

Collective Learning in Multi-Agent Systems Based on Cultural Algorithms

Juan Terán

Universidad de Los Andes, CEMISID
Mérida, Venezuela, 5101
carlostp@ula.ve

José L. Aguilar

Universidad de Los Andes, CEMISID
Mérida, Venezuela, 5101
aguilar@ula.ve

and

Mariela Cerrada

Universidad de Los Andes, CEMISID
Mérida, Venezuela, 5101
cerradam@ula.ve

Abstract

This paper aims to present a learning model for coordination schemes in Multi-Agent Systems (MAS) based on Cultural Algorithms (CA). In this model, the individuals (one of the CA components) are the different conversations that may occur in any multi-agent systems, and the coordination scheme learned is at the level of the way to perform the communication protocols into the conversation. A conversation can have sub-conversations, and the sub-conversations and/or conversations are identified with a particular type of conversation associated with a certain interaction patterns. The interaction patterns use the coordination mechanisms existing in the literature. In order to simulate the proposed learning model, we develop a computational tool called CLEMAS, which has been used to apply the model to a case of study in industrial automation, related to a Faults Management System based on Agents.

Keywords: Cultural Algorithms, Coordination, Multi-Agent Systems.

1 Introduction

A MAS is an agents community that interacts by using high level communication protocols and languages, to solve problems beyond its capabilities or knowledge [1]. This society of agents may face conflicts not only to communicate, but also by the use of resources, task allocations, etc. To handle these conflicts, there exist coordination mechanisms (CM), which solve these problems. The interactions between agents can be viewed as conversations, which in turn may have sub-conversations. To characterize these conversations and sub-conversations are used types of conversations (TCs), which defines the interaction patterns that can be performed by using well-known communication protocols [2, 3]. In this paper, a learning model based on Cultural Algorithm (CA) is proposed to optimize coordination schemes in a MAS. It aims to set up the particular CM in each one of their conversations. Then, in this sense, a set of coordination schemes that MAS can use, can be seen as the instantiation of the MAS with a particular configuration of CM in their conversations.

The CA can provide knowledge, since one of its main components is a common space of experiences providing the capacity for collective learning based on knowledge sharing. Some works related to the coordination of MAS are: in [3] it is studied the ad hoc coordination problem, that is, design autonomous agents, which are able to achieve optimal flexibility and efficiency in a MAS with no mechanisms for prior coordination. This work formally conceptualizes this problem by using a game-theoretic model, called the Stochastic Bayesian Game, in which the behavior of a player is determined by its private information. Based on this model, they derive a solution, called Harsanyi-Bellman Ad hoc coordination (HBA), which uses the concept of Bayesian Nash Equilibrium in a planning procedure to find optimal actions in the sense of Bellman Optimal Control. Other works that address the learning approach to coordinate MAS are [4, 5]. These works use one of the techniques more used in multi-agent learning, the Reinforcement Learning (RL). In [4], they propose a Bayesian model for optimal exploration in multi-agent

systems where the exploration costs are weighed with respect to their expected benefits, by using the notion of value of information. Unlike standard RL models, this model requires reasoning about how one's actions will influence the behavior of other agents. The estimated value of an action, given a current model, requires a prediction about how will be the influence of this action on the future actions of other agents. The value of information associated with an action includes the information that is provided about other agent's strategies. The work [5] studies the reactive learning in MAS. The central problem addressed is how several agents can collectively learn to coordinate their actions in a way to solve a given environmental task together. In approaching this problem, two important constraints have to be taken into consideration: the incompatibility constraint, that is, the fact that different actions may be mutually exclusive; and the local information constraint, that is a fraction of its environment. They propose two algorithms, called ACE and AGE (ACtion Estimation and Action Group Estimation, respectively), for the reinforcement learning of appropriate sequences of action sets in MAS. The model proposed by [4] will be compared later with our learning model based on AC.

The previous works have the same north 'coordination in MAS', besides to have a planning procedures-oriented approach and coordinated actions, with the goal to ensure the achievement of the MAS objectives. This means addressing the internal behavior of the agents to achieve learning. That is, they attack the problem of coordination based on internal actions and behaviors of MAS, for obtain a strategies model of the agents, and achieve collective learning. This class of approaches needs an internal knowledge model of each agent into the MAS.

Our model also aims to coordination in MAS, searching the optimization of coordination schemes through a collective learning process, but from another point of view, an external approach to the MAS based on the communication process, and using the common space provided by the CA. It is from this space of exchanging knowledge that the MAS achieves to discern (learn collectively) which is the most suited CM for a set of conversations. Thus, the proposed learning model optimizes the coordination of MAS relating to the communication tasks, by considering the costs of processing and communication generated by each CM.

The paper is organized as follows; section 2 discusses the theoretical framework in which the model is based. Section 3 presents the proposed formal learning model for coordination schemes for MAS. Section 4 presents the design and implementation of CLEMAS tool. Section 5 presents the application of the model to a case of study and the results; finally, section 6 presents the conclusions. The case study is in the field of industrial automation, which is a MAS-based Fault Management System (FMS).

2 Theoretical Framework

For this paper, we can cite three important issues: cultural algorithms, the problem of coordination in MAS, and collective learning in MAS.

2.1 The problem of coordination in MAS

We can describe coordination in SMA as the set of complementary necessary activities to be performed in a community of agents to act collectively [1]. In MAS, coordination can be seen as a process in which agents involved execute their communication acts in a coherent manner. Coherence refers to how well an agents system behaves as a unit. There are several reasons about why agents need to be coordinated [6]:

- Prevent anarchy or chaos: coordination is necessary or desirable because, with the decentralization in agent-based systems, anarchy can set in easily. Any agent no longer does possess a global view of the entire community to which it belongs. This is simply not feasible in any community of reasonable complexity.
- Meet global constraints: there usually exist global constraints which a group of agents must satisfy to be deemed successful. Agents need to coordinate their behavior if they must meet such global constraints.
- Distribute expertise, resources or information: agents may have different capabilities and specialized knowledge. Alternatively, they may have different sources of information, resources (e.g. processing power, memory), reliability levels, responsibilities, limitations, charges for services, and so on. In such scenarios, agents have to be coordinated.

Now, one way to obtain a generalized approach for the coordination of MAS from the communication point of view, is characterizing interactions among agents. In this work, a conversation is a set of interactions in the community of agents to achieve a goal at a given time. These interactions are called "speech acts" or "communicative acts". As we said, the conversations can be decomposed, in turn, into sub-conversations. The conversations or sub-conversations can be characterized by types of conversations (TCs), which are specific patterns of interactions. In [2, 3] four TCs are defined, based on the FIPA communicative acts, given by:

- TC1: Consult. An agent searches any kind of information in databases, repositories, warehouses and Internet.
- TC2: Assign. An agent assigns the tasks performing to other agents.

- TC3: Inform. An agent informs to others the occurrence of a certain event, information that is being processed or another kind of information.
- TC4: Request. The sender requests to the receiver to perform some action or service.

Moreover, these TCs are performed following some CM. In particular, our model uses the following CM standardized by FIPA: English Auction (SI), Dutch Auction (SH), the Contract Net (Tender, L), and Planning (PL). They have been formalized mathematically in previous studies [2, 3], and they are used by CLEMAS tool. Thus, in our model, a conversation can be characterized by one or more TCs, and each one may be treated by different CM.

2.2 Collective learning in MAS

There is a common agreement that there are two important reasons for studying learning in MAS: to be able to endow artificial MAS (e.g., robot swarms, software agents) with the ability to automatically improve their behavior; and to get a better understanding of the learning processes in natural multi-agent systems (e.g., human groups or societies). In MAS two forms of learning can be distinguished [5]: first, centralized or isolated learning, i.e. the learning carried out by a single agent (e.g. motor activities); and second, distributed or collective learning, i.e. the learning carried out by the agents as a group (e.g. by exchanging knowledge or by observing other agents).

2.3 Cultural Algorithms

Evolutionary computation (EC) methods have been successful used in solving many diverse problems of search and optimization due to unbiased nature of their operations, which can be interesting in situations with poor or no knowledge domain. However, there can be considerable improvement in their performance when problems with specific knowledge are used in the solving process. In order to identify patterns in their performance environment [7]. Cultural algorithms (CA) were developed by Robert G. Reynolds, as a complement to the metaphor used by EC algorithms. They are based in the fact that the cultural evolution can be seen as a process of inheritance in two levels: the micro-evolution level, which consist in the genetic material inherited from parent to their offspring, and the macro-evolutionary level, which is the knowledge acquired by individuals through generations, and once encoded and stored, serves to guide the behavior of individuals belonging to a population. These algorithms work in two spaces. The first, the population space with a set of individuals, like in all EC methods. Each individual has a set of independent features, with which it is possible to determine their fitness (objective function). On this space operate genetic operators, such as crossover and mutation, for his reproduction. The second is the belief space, where knowledge of previous individuals' generation is stored. From the five knowledge used in this belief space defined in the literature, two are the most used (and it will be used in this work), the "situational knowledge", consisting of specific examples of important events, such as successful and unsuccessful solutions, and the "normative knowledge", which is a collection of ranges of desirable values for the individuals in the population component. There are also a communication protocol that allows the interaction between these two spaces, formed by the "acceptance function" and the "influence function" [7]. In the Fig. 1 is illustrated each one of the components of the CA with its operators.

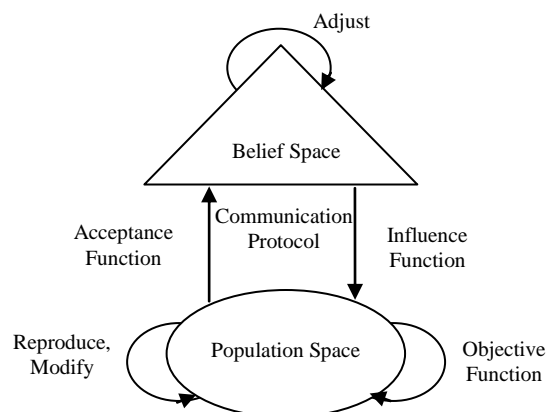


Figure 1: Cultural Algorithms Framework

3 Learning Model of Coordination Schemes in MAS

The formal model of learning proposed in this work involves the formal definition of the CM and the components of a CA [8]. Specifically, the components of the learning model are: the population, the CM specification, the belief space, the objective function, and the acceptance and influence functions. A basic pseudo-code for our learning model is shown in Fig. 2.

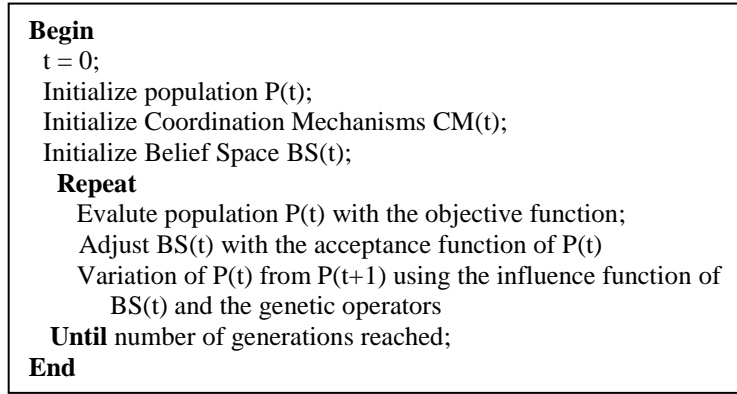


Figure 2: Basic Pseudo-code for CA- Based Learning Model

The population, as in any method of CE, is formed by individuals. In our model, each individual is a MAS composed of n different conversations in the community of agents (that is an instantiation of the MAS using in each case different CMs). Remember that, eventually, every conversation, in turn, has sub-conversations. All of them are characterized by the TCs previously defined in section 2.1.

In the Fig. 3, C_i denotes the conversation i existing in the MAS, FO is the value of the objective function of the individual (instance of the MAS), $C_{i,k}$ denotes the sub-conversation k of the conversation i , being m_i the number of associated sub-conversations to the conversation, TCs is the type of conversation, $MC_{i,k}$ is the CM used, and $P_{i,k}^u$ are the u parameters of this CM. When a conversation have not sub-conversations, $k = 1$.

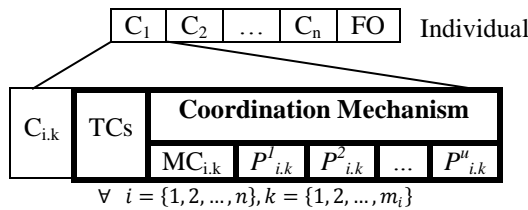


Figure 3: Internal Structure of an Individual

The highlighted part of the individual represents the knowledge or experience that it brings. In order to describe a little more the Fig. 3, the following example assumes that: There is one individual, a MAS, with three conversations (C_1, C_2, C_3), where C_1 has two sub-conversations ($C_{1,1}, C_{1,2}$), C_2 has two sub-conversations ($C_{2,1}, C_{2,2}$) and C_3 has three sub-conversations ($C_{3,1}, C_{3,2}, C_{3,3}$). Besides, it is assumed that the sub-conversation $C_{2,1}$ has a type of sub-conversation TC2 assigned. And finally, for that TC the individual uses the CM *english auction* (SI). In the Fig. 4 is shown this example, doing a specific zoom for $C_{2,1}$. This figure also represents the gene of the individual, where C_o , $C_p(j)$, and ϵ_i are the parameters of the CM used by $C_{2,1}$ (see [2, 8] for more details about the CM).

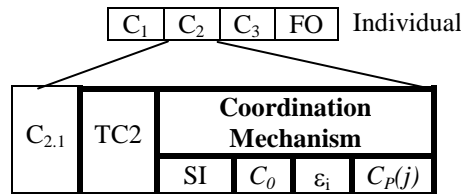
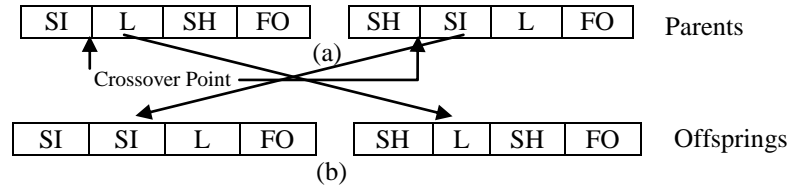


Figure 4: Example of a Gene on Individual, Zoom in C_2

For the reproduction of the population are used two genetic operators, mutation and crossover, and only a part of the individual, that is, a gene, is used to apply these genetic operators. For example, assume two individuals like that one of the Fig. 4. We further assume that each conversation (C_1, C_2, C_3) has only one sub-conversation (that is, the conversation itself), then, that individual shall be composed of three TCs. Now assume that each of these TCs is defined by a CM. For mutation and crossover, the individual genes are those ones representing the CM and their specific parameters. For these two individual, the application of the crossover operator is shown in Fig. 5.

In Fig. 5, part (a) are the parents and part (b) are their offspring. In the parent to the left, the first CM is 'SI' for the TC of C_1 , 'L' for the TC of C_2 , and 'SH' for the TC of C_3 , the same in the case of the another parent. The one-point crossover is used (indicated by the double arrow), which is only applied at the level of the CM. We see as two new children are generated. The mutation is simple, it is to take a CM or more from the conversations of a parent, and change it randomly by another CM.


Figure 5: Crossover Operator

3.2 Objective Function

The objective function evaluates the performance of each individual. This function is based on the Processing Cost (CP) and Communication Cost (CC) of each CM used by the individual (see equation (1)). There, the parameters a and b are constants defined by the user to weigh the importance of the communication part with respect to the processing part, n is the number of conversations, m_i is the number of sub-conversations in a conversation i , $CP_{i,k}$ and $CC_{i,k}$ are the processing cost and the communication cost, respectively, of the CM used in the sub-conversation k of the conversation i . For our case, the best individual will be the one that minimizes the objective function [2, 13].

$$FO = \sum_{i=1}^n \sum_{k=1}^{m_i} (a * CP_{i,k} + b * CC_{i,k}) \quad (1)$$

The processing cost $CP_{i,k}$ is given by the equation (2), and its units are based on the average execution time:

$$CP_{i,k} = PI_k + PE_k + \sum_{l=1}^j \sum_{q=1}^{n_j} A_{l,q} \quad (2)$$

This cost depends on the actors involved, and processing algorithms. For auction (English or Dutch), PI_k is the initial price setting and start of auction of the sub-conversation k , PE_k is the process of selecting the winning agent, j the number of rounds, n_j the number of bidders for round, and $A_{l,q}$ is the time to prepare the proposal for auction of the participating agents. For tender, PI_k is the specification of the initial conditions in which a service is required, PE_k is the process of selecting the service agent, j is equal to a round ($j=1$, in the tender case), n_j number of bidders, and $A_{l,q}$ is the time to prepare the proposal for bidding agents. For both coordination mechanisms, PI , PE and A , according to Table 1, are parameters qualitatively measured (e.g., low, medium and high).

Table 1: Qualitatively Values of the Parameters PI , PE and A

CM	PI	PE	A
Tender (L)	Medium	Medium	Medium
English and Dutch Auction (SI, SH)	Low	Medium	Low

To measure these parameters with numbers (due to that the equations require it), we have considered the Likert scale as reference [9], which allows us to assign numerical values to such parameters. Thus, we assign the value 0.2 to low, 0.6 to medium, and 1 to high. The other parameters are quantifiable.

The communication cost is based on the estimated time for messages exchange (communicative acts) for each CM in each conversation, and it is given in the equation (3) (FIPA has defined the communicative acts of each CM):

$$CC_{i,k} = \sum_{l=1}^j (\sum_{r=1}^{N-1} CEP_{l,r} + \sum_{s=1}^{n_j} CEO_{l,s}) + \sum_{r=1}^{N-1} CS_r \quad (3)$$

Where j is the number of rounds, $N-1$ the number of agents least the sender of message and n_j is the number of participants in each round. For auction and tender, CEP is the sending cost of the initial proposal, CEO is the sending cost of bids and CS is the cost of informing who makes service. Table 2 shows the qualitative values for the parameters CEP , CEO , CS .

Table 2: Parameters CEP , CEO , CS

CM	CEP	CEO	CS
Tender (L)	Low	Medium	Low
English and Dutch Auction (SI, SH)	Low	Medium	Low

3.3 Belief Space

There are two categories of knowledge in the belief space: situational and normative.

3.3.1 Situational Knowledge

In situational knowledge are kept examples of successful and unsuccessful individuals. In our model, this knowledge is based on each TCs, which include each CM used for performing this TC, their rate of occurrence (IO), and finally the total occurrences (TO) of the TC (see Fig. 6).

TCs	CMi	$IO_{(TCs, MCi, t-1)}$	$TO_{(TCs)}$
-------	-------	------------------------	--------------

$\forall s = \{1, 2, 3, 4\}, i = \{1, 2, \dots, n\}$

Figure 6: Situational Knowledge

3.3.2 Normative Knowledge

The normative knowledge keeps the suitable ranges for each of the variables of the CM used by the situational knowledge. In Fig. 7, LI and LS are the lower and upper limits of each parameter P^i forming each CM.

CMi	P^1		P^2		\dots	P^u	
	LI	LS	LI	LS		LI	LS

$\forall i = \{1, 2, \dots, n\}$

Figure 7: Normative Knowledge

3.4 Communication Protocol

Functions of acceptance and influence are those that allow the interaction between the population space and the belief space. These functions in this proposal are:

3.4.1 Acceptance Function for the Situational Knowledge

This function takes a percentage of the population (20% of individuals is sufficient according to Reynolds [5]), in order to nurture the belief space with their experiences. The acceptance function updates the situational knowledge as follows: for a specific TC each of the mechanisms involved in this type of conversation is updated by the equation (4), where $s=4$, because 4 is the number of the TC defined in section 2.1:

$$IO_{(TCs, MCi, t)} = IO_{(TCs, MCi, t-1)} + (NO_{(TCs, MCi, t)} / TO_{(TCs, t)}) \quad (4)$$

$\forall s = \{1, 2, 3, 4\}, i = \{1, 2, \dots, n\}$

Where $IO_{(TCs, MCi, t)}$ is the rate of occurrence in the iteration t for the $TC s$ and the MCi . $IO_{(TCs, MCi, t-1)}$ is the rate of occurrence in the iteration $t-1$ which is currently in the belief space, $TO_{(TCs)}$ is the total of occurrences of $TC s$, and $NO_{(TCs, MCi, t)}$ is the number of occurrences in the current instantiation of the MAS of MCi for this $TC s$. It is also necessary to update the total occurrences TO, by using the equation (5), where k is the number of CM used in this $TC s$:

$$TO_{(TCs, t)} = TO_{(TCs, t-1)} + \sum_{i=1}^k NO_{(TCs, MCi, t)_i} \quad (5)$$

$\forall s = \{1, 2, 3, 4\}, i = \{1, 2, \dots, n\}$

3.4.2 Acceptance Function for the Normative Knowledge

The acceptance function updates the normative knowledge by the following equation:

$$Lac(P^u) = [(lv * \bar{m} + \bar{P} * m) / 2] \quad (6)$$

Where, $Lac(P^u)$ is the current limit (either LI or LS), Lv is the previous limit, \bar{m} is the complement of the moment, namely, $(1 - m)$, \bar{P} is the average value of the limit of all individuals accepted within 20% from the population. Finally, m is the moment, that is given by the equation:

$$m = \mu / t \quad (7)$$

Where μ is a time constant between 0 and 1, and t is the iteration number, ($t = 1, 2, 3 \dots$). Thus, each time it reaches a new experience of the people, the limits of each parameter of the mechanism are updated.

3.4.3 Influence Function

The influence function determines how the knowledge of the system influences the individuals in the population. In the case of situational knowledge, it is based on the use of the mutation operator, which switches the current CM of a given conversation, according to a probabilistic rule (stochastic universal sampling or roulette wheel [10]), based on the IO parameter of each TC (we call that a targeted mutation).

In the case of normative knowledge also is based on the mutation operator, only that here the complete structure is not changed, but only specific values of the ranges are setting for each parameter of a given CM.

4 Design and Implementation of Cultural System of Multi-Agent Learning (CLEMAS).

CLEMAS is a computational tool to implement the proposed learning model. CLEMAS requires the configuration of various parameters. Upon the execution of the model, CLEMAS can display which CM is proposed for each conversation/ sub-conversation of the MAS. Moreover, CLEMAS can display the progress of the learning process.

4.1 CLEMAS Components

This tool has four main components: the execution engine, a system that emulates the CA, a graphical interface for configuring the system initially and visualizes the learning process with their results, and a database that stores the existing prior knowledge in the belief space (Fig. 8).

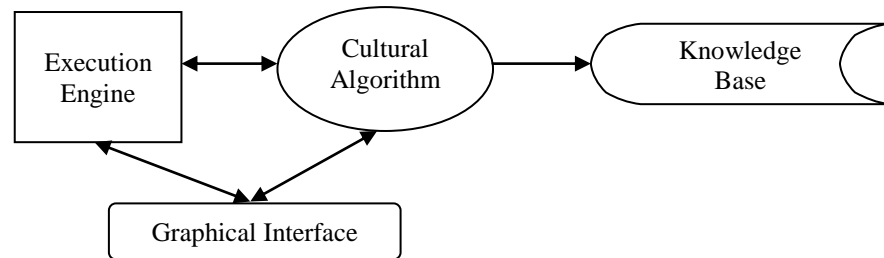


Figure 8: Structure of CLEMAS

The execution engine component runs the learning process through a class called 'simulation', using for this the initial system configuration. The CA component represents individuals and the belief space. The knowledge base component stores the situational and normative knowledge in belief space (a file with the extension .cgg). The following section details each of the components of CLEMAS.

4.2 Design of CLEMAS

4.2.1 Representation of the Individuals

To represent an individual in MAS a class called 'Individuo' is used, whose main attributes are a set of conversations, sub-conversations, an objective function and an identifier. The value of the objective function is calculated in each of the iterations; all calculations are performed by using the equations (1), (2) and (3) of section 3.2. Conversations in an individual contain a set of sub-conversations, whose main attributes are a TC and a CM. The classes used for the definition of the individual are shown in Fig. 9.

4.2.1 Representation of the Belief Spaces

To update the knowledge of the belief space the acceptance function is used, and in order to use this knowledge in the population the influence function is used. The classes used to develop the belief space are three.

- *EspacioCreencias*: it is the class that models the belief space; its main attributes are a set of both situational and normative knowledge. Also, it is responsible for implementing the functions of acceptance and influence.
- *ConocimientoCircunstancial*: it represents a situational knowledge; its main attributes are an identifier for the TC and one for the CM, an index of occurrence and the total of occurrence.
- *ConocimientoNormativo*: it represents a normative knowledge; its main attributes are an identifier of the CM, the list of its parameters and the lower and upper values of them.

4.2.2 Execution Engine

To run the learning model a class called 'Simulacion' is proposed, and its main attributes are maximum number of generations, population size, acceptance rate, crossover probability, mutation probability, a beliefs space and a set of state (this attribute allows to use the knowledge gained in previous simulations). This class 'simulacion' is responsible for the initialization of the system, determines which ones are the best individuals who will influence the belief space in each of iterations, the genetic operators to be used to create the next generation, and keeps track of what happens along simulation. To do this, it uses the following classes:

- *CruzarIndividuos*: is responsible for creating new individuals using crossover operator.
- *MutarIndividuo*: is responsible for creating a new individual using the mutation operator. The types of mutation are: direct mutation is a targeted mutation which is influenced by the belief space; and normal mutation is the classical mutation. The mutation is carried out with a certain probability. This class selects randomly the conversations and sub-conversations that are going to be mutated.

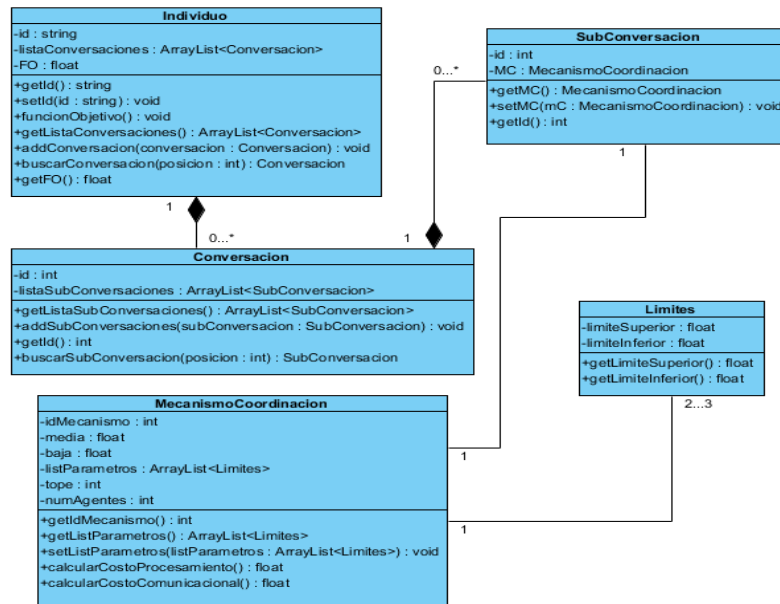


Figure 9: UML Diagram for the class “Individuo”

- *States*: This class stores, at each iteration, the knowledge generated into the belief space. It invokes the classes *ConocimientoNormativo* and *ConocimientoCircunstancial*, representing the cognitive structures that are stored in the belief space.

4.2.3 Control Panel for setting the Simulation Conditions

This panel has the different controls necessary to run CLEMAS (see Section 5 for examples of this panel). For this, the following classes are defined:

- *PanelCentral*: class with the main panel of the system, it allows to set up the initial parameters of CLEMAS.
- *PanelConversacionesCreadas*: with this class we can create, edit and modify the conversations.
- *PanelConversacion*: with this class we can edit the conversations.
- *PanelSubConversacion*: this class allows designing a sub-conversation, creating the components that allow the user to set the type of conversation, the parameters of each mechanism, the maximum of rounds (in the case of auction) and the number of agents involved in the sub-conversation.

4.2.4 Panel to display the results

It defines a window that displays the results in each iteration. To achieve it, several classes are designed:

- *PanelResultados*: It is responsible for creating the output window. It receives as parameter a reference to the 'Simulacion' class, processes all the information obtained from the simulation, and creates the visual components that allow the user to see the results (see Section 5 for examples of this panel).
- *PanelGraficoFO*: This mechanism is responsible for plotting the behavior of the CA throughout the simulation.
- *PanelGraficoMecanismo*: This class calculates the average use of CM for 20% of individuals accepted in each of iterations, and performs a graphic of CM with respect to iterations, namely, number of generations.
- *PanelTabla*: Create the table percentage of use of each CM with respect to TC.
- *PanelHistorial*: This class is in charge of creating a window for displaying detailed historical information about each of iterations in the simulation. It shows the situational and normative knowledge evolution, further genetic operators, in particular when using direct or normal mutation.

5 Experimentation with CLEMAS

In this section different experiments have been carried out to show the performance of our learning model. For this, we consider a case of study oriented to the agent-based industrial automation, described briefly below.

5.1 Case of Study: Fault Management System MAS-based

This case of study is a MAS for faults handling in industrial processes, whose specification is described in detail in [11]. The Fault Management Systems (FMS) is a system at the supervision level of an industrial process. The FMS

is composed of two modules, the first performs the monitoring and failure analysis, and the second performs the tasks of the maintenance management system. FMS interacts with the Maintenance Engineering and the Fault Tolerant Process. FMS can be seen as a system composed of intelligent agents that cooperate to solve problems related to the handling of system failures. Furthermore, some activities of the FMS follow a distributed computing model, such as those performed for the fault detection in equipment or processes, the performance index estimation, among others. To illustrate the application of our proposed CA-based learning for the coordination of the MAS, two specific conversations of MAS-based FMS are taken. Before that, the coordination model is presented in a general way.

Coordination Model: The MAS has six conversations that are: maintenance by condition (C1), maintenance tasks (C2), urgent tasks (C3), replanning of tasks (C4), state of maintenance (C5), and identify functional failure (C6). Only two will be detailed by using the TC proposed in section 2.1:

- Conversation 4 (C4): Replanning of Tasks, this conversation is made up of three sub-conversations: C4.1 of type TC1, C4.2 of type TC3 and C4.3 of type TC4.
Description: Through this conversation, the coordinator agent seeks information from the database agent to reschedule outstanding maintenance tasks on the system, and make a new maintenance plan. If the task is urgent and it cannot reschedule, an alarm is given.
- Conversation 5 (C5): State of Maintenance, this conversation is made up of four sub-conversations: C5.1 of type TC1, C5.2 of type TC1, C5.3 of type TC3 and C5.4 of type TC4.
Description: Through this conversation, the observer agent seeks information from the database and the actuator agent, to store outstanding maintenance tasks on the system.

In Fig. 10, the interaction diagram of the conversation C5 is presented, in order to show the characterization of TCs in the sub-conversations. In that conversation the observer agent (AO) makes a consult (TC1) in the database twice (process information (TC1) and maintenance information (TC1)), reports (inform, TC3) to the actuator agent (AA) the maintenance tasks, and if those have been not made, requests (TC4) to agent database (ABD) to be incorporated into the database.

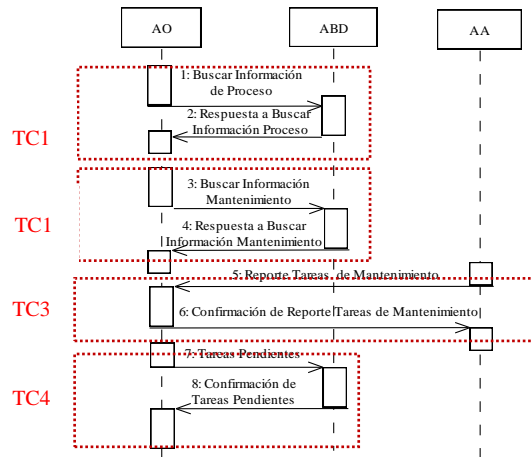


Figure 10: Conversation with his TCs

5.2 Design of the Experiments

For the case study, two different scenarios have been characterized:

Scenario 1: For this scenario, we assume C4 to be optimized only the sub-conversation (C4.1) which has 4 agents (3 database agents and 1 coordinator agent), and C5 to be optimized for a single sub-conversation (C5.3) with 3 agents (1 observer agent and 2 actuators agents). The objective of the simulation is to show how the number of iterations influences the learning process. To achieve this, CLEMAS is simply configured with a low number of generations. For the maximum number of auction rounds is 5 and for tender 1. The initial values of the parameters of the mechanisms are: $C_0 = [5 \dots 15]$, $\epsilon = [5 \dots 20]$, $C_p(j) = [1 \dots 5]$. To tender $M(F) = [1 \dots 3]$, and $f(T) = [5 \dots 20]$. For this simulation the population is 20 individuals, 8 generations (8 iterations), crossover probability of 0.7 and mutation of 0.5.

Scenario 2: In this scenario the same sub-conversations are assumed to be optimized, but for C4.1 the number of agents was increased to double, that is, 8 agents; likewise for C5.3 is increasing to 6 agents, this is in order to observe the behavior of the system scalability. Also, the number of generations is increases to 16 to see if individuals actually improve their behavior. The values of the initial parameters, population and genetic probabilities remain the same in this scenario.

5.3 Simulation Results

Fig. 11 shows the numerical summary presented by CLEMAS at the end of the simulation. In this case, for the scenario 1, a detailed description of the final results concerning to the evolutionary process and historical results in each of iterations are given in the Fig. 12.

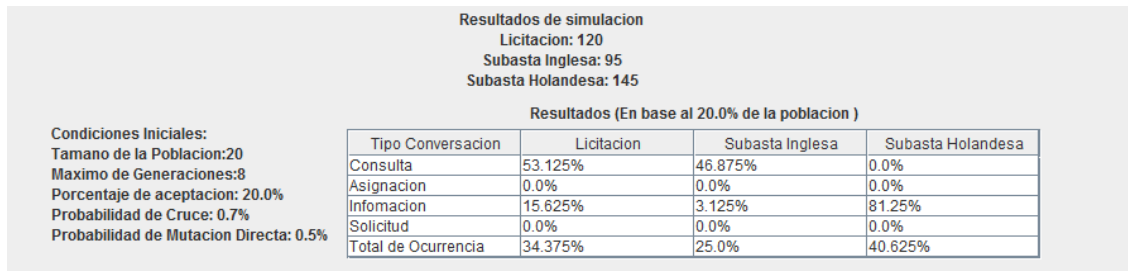


Figure 11: Summary table of results in CLEMAS for the Scenario 1

In Fig. 12, we see that the TC being optimized were TC1 (consult) and TC3 (inform), which are precisely the TC using C4.1 and C5.3. The coordination mechanisms L and SI were used almost interchangeably for TC1, and SH is used 81.25% of the time for TC3. In Fig. 12 (a) shows that during the first three generations prevailed L (called 'Licitacion' in the Fig. 12), but from that generation SH (called 'subasta Holandesa' in the Fig. 12) prevailed. In Fig. 12 (b) the red curve represents the average objective function of the population. The blue curve represents the objective function of 20% of the population (selected for the individual acceptance function). Finally the green curve represents the average of the remaining 80% of the population. As can be seen, the two curves (total average and not accepted population) are largely far from of the accepted average (desired behavior).

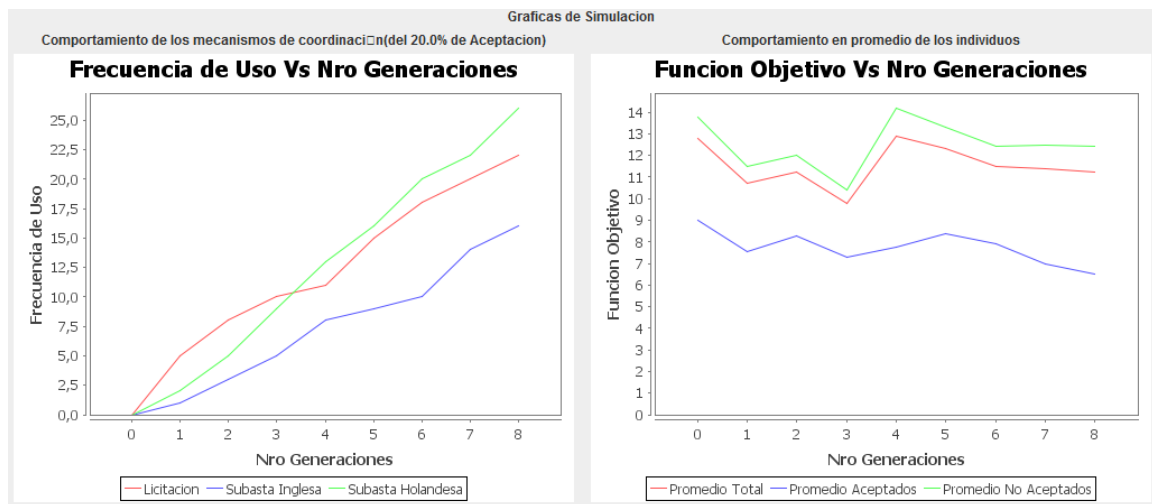


Figure 12: (a) Graphic of the evolution of the CM and (b) The objective function for the Scenario One.

According to Fig. 13, for the scenario 2, SH is predominating in TC3 and L is predominating in TC1. This makes sense because the number of agents increases in TC3, which generates a cost that can be minimized using SH. Regarding TC1, the use of L will be increased, because it reduces the costs for being a single round.

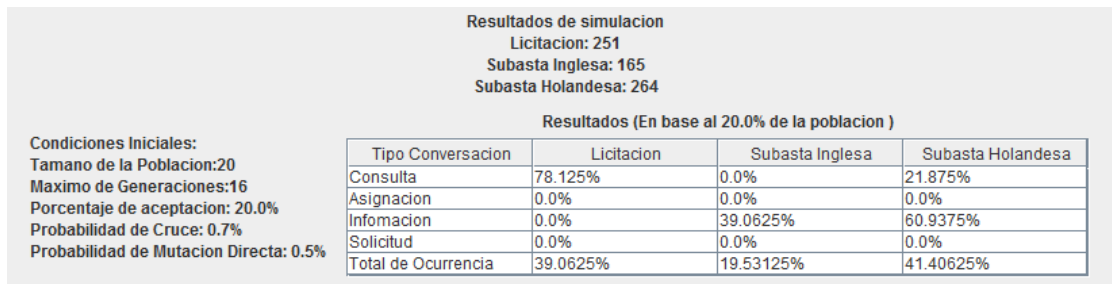


Figure 13: Summary table of results in CLEMAS for the Scenario Two

Fig. 14(a) shows that the SH and the L curves are very close, from the ninth generation; this is due to the increase in the number of agents in each sub-conversation. Finally, Fig. 14(b) shows the difference between the objective functions in the early generations, but as individuals evolve (greater number of generations), its tendency is to

follow the blue curve, which are the best individuals, this is because increasing the number of generations individuals acquire greater knowledge and learning (they follow the collective knowledge).

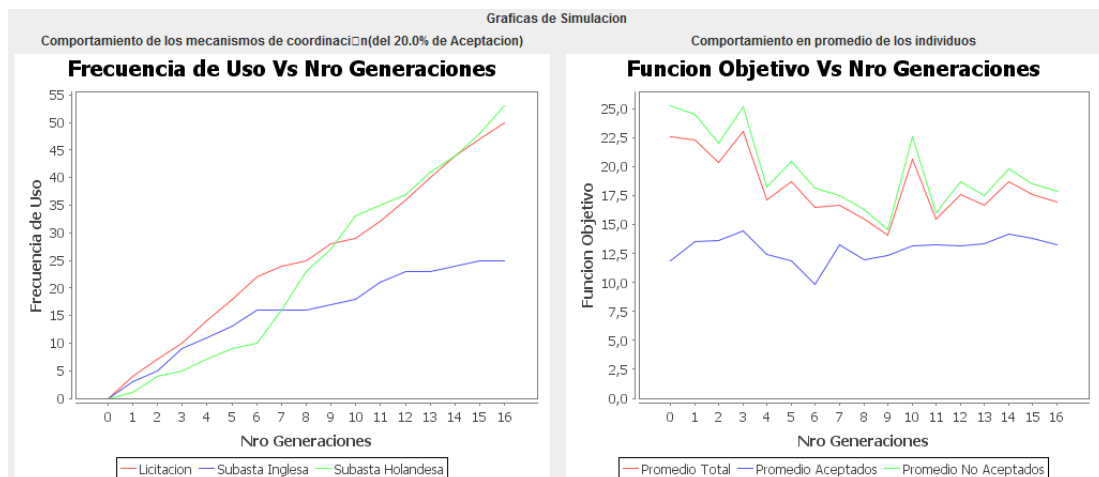


Figure 14: (a) Graphic of the evolution of the CM and (b) The objective function for the Scenario Two.

6 Comparison of Model Based on Cultural Algorithms with other Models of Learning

Before comparing the model proposed with other learning techniques, is important to note the following: the learning model proposed in this paper has the distinction of not needing to simulate the services of the MAS, i.e., our proposal does not consider decisions, actions, or internal orders, to achieve the goals of the MAS, since we assume that the MAS achieves its tasks in full. The focus of the proposal is that each individual of CA is an instantiation of the conversations of the MAS, characterized by a set of CMs, in order to communicate among them. These individuals share a common space (the space of beliefs) accessible to each ones. This is where the learning emerges, proposing a suitable CM for every TC. Below is a table comparing the Bayesian model in RL proposed in [7] and our model based on CA. The table consists of the most important aspects in both models: learning source (here we compare in which model for learning is based), performance measurement (here we explore how the evaluation of the learning process in each model is), knowledge and experience (which past experiences and knowledge are used for learning).

Table 3: Comparison of Models RL and CA

Features	RL	CA
Learning source	Action and reward functions	Conversations, types of conversations, coordination mechanisms and objective function
performance measurement	Rewards (R) actions based calculated on the Bellman equation	Processing and communication costs calculated by the objective function (OF)
knowledge and experience	$\langle s, a, r, t \rangle$	Situational and normative

Table 3 shows that the learning source is different in both models. The RL models observe the actions of individuals, and based on achieved rewards they learn. This requires sharing actions and rewards among the agents to make everyone learn/know what the other do, which is quite demanding. In our model based on CA, the individuals base their learning for each observed performance (through the FO) in a particular type of conversation (TC), and transfer that information, characterized by the TC and CM used, in a common space for all individuals (belief space). Both models have performance measures based on costs and rewards. The learning equation in RL is based on the Bellman equation, which seeks to maximize rewards. In the case of CA the idea is to minimize the costs of communication and processing of the CM. The experiences used in RL are based on states (s, current state; t, future state; a, actions; and r, rewards). The level of knowledge in RL is very particular because it is based on events leading from a current state to a future state, and accumulated rewards of these actions. In CA the belief space consists of a situational knowledge, which is knowledge of successful experiences of individuals, and

normative knowledge, which are ranges of favorable values of each CM, but also other kinds of knowledge could be incorporated. Thus, in general, one can conclude:

- The goal of coordination in RL is based on actions and strategies adopted by individuals, having as indices performance rewards that allow assessing these actions. In CA, the learning is achieved about the coordination schemes that would best suit the MAS for different TC, considering its computation and communication costs.
- CA has greater robustness with respect to RL, because the knowledge and experiences it uses are the result of an evolution, beyond seeking to coordinate actions to achieve a given task.

7 Conclusions

Cultural algorithms are presented as a powerful learning tool for individuals in different societies. In this work it has been used to learn how to coordinate MAS. A learning model of coordination schemes for communities of agents using CA is proposed. The cultural model is systematized in the CLEMAS platform, which allows interactively test different scenarios for a case of study. It graphically displays the results of these scenarios.

One of the main advantages of the system is its simplicity and flexibility to adapt to any scenario, allowing tests of aspects like scalability, size of the agent community, etc. In addition, all the accumulated knowledge in the belief space can be reused by the system to optimize the CM of other MAS. In summary, we have presented a Cultural Learning System for coordination schemes for MAS, and the same has been applied to a case study, a fault handler system based on MAS. CLEMAS is presented as a useful tool for collective learning in communities of agents, and can handle different types of knowledge (in our case, situational and normative).

Further works will do a more thorough study of the different parameters that can be considered in a learning process of coordination mechanisms of MAS (number of agents, number of communications, etc.), and about the suitable values of CLEMAS (number of generations, probabilities, etc.).

Acknowledgement

To the CDCHT project I-1237-10-02-AA of the Universidad de Los Andes, and the FONACIT project PEI-2011001325, for their financial support.

References

- [1] J. Aguilar, A. Rios Bolivar, F. Hidrobo, M. Cerrada, *Sistemas multiagentes y sus aplicaciones en automatización industrial*, First edition, Consejo de Publicaciones, Universidad de Los Andes, 2012.
- [2] J. C. Terán, J. L. Aguilar, M. Cerrada, "Modelo cultural para la coordinación de sistemas multiagente", Centro de Estudios en Microelectrónica y Sistemas Distribuidos (CEMISID), Universidad de Los Andes, Tech. Rep. 3-2013, 2013.
- [3] A. Stefano, S. Ramamoorthy, "A game theoretic model and best-response learning method for ad hoc coordination in multi-agent systems". In *Proc. of the 12th International Conference on Autonomous Agents and Multiagent Systems*, St. Paul, Minnesota, USA, 2013
- [4] G. Chalkiadakis, C. Boutilier, "Coordination in Multiagent Reinforcement Learning: A Bayesian Approach." In *Proc. of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, Melbourne, Australia, 2003, pp. 709-716.
- [5] R. Reynolds, "Cultural Algorithms: Theory and Applications". In *New Ideas in Optimization* (David Corne, Marco Dorigo and Fred Glover, Editors). McGraw-Hill, pp.367-377, 1999.
- [6] H. Nwana, L. Lee, N. Jennings, "Coordination in software agent systems". *BT Technol J*, vol. 14, pp. 79-89, 1996.
- [7] G. Weiss. Lernen und Aktionskoordination in Mehragentensystemen. In *Verteilte Künstliche Intelligenz – Methoden und Anwendungen* (J. Müller editor), BI Verlag, pp. 122–132. 1993.
- [8] J. Terán, J. L. Aguilar and M. Cerrada, "Mathematical models of coordination mechanisms in multi-agent systems," *Clei Electronic Journal*, vol 16, 2013.
- [9] D. Bertram, "Likert Scales", CPSC 681. Tech. Rep. <http://poincare.matf.bg.ac.rs/~kristina/topic-dane-likert.pdf>
- [10] M. Mitchell, *An introduction to genetic algorithms*, Eighth printing, Massachusetts Institute of Technology (MIT), 2002.
- [11] M. Cerrada, J. Aguilar, J. Cardillo, R. Faneite, "Agents-based design for fault management systems in industrial processes", *Journal Computers in Industry*, vol. 58, pp. 313-328, 2007.