three mechanisms: (1) a strong elitist strategy for updating pheromone trails; (2) a rule for updating pheromone trails during the search phase, and (3) a pseudo-random transition probability rule.
Equation (13) formalizes the strategy for updating the pheromone trails, where τ_{ij}^* denotes the pheromone amount associated with the ant having better heuristic value. It means that the evaporation step takes place across all edges as in AS, but the updating process only occurs in the tour discovered by the best individual.

$$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t) + \rho\tau_{ij}^*(t) \quad (13)$$

In order to fully exploit the best knowledge elicited by the ants in their journey, ACS also introduces a pseudo-random proportional rule (see Equation (14)). More specifically, a random decision is made with probability q_0 to move to the node maximizing the product of the pheromone trail and the heuristic information; otherwise ACS will adopt the standard decision rule featured by AS. The value q_0 is a parameter that should be set by the expert a priori; when it is close to 1, exploitation is favored over exploration.

$$j = \underset{r \in \mathcal{N}_i^k}{\operatorname{argmax}} \left\{ \left[\tau_{ij}(t) \right]^{\alpha} \left[\eta_{ij} \right]^{\beta} \right\} \; if \; q \leq q_0 \tag{14}$$

Finally, in the ACS model, the ants use an additional rule for updating the pheromone trails when they are building the candidate solution, as displayed in Equation (15). This approach has the same effect of decreasing the probability of selecting the same path for all ants; it thus fights the stagnation problem present in AS given that it introduces a balance between exploitation and exploration.

$$\tau_{ij}(t+1) = (1-\rho)\, \tau_{ij}(t) + \rho \tau_{ij}(0) \tag{15}$$

MAX MIN Ant System

The MAX-MIN Ant System (MMAS) was specifically developed to promote a stronger exploitation of the solutions and therefore avoid falling into a stagnation state [39]. In a nutshell, we could define a stagnation state as the situation where ants construct the same solution over and over and the exploration eventually stops. This model has the following features. Equally, to ACS, a strong elitist strategy regulates the agent allowed to update the pheromone trails. It could be the ant having the best solution so far, or the one with the best solution in the current iteration. Second, all pheromone trails are limited to the range $[\tau_{\mathrm{MIN}}, \tau_{\mathrm{MAX}}]$. If $\tau_{\mathrm{MIN}} > 0$ for all solution components, then the probability of picking a specific state will never be zero, which avoids stagnation configurations [39]. As a final point, pheromone trails are initialized with τ_{MAX} to ensure further exploration of the search space at the beginning of the optimization phase.
The three variants described above are implement.

3. BUILDING A MULTI-CLASSIFIER SYSTEM WITH ACO

Next, we address the problem of building an "optimal" MCS by using the ACO metaheuristic. Here the optimality criterion refers to the accuracy of the final system and to the number of selected base classifiers. We certainly cannot ensure that our proposal will always find the global optimum because the selected optimizer (ACO) may converge to a suboptimal solution. Nevertheless, we adopted the ACO metaheuristic since it is capable of finding near-optimal solutions in a reasonably short execution time while not imposing any constraints on the objective function (e.g. continuity, differentiability, convexity or gradient information) which are seldom known beforehand.
Let us assume a family of classifiers $\Phi = \{\phi_1, \ldots, \phi_{\mathrm{i}}, \ldots, \phi_{\mathrm{N}}\}$ where each classifier has an associated classification error $\mathrm{E}_{\wp}(\phi_{\mathrm{i}})$, where $\wp$ denotes the classification problem to be solved. The issue of building an MCS $\mathcal{M}$ consists of finding a subset of these classifiers $\Phi' \subset \Phi$ with maximal diversity such that $E_{\wp}(\mathcal{M})$ tends to the minimal error. Notice that $|\Phi'|$ must necessarily be strictly lower than $|\Phi|$, otherwise the solution will be the trivial one (e.g., all individual classifiers are included in the ensemble). Furthermore, the model needs to fulfill another constraint: $E_{\wp}(\mathcal{M}) < E_{\wp}(\phi_*)$ where $\phi_* = \mathrm{argmin}_{\mathrm{i}\in 1..\mathrm{N}}\{\phi_{\mathrm{i}}\}$ is the best classifier included in the ensemble. This constraint ensures that the ensemble improves the classification performance over any of its constituent classification schemes.
From the optimization point of view, candidate solutions for our problem can be encoded as a binary vector where the state "1" at the i-th dimension means that the classifier ϕ_{i} will be included in the ensemble whereas the state "0" indicates that ϕ_{i} will not be included in $\mathcal{M}$. Therefore, the proposed model $\mathrm{P}_{\mathrm{ij}}^{\mathrm{k}}$ represents the probability of assigning the state $\mathrm{S_j} \in \{0,1\}$ to the i-th dimension (i.e., the probability of including the i-th classifier). Equation (16) unveils the objective function to be minimized during the search process undertaken by ACO, where X is the candidate solution, $\mathcal{M}_X$ denotes the ensemble computed from X and $E_{\wp}(\mathcal{M}_X)$ denotes its classification error. In this formulation, a factor $0 < \omega < 1$ is introduced in order to control the relevance that the expert bestows to the system accuracy with respect to the ensemble cardinality, that is, the number of selected classifiers.

$$\text{minimize } F(X) = \omega E_{\wp}(\mathcal{M}_X) + (1-\omega)\|X\|_{L_1}(|\Phi|)^{-1} \tag{16}$$

During the search process, those solutions that have an error rate greater than the error rate associated to the best classifier included in the ensemble (i.e. $E_{\wp}(\mathcal{M}_X) > E_{\wp}(\phi_*)$) must be penalized by a positive factor. Besides, two infeasible solutions may induce different errors, and therefore the penalization strategy should consider this fact when modifying the objective function $\mathrm{F(X)}$. For example, let us consider two different solutions $\mathrm{X_1} = (\mathrm{x_1^1}, \mathrm{x_2^1}, \ldots, \mathrm{x_N^1})$ and $\mathrm{X_2} = (\mathrm{x_1^2}, \mathrm{x_2^2}, \ldots, \mathrm{x_N^2})$ which encode the $\mathcal{M}_{X_1}$ and $\mathcal{M}_{X_2}$ ensembles, respectively. If the induced errors $E_1 = \left(E_{\wp}(\mathcal{M}_{X_1}) - E_{\wp}(\phi_*)\right)$ and $E_2 = \left(E_{\wp}(\mathcal{M}_{X_2}) - E_{\wp}(\phi_*)\right)$ are greater than zero, then $\mathrm{X_1}$ and $\mathrm{X_2}$ are both considered infeasible. However, it is unlikely that $\mathrm{E_1 - E_2 = 0}$. This suggests that we should not penalize both solutions with the same positive value; instead, we should penalize each solution according to their error $\mathrm{E_i - E_{\wp}(\phi_*)}$. In this paper, we make use of a dynamic penalization function $\mathrm{P(X)}$ that takes into account the induced error as follows:

$$P(X) = \frac{1 + \Delta F(X)\left[E_{\wp}(\mathcal{M}_X) - E_{\wp}(\phi_*)\right]}{1 + E_{\wp}(\mathcal{M}_X)} \tag{17}$$

Another relevant aspect to be considered when solving a combinatorial optimization problem via any ACO algorithm is the estimation of the heuristic information. This component allows improving the search, even in large search spaces, as it corresponds to problem-specific knowledge that is incorporated into the state transition probability rule. In this study, we put forth three different heuristics that will be detailed next. The first one assumes that classifiers inducing large diversity are likely to improve the overall ensemble performance, even when they are not the most accurate ones. Equation (18) formalizes this reasoning when estimating the heuristic matrix $\eta_{\mathrm{N}\times 2}$, where $\mathcal{D}$ represents the ensemble diversity measure under consideration, $\{\phi_{\mathrm{i}}\}_{\mathrm{i=1}}^{\mathrm{L}}$ denotes the family of base classifiers and the index j indicates the status of the i-th classifier (0 = excluded, 1 = included) in the ensemble.

$$\eta_{ij}^{\mathcal{D}} = \begin{cases} \mathcal{D}\left(\{\phi_i\}_{i=1}^{L} - \{\phi_i\}\right), j = 0 \\ \mathcal{D}\left(\{\phi_i\}_{i=1}^{L}\right), j = 1 \end{cases} \tag{18}$$

The second heuristic assumes that classifiers that report lower classification errors are more likely to improve the ensemble performance even when the diversity among them is not maximal (see Equation (19)). In other words, the probability of picking a specific classifier is subject to its individual quality. Based on this strategy, we could conclude that the i-th classifier will be excluded from the ensemble $\mathcal{M}$ with conditional probability $\mathrm{P_j(\phi_i | E_{\wp}(\phi_i))}$ while the probability of having it as part of the ensemble is given by the conditional probability $\mathrm{P_j(\phi_i | 1 - E_{\wp}(\phi_i))}$. Here we assume that $\mathrm{E_{\wp}(\phi_i)}$ denotes the classification error achieved by the i-th individual classifier.

$$\eta_{ij}^{E} = \begin{cases} E_{\wp}(\phi_i), j = 0 \\ 1 - E_{\wp}(\phi_i), j = 1 \end{cases} \tag{19}$$

The last heuristic strategy is actually a combination of the previous diversity-based heuristic and the accuracy-based heuristic. Being more explicit, we assume that classifiers having lower classification errors and inducing higher diversity are more likely to improve the system performance, and thus they should be included in the ensemble. Equation (20) formalizes this heuristic for some diversity measure $\mathcal{D}$, where $\{\phi_{\mathrm{i}}\}_{\mathrm{i=1}}^{\mathrm{N}} - \{\phi_{\mathrm{i}}\}$ denotes the family of base classifiers excluding the classifier ϕ_{i}, which means that the state $\mathrm{j = 0}$ will be observed at the i-th dimension. It should be highlighted that in ACO, states are actually selected based on the combination of both the heuristic information and the pheromones trails, so we must select the relevance of each component.

$$\eta_{ij}^{\mathcal{D}+E} = \begin{cases} \mathcal{D}\left(\{\phi_i\}_{i=1}^{L} - \{\phi_i\}\right) + E_{\wp}(\phi_i), j = 0 \\ \mathcal{D}\left(\{\phi_i\}_{i=1}^{L}\right) + \left[1 - E_{\wp}(\phi_i)\right], j = 1 \end{cases} \tag{20}$$

Based on the above configuration, the ACO-based optimizer should be capable of finding a subset of classifiers $\Phi' \subset \Phi$ with a desired behavior (i.e., high classification rate for difficult pattern recognition problems). In order to evaluate our model, we conducted an empirical analysis in the following section across a set of benchmark data sets, which are often employed when assessing the performance of new classifiers.

4. RESULTS AND DISCUSSION

To empirically validate the methodology proposed in this study, we designed several experiments to compare the performance of our ACO-based MCS building, with some of the most well-known state-of-the-art classification models. These models were taken from the WEKA tool (Waikato Environment for Knowledge Analysis) [19] and are listed below:

- Alternative Decision Tree: This model, also known as ADTrees, generates an Alternative Decision Tree. The WEKA version only supports binary class problems. The number of boosting iterations needs to be manually tuned to suit the dataset and the desired complexity/accuracy tradeoff. The tree induction process has been optimized and heuristic search methods have been introduced to speed up the learning [18].
- J48: This algorithm, described in [36], aims at generating a pruned or unpruned C4.5 decision tree. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by J48 can be used for classification purposes. J48 builds decision trees from a set of labeled training data using the concept of information entropy to decide which attribute should be chosen for splitting the dataset at that level of the tree.
- Logistic: This classifier, also known as Logistic function, is a classifier that uses a multinomial logistic regression model with a high estimate [28].
- K-Nearest Neighbor with k = 1. Also known as IB1, this simple classifier uses the normalized Euclidean distance to find the training instance closest to the given test instance, and predicts the same class that this training instance bears. If more than one training instance lies at the same distance from the test instance, the first one found is used [1].
- *Naïve Bayes*: This classifier is rooted on applying Bayes' theorem with strong (naïve) independence assumptions among the features. It is a highly scalable classifier; require a number of parameters that is linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can do by evaluating a closed-form expression, which takes linear time, rather than through an expensive iterative approximation as used for many other types of classifiers. Naïve Bayes employs a class estimator; the numerical estimator and precision values are chosen based on an analysis of the training data [22].
- *Multi-Layer Perceptron (MLP)*: This classifier relies on a feedforward neural network using the backpropagation training algorithm to classify instances [3]. The MLP network can built by hand, learned from data or both. The neurons in this network all make use of sigmoidal activation functions, except for when the class is numeric [21].

The multi-classifier systems are generated by using the voting mechanism, also stems from WEKA, were combined multiple base classifiers of different types. Two strategies are considered when combining the outputs (classifications) of the base algorithms in the MCS: average and majority vote. All experiments involving the ACO metaheuristic are implemented the following parametric configuration: 50 iterations, evaporation constant $\rho = 0.9$, ACS' phi = 0.9 and $q_0 = 0.7$. The original training data is divided in two subsets: 66% of the instances used for training and the rest for evaluation. All classification techniques were run against 10 data sets coming from the Machine Learning Repository at the University of California Irvine [4]; these data sets are outlined in Table 3.

Databases	Nominal Features	Numeric Features	Classes	Cases	Distribution by classes
Australian	5	9	2	690	383-307

Breast Cancer	9	0	2	683	444-239
Diabetes	0	8	2	768	500-268
Echocardiogram	1	11	2	132	79-53
Heart Stlatlog	0	13	2	270	150-120
Hepatitis	13	6	2	155	32-123
house-votes	16	0	2	435	201-99
German Credit	13	7	2	1000	300-700
Pro Ortology	0	11	2	4314	1438-2876
Tic Tac Toe	9	0	2	958	626-332

The experiments are geared to identifying what ACO version (AS, ACS or MMAS) yields better results as well as determining the worth of the three heuristic functions proposed in Section 4.

Experiment 1: Ensemble Size per Heuristic Function

Pearson's Chi-square statistical test [34] is applied to sets of categorical data to evaluate how likely it is that any observed difference between the sets arose by chance; this test is suitable for unpaired data from large samples. It tests a null hypothesis stating that the frequency distribution of certain events observed in a sample is consistent with a particular theoretical distribution. The events considered must be mutually exclusive and have the total probability add up to 1. A common case for this rule is where the events each cover an outcome of a categorical variable.
To apply the above test, we leaned on the SPSS software. Two groups were created to represent the number of classifiers included in the solutions: in the first group, all solutions containing two or three classifiers were isolated whereas the second group denotes the solutions with more than three classifiers. The maximum ensemble size is six since that is the number of classifiers included in the experiment.
The results reveal the existence of statistical differences at the 95% significance level among the three heuristics used: diversity, accuracy and hybrid (the obtained significance value was below 0.05). Table 4 shows that the diversity heuristic exhibits the largest number of cases in the first group (58) and the smaller number of cases in the second group (2), i.e., solutions having a smaller number of base classifiers (therefore with reduced ensemble complexity).

		Heuristic			Total
		Diversity	Accuracy	Hybrid	
Number of Classifiers	1st group	58	39	53	150
	2nd group	2	21	7	30
Total		60	60	60	180

Table 4. Results of Chi-square test with heuristics

To confirm the above claim, we report in Figures 1-6 the ensemble size produced by the three ACO variants (AS, ACS and MMAS algorithms) under both voting criteria (average and majority) on a subset of the data sets under consideration according to the three distinct heuristic strategies: diversity, accuracy and hybrid. generally larger. The accuracy heuristic produces the largest (and hence most complex) ensembles, so it does not fare well along this indicator.

Experiment 2: Ensemble Accuracy per Heuristic Function

The Kruskal-Wallis test is applied to determine the heuristic strategy that provides the best results in terms of ensemble classification accuracy. The Kruskal–Wallis one-way analysis of variance by ranks [25] is a nonparametric method for testing whether samples originate from the same distribution. This test is used to compare two or more independent samples that may have different sizes; it extends the Mann–Whitney U test to more than two groups. The parametric equivalent of the Kruskal-Wallis test is the one-way analysis of variance (ANOVA). When rejecting the null hypothesis of the Kruskal-Wallis test, then at least one sample

stochastically dominates at least one other sample. The test does not identify where this stochastic dominance occurs or for how many pairs of groups the stochastic dominance is obtained.

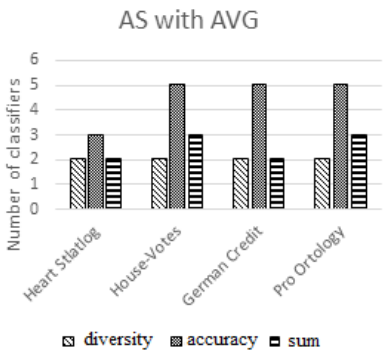

Fig. 1: Ensemble size per heuristic function under the AS algorithm and average voting.

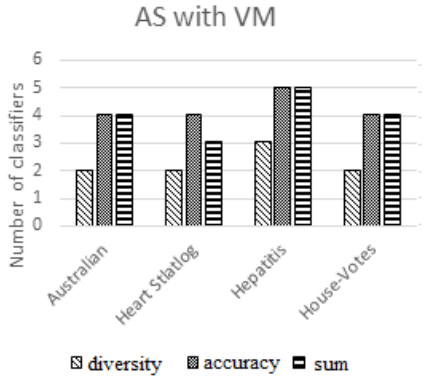

Fig. 2: Ensemble size per heuristic function under the AS algorithm and majority voting.

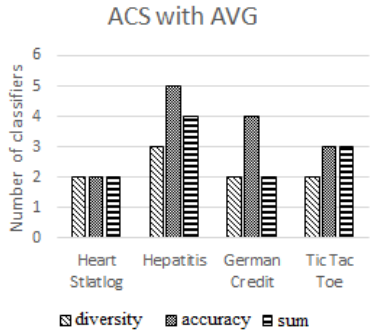

Fig. 3: Ensemble size per heuristic function under the ACS algorithm and average voting.

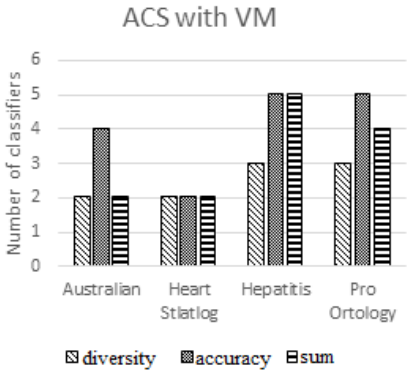

Fig. 4: Ensemble size per heuristic function under the ACS algorithm and majority voting.

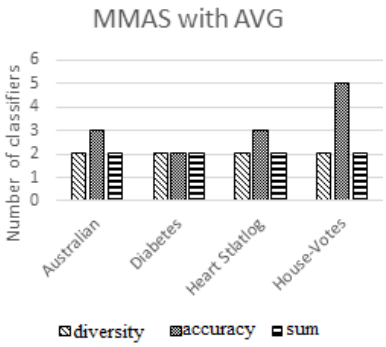

Fig. 5: Ensemble size per heuristic function under the MMAS algorithm and average voting.

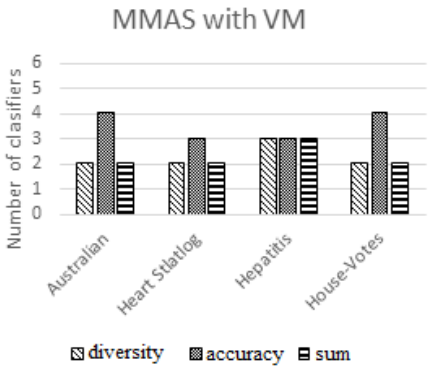

Fig. 6: Ensemble size per heuristic function under the MMAS algorithm and majority voting

	Heuristics	N	Average Range
Accuracy	Diversity	60	97.00
	Accuracy	60	79.04
	Sum	60	94.00
	Total	180	

Table 5. Results of Kruskal-Wallis test with the three ACO heuristics

The previous test shows that there are no significant differences among the three heuristics at the 95% significance level, although Table 5 portrays that the higher value of the average range belongs to the diversity heuristic.

One may notice from Figures 1-6 that the heuristic strategy that leads to ensembles that are more compact is the diversity heuristic. In some of the data sets, the hybrid heuristic is able to produce an ensemble of the same (smallest) size as the one induced by diversity; however, in the rest of the data sets the ensemble size is

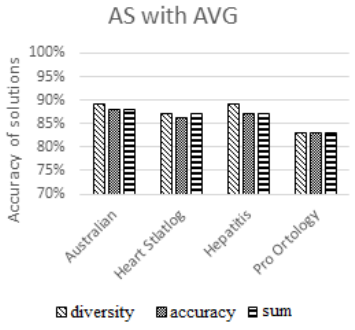

Fig. 7: Ensemble accuracy per heuristic function under the AS algorithm and average voting.

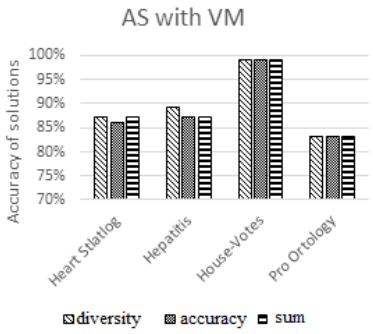

Fig. 8: Ensemble accuracy per heuristic function under the AS algorithm and majority voting.

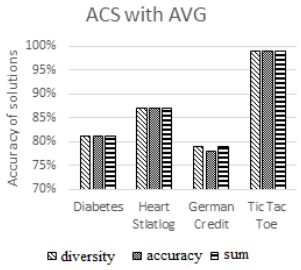

Fig. 9: Ensemble accuracy per heuristic function under the ACS algorithm and average voting.

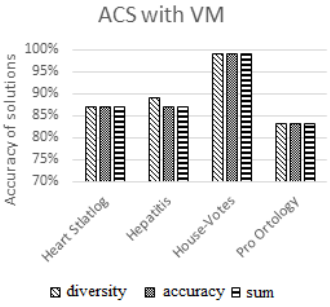

Fig. 10: Ensemble accuracy per heuristic function under the ACS algorithm and majority voting.

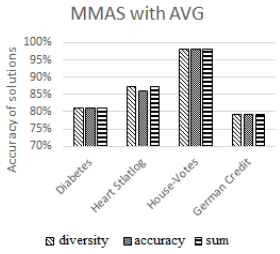

Fig. 11: Ensemble accuracy per heuristic function under the MMAS algorithm and average voting.

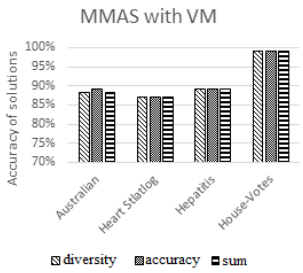

Fig. 12: Ensemble accuracy per heuristic function under the MMAS algorithm and majority voting.

Figures 7-12 report the ensemble classification accuracy achieved in a subset of the data sets under study by the three ACO algorithms with the three heuristic functions and the two voting schemes.
The conclusion drawn from the results in Figures 7-12 is that the diversity heuristic seems slightly more effective in securing higher ensemble classification rates, although the two other heuristics closely follow. This visual inspection confirms the results of the Kruskal-Wallis test.
Based on the outcomes of Experiments 1 and 2 (showing that the diversity heuristic function is to be preferred over the other two given its superior ensemble size and accuracy rates), we will stick to this heuristic function as we move forward to assess the three ACO variants in the remainder of this section.

Experiment 3: Comparing the three ACO Algorithms

To investigate the performance of the three ACO techniques in terms of the complexity of the produced MCSs, we resorted again to the Chi-square test as done in Experiment 1. That is, for each ACO algorithm, we created two groups based on the size of the ensembles therein (group 1: 2 or 3 classifiers; group 2: > 3 classifiers). Then we applied the Chi-square test to the frequency distribution in both groups to determine whether it is reasonable to believe that they originate from the same distribution or not. The test revealed the existence of significant differences at the 95% significance level between the two groups respect of three ACO variants. In Table 6, it can be seen that the MMAS variant shows more cases in the first group, i.e., the solutions have fewer classifiers using this variant.

		Variants			Total
		AS	ACS	MMAS	
Number of Classifiers	1st group	44	49	57	150
	2nd group	16	11	3	30
Total		60	60	60	180

Table 6. Results of Chi-square test with variants

Additionally, the Kruskal-Wallis test is applied to determine the ACO version that offers the best results according to the ensemble accuracy; this test reveals that there are not significant differences detected at the 95% significance level, although Table 7 indicates that MMAS yields a slightly higher average range over the two other ACO variants.

	Variants	N	Average Range
Accuracy	AS	60	97.00
	ACS	60	96.04
	MMAS	60	98.00
	Total	180	

Table 7. Results of Kruskal-Wallis test with the three ACO variants

4.1 CASE STUDY: PREDICTING HIGH BLOOD PRESUARE IN CHILDREN

High blood pressure (HBP) is a common condition that affects the health of individuals worldwide, adults and children alike, with the potential to cause extensive damage. Due to its asymptomatic nature it has been

termed a "silent epidemic" because usually there are no clear signs demonstrating its presence [2]. Nowadays, the concept “Systemic Arterial Hypertension” is very common in our society and it has been identified as cardiovascular risk factor. However, not everybody thinks in this risk for the children’s welfare. At the same time, this disease is a major risk factor for other brain, heart or kidney-related ailments. Prevention of cardiovascular disease is not limited to adults but must start with childhood. It has been shown that hypertension in children is an independent risk factor for hypertension in adulthood and that it is associated with early markers of cardiovascular disease such as left ventricular hypertrophy, thickness of the intima-media, atherosclerosis and diastolic dysfunction.

In this study, the sample consisted of 680 children apparently healthy between 8-12 years of age, of both genders, from 4 primary schools in the city of Santa Clara. The data were supplied by the PROCDEC[2] project; it consists of 680 children whose medical history was access with the consent of their parents. From the analysis of the predictive attributes, patients are categorized into two groups according to the risk of the disease: risky patients and non-risky patients. A brief statistical summary is presented in Table 8.

	N	Minimum	Maximum	Media	Typical Des.
Current weight (kg)	680	20.50	74.00	35.2545	8.92034
Size (cm)	680	113.00	161.00	137.6217	8.19466
Waist circumference (cm)	680	47.00	104.00	64.2578	8.72743
Hip circumference (cm)	680	52.00	106.00	71.8132	8.48275
Waist hip index	680	0.65	1.25	0.8949	0.05754
TA Higher systolic Member	680	81	150	114.46	12.811
TA Diastolic Superior Member	680	49	99	67.00	7.711
TA Systolic 5min	680	73	152	111.72	12.254
TA Diastolic 5min	680	46	95	66.31	8.133
TA Systolic 10min (before 15min)	680	70	149	109.30	12.486
TA Diastolic 10min (before 15min)	680	45	94	64.67	8.061
TA first day Mean Systolic	679	77.33	146.00	111.7712	11.45872
TA first day Mean Diastolic	679	43.33	88.67	65.9686	6.77948
TA second day basal Systolic	674	82	150	112.12	10.928
TA second day basal Diastolic	674	44	94	65.69	7.469
TA second day Systolic P sustained weight	674	78	166	120.33	12.954
TA second day Diastolic P sustained weight	674	48	116	75.12	9.924
PAM2d	674	58.00	129.67	90.2028	9.79314
TA third day basal Systolic	655	81	144	112.28	10.503
TA third day basal Diastolic	655	40	110	66.00	7.638
TA third day Systolic P sustained weight	654	83	160	120.31	12.190
TA third day Diastolic P sustained weight	654	49	112	75.34	10.186
PAM3d	654	62.33	120.00	90.2798	9.76576

Table 8. Descriptive statistics of HTA data set

The experiments are executed with the diversity heuristic and the MMAS algorithm given their promising results in the previous experiments. Three different sets of classifiers are used with 6, 12 and 18 classifiers respectively. The best individual accuracy was achieved by MLP with 91% in the three sets. In the solutions, the positions with one means that the classifier in this position is include.

Classifiers	**AVG**			
	Solution	**Goal Function**	**Accuracy**	**Diversity**
6	000011	0.10	0.92	0.96
12	010010000000	0.09	0.92	0.94
18	0101000110000000100	0.09	0.95	0.95

Table 9. Results obtained in experiments

We can see in Table 9 that the best individual accuracy was exceeded in all cases, reaching a maximum of 95% with five classifiers. More solutions were obtained; however, we have displayed only the ones with a small ensemble size in each of the three classifier sets. The classifier with the highest individual accuracy is included in the ensemble for most of the solutions; its corresponding position is highlighted in red. Note that the diversity in the found combinations of classifiers was relatively high.

In addition, below is shown a comparison with the results obtained by the built multi-classifier with ACO and

[2] Project PRODEC: Projection Electronic Development Center to the Community, Universidad Central “Marta Abreu” de Las Villas, Santa Clara, Cuba.

multi-classifier models mentioned in the introduction of the document, taking into account the six classifiers mentioned before, at the beginning in this section. In this comparison, was chose each individual classifier combined in the built multi-classifier as individual model in Bagging and Boosting (AdaBoost variant was used), because they working in different ways, as was explained in the introduction (the analysis with MLP as individual classifier in these models not is recommendable but was done any way). Stacking allows combine different individual classifiers at the same time.

Multi-classifier	Accuracy
Bagging (ADTree)	87.40
Bagging (J48)	87.44
Bagging (Logistic)	89.61
Bagging (IB1)	79.22
Bagging (Naives Bayes)	83.54
Bagging (MLP)	89.04

Multi-classifier	Accuracy
AdaBoost (ADTree)	87.44
AdaBoost (J48)	87.87
AdaBoost (Logistic)	89.60
AdaBoost (IB1)	80.10
AdaBoost (Naives Bayes)	87.87
AdaBoost (MLP)	89.04

Table 10. Results obtained with Bagging and Boosting using each individual classifier

In case of Stacking was combined the classifiers at the same time, the accuracy obtained was 88.02. Below is presented Table 11 with the best result of Bagging and boosting, Stacking and also Random Forest as other multi-classifier model that tends to offer good results [15], for last, our MCS with ACO.

Multi-classifiers	Accuracy	Execution time
Bagging	89.61% (0.89)	3.6 seconds
Boosting (AdaBoost)	89.60% (0.89)	3.5 seconds
Stacking	88.02% (0.88)	5 seconds
Random Forest	89.91% (0.89)	3.3 seconds
MCS with ACO	**92% (0.92)**	8 seconds

Table 11. Comparison obtained between multi-classifier models and MCS with ACO.

We can see that the best result is obtained with MCS with ACO, where also is guaranteed diversity between classifiers in the system, and the different in the execution times is not considerable, it remembers that they working in different ways.

5. CONCLUSIONS

In this paper, we shown how to build multi-classifier systems using ACO. In particular, three popular variants of the ACO metaheuristic algorithm have been tailored to this end. We describe the representation of the search space using a binary vector, the objective function and the three heuristic functions: diversity, accuracy and a hybrid one. The experimental evidence over 10 UCI MLR data sets, supported by a proper statistical validation, confirming that the principal discovered is that the best results are obtained using the z and the diversity heuristic, which guarantees the existence of diverse classifiers sets in the solutions what achieve high classification rates.

A real-world application was also reported that investigates the performance of ACO-based MCSs to predict high blood pressure in children. In this scenario, the best individual classifier's accuracy is exceeded in 4%. In addition, is shown a comparison between the built multi-classifier with ACO and others multi-classifier models mentioned in the introduction of the document, where the built multi-classifier with ACO obtains the best result.

RECEIVED: MAY, 2016
REVISED: NOVEMBER, 2016

REFERENCES

[1] AHA, D. W. A. K., DENNIS AND ALBERT, MARC K (1991): Instance-based learning algorithms. **Machine learning**, vol. 6 (1), pp. 37-66.

[2] ARMARIO, P. A. D. R., RAQUEL HERNÁNDEZ AND MARTÍN-BARANERA, MONTSERRAT (2002): Estrés, enfermedad cardiovascular e hipertensión arterial. **Medicina Clínica**, vol. 119, pp. 23--29.

[3] BAUM, E. B. (1988): On the capabilities of multilayer perceptrons. **Journal of complexity**, vol. 4 (3), pp. 193--215.

[4] BLAKE, C. and MERZ, C. J. UCI Repository of machine learning databases. Irvine, University of California CA. [Online]. Available: http://archive.ics.uci.edu/ml/

[5] BONET, I. and MONTERO, P. (2013): Classifier Ensemble Based on Feature Selection and Diversity Measures for Predicting the Affinity of A(2B) Adenosine Receptor Antagonists. **Journal of Chemical Information and Modeling**, vol. 53, nro 12, pp. 3140--3155.

[6] BONET, I., RODRÍGUEZ, A. , GARCÍA M. M. Y GRAU R. (2012): Combinación de clasificadores para Bioinformática. **Computación y Sistemas**, vol. 16, Nro 2, pp. 191-201.

[7] BREIMAN, L. (1996): Bagging predictors. ***Machine Learning***, vol. 24, Nro 2**,** pp. 123-140.

[8] CABRERA, L., MORALES, A., CASAS, G., and MARTÍNEZ, Y. (2015): GENETIC ALGORITHMS WITH DIVERSITY MEASURES TO BUILD CLASSIFIER SYSTEMS. **Revista Investigación Operacional**, vol. 36, nro 3, pp. 206-224.

[9] COLETTA, L. F. S., HRUSCHKA, E. R., ACHARYA, A., and GHOSH, J. (2015): Using metaheuristics to optimize the combination of classifier and cluster ensembles. **Integrated Computer-Aided Engineering**, vol. 22, nro 3, pp. 229-242.

[10] CUNNINGHAM, P. and CARNEY, J. (2000): Diversity versus Quality in Classification Ensembles Based on Feature Selection. In: **11th European Conference on Machine Learning**, Trinity College, Dublin, pp. 109-116.

[11] DIETTERICH, T. G. (2000): Ensemble methods in machine learning. In: **Proceedings of the First International Workshop on Multiple Classifier Systems**, London, United Kingdom, pp. 1-15.

[12] DORIGO, M., BONABEAU, E., THERAULAZ, G. (2000): Ant algorithms and stigmergy. **Future Generation Computer Systems.**, vol. 16, pp. 851--871.

[13] DORIGO, M., DICARO, G., and GAMBARDELLA, L. (1999): Ant algorithms for discrete optimization. **Artificial life**, vol. 5, nro 2, pp. 137-172.

[14] DORIGO, M., MANIEZZO, V., and COLORNI, A. (1996): Ant system: optimization by a colony of cooperating agents. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, vol. 26, nro 1, pp. 29-41.

[15] FERNÁNDEZ-DELGADO, M., CERNADAS, E., BARRO, S., and AMORIM, D. (2014): Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?. **Journal of Machine Learning Research** vol. 15, pp. 3133-3181

[16] FLEISS, J. L. (1981): **Statistical Methods for Rates and Proportions**. John Wiley & Sons, New York.

[17] FREUND, Y. and SCHAPIRE, R. E. (1995): A desicion-theoretic generalization of on-line learning and an application to boosting. In: **European Conference on Computational Learning Theory**, pp. 23-37.

[18] FREUND, Y. A. M., LLEW (1999): The alternating decision tree learning algorithm. In: **Proceeding of the Sixteenth International Conference on Machine Learning**, Bled, Slovenia, pp. 124--133.

[19] GARNER, S. R. A. O. (1995): Weka: The waikato environment for knowledge analysis. In: **Proceedings of the New Zealand computer science research students conference**, New Zealand, pp. 57--64.

[20] GIACINTO, G., ROLI, F. (2001): Design of effective neural network ensembles for image classification purposes. **Image vision and computing journal**, vol. 19, pp. 699-707.

[21] GRAU, I., NÁPOLES, G. ,BONET, I. , GARCÍA M. M. (2013): Backpropagation through Time Algorithm for Training Recurrent Neural Networks using Variable Length Instances. **Computación y Sistemas**, vol. 17 (1), pp. 15-24.

[22] JOHN, G. H. A. L., PAT (1995): Estimating continuous distributions in Bayesian classifiers. In: **Proceedings of the Eleventh conference on Uncertainty in artificial intelligence**, San Mateo, pp. 338--345.

[23] KOHAVI, R. and WOLPERT, D. H. (1996): Bias Plus Variance Decomposition for Zero-One Loss Functions in Machine Learning. In: **Machine Learning: Proceedings of the Thirteenth International Conference**, Los Altos, California, pp. 275-283.

[24] KRAWCZYK, B. and WOŹNIAK, M. (2014): Diversity measures for one-class classifier ensembles. **Neurocomputing**, vol. 126, pp. 36-44.
[25] KRUSKAL, W. H. and ALLEN WALLIS, W. (1952): Use of ranks in one-criterion analysis of variance. **Journal of the American Statistical Association**, vol. 47 (260), pp. 583–621.
[26] KUNCHEVA, L. I. (2004): Diversity in Classifier Ensembles. In **Combining Pattern Classifiers: Methods and Algorithms**, 295-327. John Wiley & Sons, Inc., New Jersey.
[27] KUNCHEVA, L. I. and WHITAKER, C. J. (2003): Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. **Machine Learning**, vol. 51, pp. 181-207.
[28] LE CESSIE, S. A. V. H., JOHANNES C (1992): Ridge estimators in logistic regression. **Applied statistics**, vol. 41(1), pp. 191--201.
[29] MATOS, C. C. (2007): **Selección/identificación asistida por computadora de nuevos compuestos líderes con actividad anti-inflamatoria.** Tesis Doctoral, Universidad Central "Marta Abreu" de las Villas, Villa Clara.
[30] ORDOÑEZ, J., LEDEZMA, A., and SANCHIS, A. (2008): Genetic Approach for Optimizing Ensembles of Classifiers. In: **Proceedings of the Twenty-First International FLAIRS Conference**, pp. 89-94.
[31] PALANISAMY, S. and KANMANI, S. (2012): Classifier Ensemble Desing using Artificial Bee Colony based Feature Selection. **IJCSI International Journal of Computer Science Issues**, vol. 9, Issue 3, No 2, pp. 522-529.
[32] PARTRIDGE, D. and KRZANOWSKI, W. (1997): **Distinct failure diversity in multiversion software**. University of Exeter, United Kingdom.
[33] PARTRIDGE, D. and KRZANOWSKI, W. (1997): Software diversity: practical statistics for its measurement and exploitation. **Information and Software Technology**, vol. 39, pp. 707-717.
[34] PEARSON, K. (1900): "On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling". **Philosophical Magazine Series 5** vol. 50 (302), pp. 157–175.
[35] POLIKAR, R. (2006): Ensemble based systems in decision making. **Circuits and Systems Magazine, IEEE**, vol. 6, pp. 21-45.
[36] QUINLAN, J. R. (1993): ***C4.5: Programs for Machine Learning.*** Morgan Kaufmann. San Mateo California.
[37] SCHAPIRE, R. E. (1990): *The strength of weak learnability*. **Machine Learning**, vol. 5, pp. 197-227.
[38] SKALAK, D. B. (1996): The sources of increased accuracy for two proposed Boosting algorithms. In: **Proc. American Association for Arti Intelligence, AAAI-96, Integrating Multiple Learned Models Workshop**, pp. 120-125.
[39] STUTZLE, T. and HOOS, H. H. (2000): MAX--MIN ant system. **Future generation computer systems**, vol. 16, nro 8, pp. 889-914.
[40] TANG, E. K. A. S., PONNUTHURAI N AND YAO, XIN (2006): An analysis of diversity measures. **Machine Learning**, vol. 65 (1), pp. 247-271.
[41] WITTEN, I., FRANK, E. (2005): *Data Mining: Practical Machine Learning Tools and Techniques*. **San Francisco, Diane Cerra.**, vol.
[42] WOLPERT, D. (1992): *Stacked generalization.* **Neural Networks**, vol. pp. 5**,** 241-259.