

Intellectual Development Model for Multi-Robot Systems

An Approach Based on Social Learning Models

Rosier S. Maia · Luiz M. G. Gonçalves

Received: 3 June 2014 / Accepted: 9 March 2015 / Published online: 31 March 2015
© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract We propose the IDeM-MRS learning formalism to be used by a group of robots for solving practical tasks in indoor environments. The formalism is inspired on the theory of social learning models for human beings that is traditionally developed in Psychology and Education fields. Our model can be used for coordination of the group, as for, allowing assimilation and accommodation of knowledge through experience exchange. Besides explaining the theoretical model itself, we formalize the mathematics involved with it in a very simple and straightforward fashion. Some issues are especially investigated such as the realistic representation of the multi-robot environment involving the global mission, the tasks belonging to the mission and the active set of robots. A way for task selection is proposed based on social learning theories and approaches that allow cooperative and efficient execution of tasks by robots. To this end, IDeM-MRS can be used in different types of missions varying from simple to complex. Experiments and results validate the efficiency of the formalism compared to a traditional empirical model.

Keywords Multi-robot cooperation · Social learning models · Learning formalism

1 Introduction

In general, applications in the field of autonomous mobile robotics require a high degree of flexibility, adaptability and efficiency mainly in tasks in which cooperation among a group of robots is necessary in order to solve a global problem. The term *multi-robot system* is adopted to denominate a group of robots whose global task is common to all of them [18]. Group cooperation is one of the main capabilities of a multi-robot system that is necessary to reach the global objective of the mission. So it is of great relevance to look for alternatives that make this cooperation as efficient as possible.

In the context of multi-robot systems directed to the solution of tasks through cooperation, it would be interesting to have a good relationship for knowledge and experience exchange inside the group and, consequently, to make possible the inclusion of knowledge, individually.

The problem of task allocation is not simple because it depends of acting issues like: designing the order of the basic operations to be carried out, establishing the hardware features necessary for the robot in order to perform a specific operation, defining which robot is suitable to solve a given operation. In particular, multi-robot systems deal

R. S. Maia (✉)
Computer Science Department, State university of Rio Grande do Norte, Natal RN, Brazil
e-mail: rosierymaia@uern.br

L. M. G. Gonçalves
Computer Engineering Department, Federal University of Rio Grande do Norte, Natal RN, Brazil
e-mail: lmargos@dca.ufrn.br

with difficulties inherent to sensor noise, unexpected results from actions, restrictions in environmental communication, and hardware failures, between others. As these features may affect the consistency of the global solution, the present research raises questions about real-time execution when facing a group of robots committed to solve a global task without the presence of a leader robot. The leader robot is generally one whose authority has priority over the rest of the robots in the environment. Therefore, we suppose that each individual robot is able to find its own way to solve the global problem from a local perspective in a decentralized way.

The main contribution of this research is the formalization of a mathematical model, named Intellectual Development Model for Multi-Robot Systems or IDeM-MRS for short, to be used by a group of robots which must cooperate to solve a global objective in a common environment. This multi-robot system must solve problems in a wide array of application domains, so IDeM-MRS is convenient and intended to be used by groups of robots cooperating to achieve a global mission. In the model, group members cooperate to complete the mission through the division of task execution, through individual decisions coordinating their actions, and contributing to the fulfillment of the objective in a distributed and collaborative manner. The behavioral model of robots under IDeM-MRS is formalized based on Social Learning Models, which are traditionally standardized for humans. In short, each robot keeps its own beliefs about the current state and the available resources of the environment. Its actions are driven by its own wishes and intentions while selecting and executing a given task of the mission. The robots determine their action executions by building their own schedule and announcing their intentions in the execution of tasks. IDeM-MRS presents better performance in the execution of missions where each robot modifies its own state and the state of the environment.

In general, when a set of several tasks that composes a mission is assigned to a multi-robot system, the problem of cooperative performance of the mission appears. Each robot individually solves the problem of coordinated selection of tasks considering the performance. This is defined as Cooperative Mission Achievement Problem - CMAP and Coordinated Task

Selection Problem - CTSP, respectively. These two problems were previously dealt with and formally declared by Talay [20].

The IDeM-MRS formalism proposed here is able to solve CMAP and CTSP for each robot in a cooperative way following Talay approach [21], nevertheless allowing individual knowledge acquisition, thus differing from the above mentioned proposal.

The model developed warrants an efficient way to solve planning, allocation and implementation of tasks by independent entities (robots or agents) through action policies on which each robot is exposed in every moment. Each action policy of IDeM-MRS drives to solve the CMAP in an efficient way. The efficiency results of the formalism are verified experimentally through simulations of applications with different domains of tasks execution.

IDeM-MRS is focused to complex missions involving tasks with restrictions of resources and own knowledge. Besides, each robot has its own hardware resources and a database composed of basic commands directly related with basic tasks of the environment. Therefore, environments with a team of independent entities (agents or robots) needing to achieve a global goal are perfect domains for applying IDeM-MRS. The main goal of applying this formalism goes from minimizing the time spent in solving the entire mission to updating the database of each agent of the environment caused by the acquisition of individual knowledge. This is a result of the rules established by the IDeM-MRS that are applied on-line to the environment, obeying real-time restrictions. As said, these rules are based on social learning models of humans and they are mapped on the environment through the basic actions of each agent (robot).

In the remaining of this work, Section 2 presents the problem of cooperative implementation of a mission by a team of robots, as well as other important issues related to multi-robot systems. Related works are revised in Section 3. The theories about the learning process in humans are provided in Section 4. The IDeM-MRS formalism and its modules are designed and brought to the reader in Section 5. Section 6 lists some study cases and describes the experimental setup built for evaluation of the paradigm showing evolutions of the performance of IDeM-MRS in complex mission. Finally, the computational analysis based on the

results given by the experiments is given in Section 7, that presents final remarks about the work.

2 Problem Statement and Motivation

Among works involving autonomous mobile robots for task solving, the waiter serving drinks in a hotel [16], the postman [23], and the man offering transportation in a hospital [8] are some that have high level of flexibility and adaptability. Nonetheless, these works deal with a single robot, directed to cover previously known and specific tasks. Other examples involving a team of robots include problem modeling in electronic commerce [7], letter sending [5], truck route scheduling [13], regional energy demand modeling [22], rescue missions for victims of disasters [17]. Another work intended to be a complete system for programming a fleet of robots in a closed environment, whose objective is to carry out delivery orders, performing staggering and planning of routes for the robots is proposed by Surmann [19]. A common issue among these works is the need for coordination between the several robots in order to find out the solution of the problem despite not offering options to enhance the knowledge databases of the robots.

Cooperative robotics has being an important research field and will remain for a long according to studies carried out by the robotics community. Mainly, the use of a group of robots working together to perform different kinds of tasks can bring advantages over single-robot solutions, like a better performance in solving a problem that is possible in some tasks, more tolerance to general system failures and possibilities of distributed sensing.

The main feature of multi-robot typical environments is group cooperation, a fact that forces the system designer to find alternatives to coordinate this cooperation in the most efficient possible way. In this direction Chaimowicz [6] provided a strongly cohesive coordination structure for handling cooperative tasks. He established a framework of distributed multi-robot cooperation that can be used to solve problems in a wide array of application domains. Botelho [1] depicts a scheme for task allocation and enhancement of previously planned tasks at execution time based

on project scheduling. She comes up with a centralized scheme to be used by a robot team in order to enhance the mission execution time, on-line. Nevertheless, learning mechanisms are not implemented thus making robot learning with a static approach.

On its turn, Kambayashi [11] uses kind of an Ant Colony Optimization algorithm in mobile agents to simulate a multi-robot in a cooperative search. However, the agents are not able to learn neither by themselves nor with the help of other agents. They depend on the knowledge given by the user, according with the needs during collaboration with the environment. In an ancient work, Kalmar [10] already makes feasible the task learning, starting from an environment in which each robot has the same set of previous knowledge and intrinsic characteristics. But, as the environment is not restricted into an appropriate subspace, there is an exponential increasing in the complexity of the decision making problem. Besides this weakness, the work considers the learning of each robot separately, so the final goal is to define how much knowledge is added to the most evolved robot with and without considering collective behavior.

The problem of cooperative execution of complex missions by a team of robots, according to [2], can be formulated on the basis of the Resource Constrained Project Scheduling Problem (RCPSP), which is known to be a hard to solve problem (NP-Hard) in Operative Research [26]. Issues like unpredictability of on-line task execution (that causes unstable results) and inconsistencies (that are due to uncertainty in data from the environment) could be added to this problem thus being part of the outline of the main difficulties of the task allocation problem by a team of robots. In this direction, Talay [21] defines the main circumstances that may affect the global solution of the problem of task allocation in the real world. A formulation of the problem of cooperative execution of the mission through an adapted version of the RCPSP is proposed coming up with a framework for multi-robot system cooperation. This framework elegantly approaches the environmental components in a distributive way. However, it does not allow knowledge gaining in the robots either individually or in group. The lack of complete solutions has motivated us to work in this issue in this work. To this end we come up with a task memory com-

posed of actions that can be incrementally modified in each robot. Such dynamical structure allows the robots to learn while performing their tasks even allowing to enhancing a given way to perform a task.

3 Formulation of the Cooperative Execution of Task Problems

With an adapted version of the RCPSP formulation, the Cooperative Execution of Task Problem (CETP) by a multi-robot system is formulated as follows. Being a team of robots $A_t = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$, such that $m \in \mathbb{N}$, working at a given time t in the environment, and a set of tasks $T_t = \{\tau_1, \tau_2, \dots, \tau_n\}$, such that $n \in \mathbb{N}$, making part of the mission M , specified in the time t . The CETP consists of attaining the execution of M by the robots A_t working in the environment. Each robot α_i has a set of physical features and a set of basic knowledge previously conceived, defined by $C(\alpha_i) = \{c_1, c_2, \dots, c_x\}$ and $F(\alpha_i) = \{f_1, f_2, \dots, f_y\}$, respectively. The tasks also have these two contexts associated: a set of knowledge $C(\tau_j)$ necessary for execution and a set of required physical capabilities $F(\tau_j)$, such that $j \in \mathbb{N}$. Following those notations, the CETP has some restrictions:

- The same robot α_i cannot be designated to execute more than a single task τ_j ;

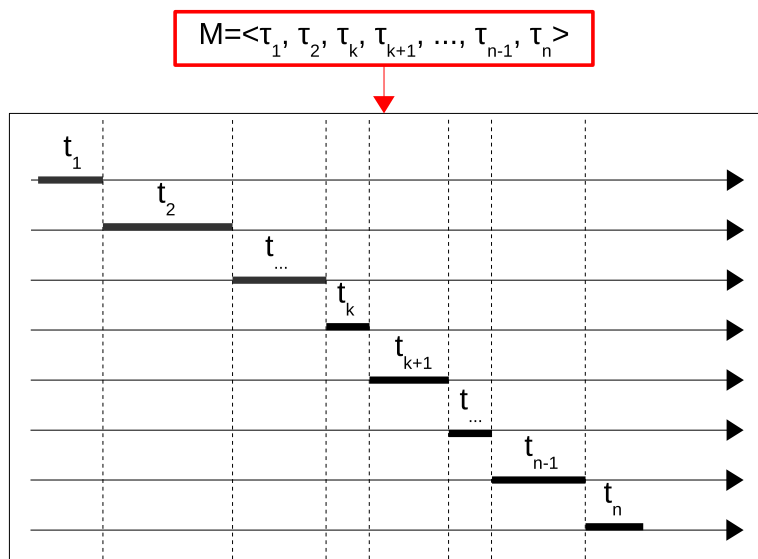
- Mission M is a list of tasks to be executed in a given order. Therefore, a task $\tau_j + 1$ will only be executed if τ_j has been assigned to some robot α_i ;
- The objective of the CETP is to finish the execution of all the tasks of mission M in the shortest possible time, knowing that the total mission time is calculated according to Fig. 1.

This problem can be exemplified with a specification of the *multiprocessor task scheduling problem* which is an NP-Hard problem, as Brucker [3] states. As this is a hard to solve problem, an optimization on the cooperation of the agents is welcome in order to solve the global problem of task scheduling. Note that, similarly and without loss of generality, this can be done first in a simulated environment and then extended to the real robots when facing application problems in robotics.

The basic premises A and T , when joining the conjuncts C and F , allows using the IDeM-MRS formalism. The remaining construction of this formalism is inspired by theoretical research known as Social Learning Models (SLM) of humans that is intensively studied in the field of Developmental Psychology.

The learning process of a human is quite complex and involves several aspects and variables as cognitive, affective, social, economic and even political. Once knowing this, in this research we investigate the SLM covering the social, humanistic and cognitive approaches as well as the ideas and theories of

Fig. 1 The total mission time is the execution time of all the tasks of mission. However, $total_t = t_1 + t_2 + \dots + t_{n-1} + t_n$, such that n is the number of tasks of the mission



the researchers Jean William Piaget and Lev Semyonovich Vygotsky. Basically, we formalize a mapping of the features of the investigated SLM to our multi-robot system earmarked for the CETP. Therefore, as Fig. 2 shows, the IDeM-MRS is obtained by mapping the approaches of the SLM in the CETP.

4 Theory of the Social Learning Models

As stated above, the goal of this work is to use the traditional theories about the learning process in humans in order to create a mathematical formulation able to direct the cooperation of a group of robots based on the exchange of experience and without human mediation. The IDeM-MRS makes possible, by using the concepts mapped in the SLM approaches, the cooperation among a group of robots to allow knowledge acquisition or modification of previous knowledge of each robot. After executing the CETP, the multi-robot system obtains a knowledge gain, as verified in the experiments performed in this research. Each approach analyzed is responsible for the construction of models for the final objective, which is the mathematical formulation transcribing all of the concepts included in studies with humans. In order to choose

the best one, we have investigated and mapped several of these approaches as will be described next. Besides not all of them are used in the experiments, we decide to leave all the studied approaches in this text in order to support alternative implementations by interested readers.

4.1 Vygotsky: Zone of Proximal Development

The psychologist Lev Semyonovich Vygotsky has been interested in the areas of developmental psychology, child development and education [24]. His socio-interactionist theory covered some of the concepts important for the formalism proposed in the current work. One of them is the Zone of Proximal Development (ZPD), that Vygotsky [25] describes as the difference between the Level of Potential Development (LPD) and the Level of Real Development (LRD), that is $ZPD_i = LPD_i - LRD_i$, where ZPD_i is the ZPD of individual i .

The LRD is the already consolidated knowledge of the individual, making the subject able to solve situations autonomously or, in other words, it provides the capacity of a person to solve a problem without help. This level of development is dynamic and grows in the course of the process of learning.

The LPD is already determined by the skills that the individual is able to construct being inferable from what the individual is able to solve with help.

In this way, the ZPD provides evidence of the potential, namely a series of information that the individual is yet capable to learn, even if his learning process is not complete. This information can be seen on Fig. 3.

When brought to the multi-robot environment, the LRD reports the capability of a robot to solve problems by using its own knowledge, autonomously. The LPD, on the other hand, defines the ability that the robot will develop when necessary. The ZPD of each robot, therefore, reports the difference between its capability to perform a task, autonomously or not, and its hability necessary to perform it. This information can be seen on Fig. 4. Basically, the ZPD means, for a given individual, what is he able to develop and, for a robot, what is it able to execute.

Through the knowledge acquired with the studies about the level of proximal development of humans, it is possible to construct the definition of the functions describing the LRD, LPD and ZPD of the robots as

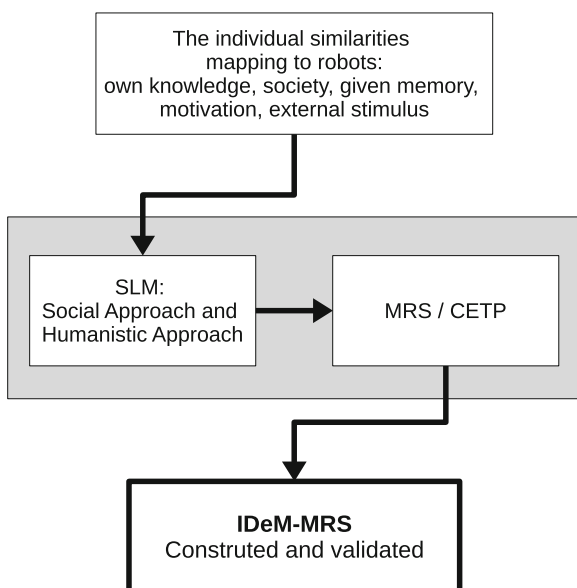
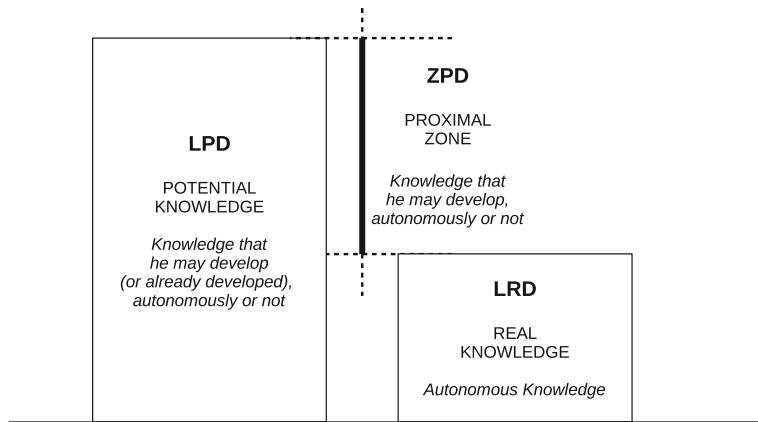


Fig. 2 The IDeM-MRS is construted through the mapping of the ideas preached by the SLM to the CETP. Each robot of the group is able to acquire new knowledge (for task achievement) or to modify their own way to solve a particular task

Fig. 3 For an individual, the ZPD is the knowledge that he may develop, autonomously or not; The LPD is the knowledge that he may develop (or already developed), autonomously or not; and the LRD is the autonomous knowledge that he has



being f_{Lrd} , f_{Lpd} and f_{zpd} , respectively. They are described below, considering that $i, j, y, w \in \mathbb{N}$, $1 \leq i \leq m$, $1 \leq j \leq n$, $y = \{0, 1\}$ and $w = \{-1, 1\}$, and that m is the number of robots at a same given time, and n is the number of tasks performed at the same given time.

$$f_{Lrd}(\alpha_i, \tau_j) = y \tag{1}$$

$$f_{Lpd}(\alpha_i, \tau_j) = -w \tag{2}$$

$$f_{zpd}(\alpha_i, \tau_j) = f_{Lrd}(\alpha_i, \tau_j) - f_{Lpd}(\alpha_i, \tau_j) \tag{3}$$

The LRD is obtained through (1). In this case, $f_{Lrd}(\alpha_i, \tau_j) = 0$ when the robot α_i cannot alone solve the task τ_j and $f_{Lrd}(\alpha_i, \tau_j) = 1$, otherwise. The (2) reports the LPD. When $f_{Lpd}(\alpha_i, \tau_j) = -1$, this means $w = 1$, the robot α_i has the capability to solve the task τ_j , and $f_{Lpd}(\alpha_i, \tau_j) = 1$ otherwise. The function f_{zpd} placed in Eq. 3 reports the ZPD of each robot. It can assume the values -1 and 1 making it easy to infer the obtained results thus enabling

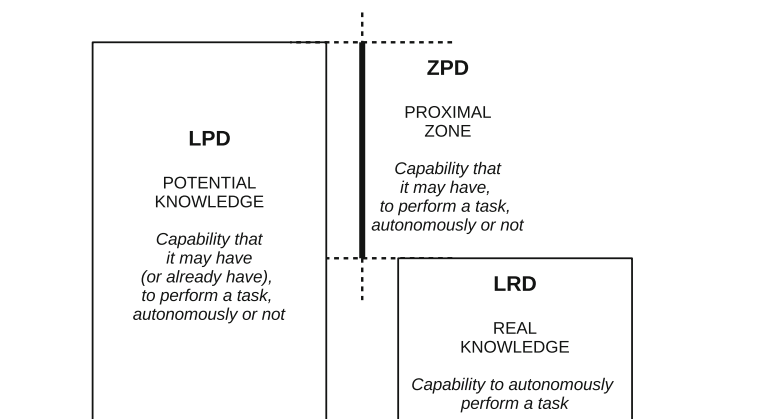
to identify the real and potential characters. Considering the evaluation of function f_{zpd} for the robot α_i related to the task τ_i , the results for $f_{zpd}(\alpha_i, \tau_j)$ are:

- i) If $f_{zpd}(\alpha_i, \tau_j) = -1$, then α_i does not know how to solve τ_i and is not able to learn how;
- ii) If $f_{zpd}(\alpha_i, \tau_j) = 0$, then α_i already knows how to solve τ_i , but is not able to develop its knowledge about the solution;
- iii) If $f_{zpd}(\alpha_i, \tau_j) = 1$, then α_i does not know how to solve τ_i , but is able to learn how;
- iv) If $f_{zpd}(\alpha_i, \tau_j) = 2$, α_i already knows how to solve τ_i and is also able to improve its knowledge.

4.2 Piaget: Balance Between Assimilation and Accommodation of Knowledge

Jean William Piaget [14] tries to explain, based on scientific findings, how intelligence is developed in human beings. According to him, the Genetic

Fig. 4 Mapping Vygotsky theory in a robot: ZPD reports the capability of task solution that it may have; the capability of performing a task, which it may have (or already has), autonomously or not is the LPD; and the autonomous capability to perform a task is the LRD



Epistemology shows the factors that influence the cognitive development of the individual as a process resulting from behavioral, cultural, social and biological features where the subject is inserted. Besides this, the psychologist preached a need to explain the Equilibrium Theory [9], imposing that the individual must be cognitively able to assimilate experiences and/or to adapt them, being therefore in harmony with his knowledge. In this case, we understand that *assimilation* is the incorporation of new knowledge and *accommodation* is updating of previous knowledge.

In fact, it is possible to establish a link between the ideas of Piaget and Vygotsky. *Assimilation* occurs if the knowledge is in the zone of proximal development of the individual. For *accommodation* the knowledge must be in the level of real development.

Of course, when mapping these concepts into a group of robots, applied to the CETP, the development of knowledge of each robot is a process resulting from intrinsic hardware and software characteristics. Furthermore, for the robot, assimilation covers acquisition of knowledge into its database, and accommodation is the development of a task, previously known. Meanwhile, the balance between assimilation and accommodation is induced, differently from the balance preached by Piaget, which is necessary here. Figure 5 shows these concepts mapped from real subjects to the robot.

Some relations need to be well defined for dealing correctly with the concepts related to Piaget's studies and, consequently, to obtain the basic requirements for those theories to be well structured in the context of the CETP. By knowing that the set of knowledge of the robot α_i is specified by $C(\alpha_i) = c_1, c_2, \dots, c_x$, it is necessary to understand the information described in the following, which is about assimilation and accommodation of the robot knowledge as well as about the expected results of those operations.

4.2.1 Assimilation

For the integration of knowledge into a robot to happen, it needs to be in a condition of not knowing how to solve the given task but it must be able to learn how to solve it. In this case, the function defining its ZPD must assume the value 1, once its LPD is -1 and its LRD is 0. In this way, for an robot α_i and a given task τ_j , the condition for assimilation is:

$f_{zpd}(\alpha_i, \tau_j) = 2$, being that, $f_{lrd}(\alpha_i, \tau_j) = 1$ and $f_{lpd}(\alpha_i, \tau_j) = -1$.

Then, to include the knowledge c in the mind of the robot α_i , in the instant t , such that $c' \notin C(\alpha_i)$, in the instant $t + 1$ we have that $C(\alpha_i) = C(\alpha_i) \cup \{c'\}$.

4.2.2 Accommodation

For a modification of the robot knowledge to happen, the robot needs to have the knowledge to solve a task and also to have the possibility of developing that knowledge. In this case, the function defining its ZPD must assume the value 2, once its LPD is -1 and its LRD is 1. In this way, for an robot α_i and a given task τ_j the condition for accommodation is: $f_{zpd}(\alpha_i, \tau_j) = 2$, such that $f_{lrd}(\alpha_i, \tau_j) = 1$ and $f_{lpd}(\alpha_i, \tau_j) = -1$. So, to modify the knowledge c in the robot α_i mind in the instant t by knowledge c' , this operation corresponds, in the instant $t + 1$, to $C(\alpha_i) = C(\alpha_i) - \{c\} \cup \{c'\}$.

4.2.3 Knowledge is the Result of Stimulus

Any modification resulting from a stimulus of the environment may produce manifestations in human behavior. Under these conditions, the individual needs to receive a stimulus in order to give a response. In fact, by stimulus it can be understood a request of knowledge coming from the environment, being it used immediately to solve tasks or being later used as an experience. However, it is possible to accept that the individual, when in the process of assimilation of knowledge, can be stimulated by the environment. The first moment is not an individual decision-making, but it is an acquisition of information. After this operation, the individual can take autonomous decisions or make accommodations of knowledge, since this information is on his level of Real Development.

In a multi-robot system, the robot receives a stimulus from a given task and responds with its knowledge about solving it or not, even without having autonomy for execution. Then, the function of stimulus is the response of the robot α_i to the task τ_j . It is defined by Eq. 4, such that $k \in \mathbb{N}$ and $k = \{0, 1\}$. This response reports if α_i detains the knowledge to solve τ_j , thus being autonomous or not. In the case of this function assuming value 1, then α_i knows how to solve τ_j . But assuming 0, then α_i does not know how to solve τ_j . Anyway, in the first case, α_i knows how to

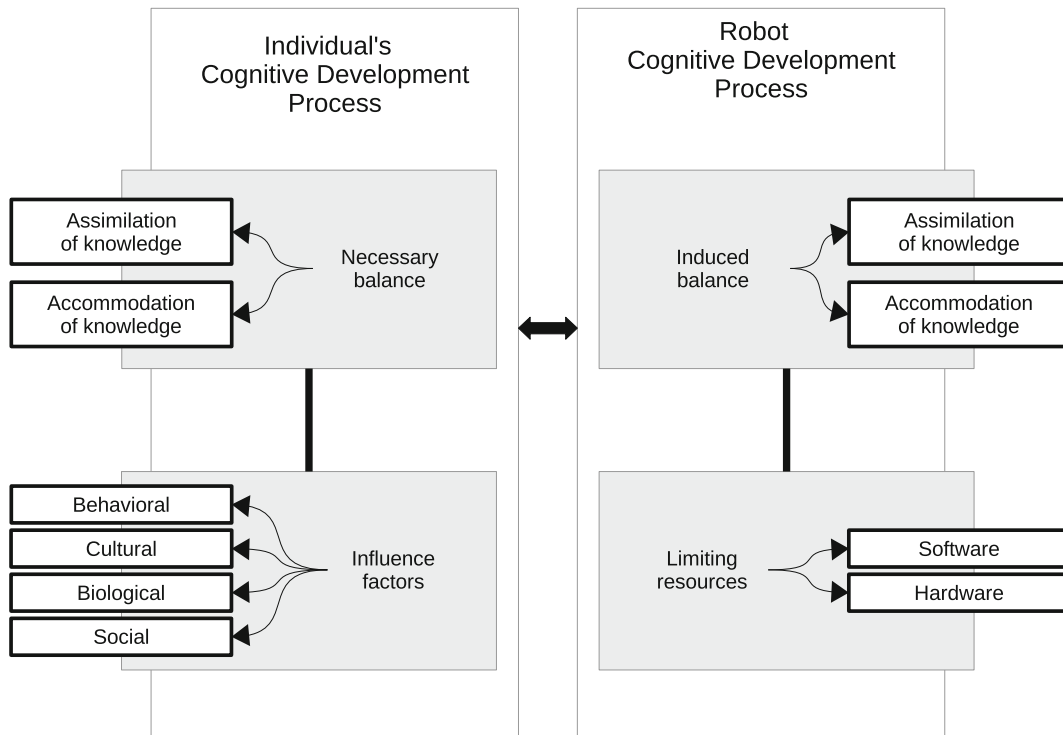


Fig. 5 The evolutionary process of cognitive development preached by Piaget has its sources in behavioural, cultural, social and biological factors. Besides, a balance is also necessary between assimilation of new knowledge and accommodation of previous knowledge. Therefore, for a robot, there must be an induced equilibrium between Assimilation (knowledge acquisition) and Accommodation (development about executing a task). For Piaget, the process of individual’s cognitive

development is the result of two factors: 1-behavioural, cultural, social and biological characteristics; 2-the necessary equilibrium between assimilating new experiences and accommodating others. By mapping these concepts, the process of the robot’s cognitive development is the result of: 1-hardware and software characteristics; 2-the induced equilibrium between assimilating new experiences and accommodating others

solve τ_i , even if needing to subdivide τ_i into smaller ones in order for other robots to help him. Note that this differs from function (1) when $f_{Jrd}(\alpha_i, \tau_j) = 1$ because this result reports that α_i is able to solve the task autonomously without the help of other robots of the environment.

$$f_{\mathcal{E}}(\tau_j, \alpha_i) = k \tag{4}$$

Therefore, from Eq. 4 it is possible to infer that:

- i) $f_{Jrd}(\alpha_i, \tau_j) = 1 \Rightarrow f_{\mathcal{E}}(\tau_j, \alpha_i) = 1$, because, if α_i knows how to solve τ_j alone, he will also produce a stimulus for that task;
- ii) $f_{Jrd}(\alpha_i, \tau_j) = 0$ does not restrict any value for $f_{\mathcal{E}}(\tau_j, \alpha_i)$ because even without getting to solve

τ_j alone, α_i is able to produce a response for the stimulus of that task;

- iii) $f_{\mathcal{E}}(\tau_j, \alpha_i) = 0 \Rightarrow f_{Jrd}(\alpha_i, \tau_j) = 0$, because if α_i cannot report a response for the stimulus of τ_j , it does not have the knowledge for its autonomous execution either;
- iv) $f_{\mathcal{E}}(\tau_j, \alpha_i) = 1$ does not restrict any value for $f_{Jrd}(\alpha_i, \tau_j)$ because even having a response for the stimulus of τ_j , α_i is not necessarily able to solve it alone.

4.3 The Social Environment Interferes with the Knowledge

Learning is a direct transformation, that is, a function that is directly related to the daily interactions among

people in the same social environment. This approach emphasizes that the individual learns by observing and interacting with other people of its social context. Biological questions are not discussed, because the individual is born with the basic knowledge (the same for every human being) and then goes adapting, according to the contact established with the society. The response of an individual to a stimulus undergoes according to the social environment where it lives in and interacts with others.

In this case, the social context of the robot is its environment knowing that the environment is reconfigurable. As all of the robots are inserted in the same environment, the social context is the same for all robots of the system, being specified by Φ , as we already used in previous research [12]. Anyway, with this approach it is possible to define the response of a robot α_i to a stimulus from the environment by the relation between the set of knowledge ($C(\alpha_i) = \{c_1, c_2, \dots, c_x\}$) and physical capability of the robots ($F(\alpha_i) = \{f_1, f_2, \dots, f_y\}$) and the social context Φ . This relation is expressed by Eq. 5.

Even in this simplified way, we need to raise the concepts that involve the task itself. We must notice the necessary elements for its execution, no matter what the robot is responsible for executing. In this case, the task needs a series of basic knowledge that the robot needs to incorporate to its set, in order to physically execute it. Namely, if $C(\tau_j) = \{c_1, c_2, \dots, c_x\}$ and $F(\tau_j) = \{f_1, f_2, \dots, f_y\}$, then Eq. 6 verifies if task τ_j can or cannot be executed by one or more robots. If $y = 1$, then τ_j can be executed by a group of robots (or a subset of A , defined by A'). However, if $y = 0$, there is no group of robots able to solve task τ_j .

Function (6) is similar to Eq. 4 for a single robot, however evaluating a group (or all of them). With this in mind and accepting that in a multi-robot system there might always be cooperation into the group, when $f_{\text{-}egr}(\tau_j, A') = 1$ this implies to assert that $\{\bigcup_{\alpha_i \in A'}^n C(\alpha_i)\} \subseteq C(\tau_j)$ and $\{\bigcup_{\alpha_i \in A'}^n F(\alpha_i)\} \subseteq F(\tau_j)$.

$$\rho(\alpha_i) = (C(\alpha_i), F(\alpha_i), \Phi) \tag{5}$$

$$f_{\text{-}egr}(\tau_j, A') = y \tag{6}$$

By modifying Eq. 5 and 6 according to the response of a group of robots for a given task, we obtain the relations (7) and (8), respectively. These new Equations allow to get a response to the stimulus of a task by a set of robots cooperating in their physical capabilities and transmitting their knowledge. To better understand this situation, let's suffice to suppose that the robot does not have knowledge but has enough physical capability to execute a given task. It can execute this task despite the fact of being helped by another robot that will make the necessary knowledge available to it.

$$\rho(A'_t, \tau_j) = \{\{\bigcup_{\alpha_i \in A'_t}^n C(\alpha_i)\}, \{\bigcup_{\alpha_i \in A'_t}^n F(\alpha_i)\}, \Phi\} \tag{7}$$

$$f_{\text{-}egr}(A'_t, \tau_j) = \{\bigcup_{\alpha_i \in A'_t}^n C(\alpha_i)\} \cup \{\bigcup_{\alpha_i \in A'_t}^n F(\alpha_i)\} \tag{8}$$

4.4 Freedom of Action

According to the humanistic approach, humans control their actions and have freedom of action taking their potential into account. In the case of robots, they are free to take decisions accordingly with the implementation of some rules of the environment that must be motivational for an efficient cooperation. However, there is no formal structure for these rules because they are inlaid in the conceived environment.

5 IDeM-MRS Formalized

At this point, once most of the theory behind the learning process is understood, the mathematical formulation for the IDeM-MRS is introduced next. Consider that:

- $A = \{\alpha_1, \alpha_2, \dots, \alpha_m\}, \forall m \in \mathbb{N}$, is a set of robots of the environment, in the instant t ;
- $T = \{\tau_1, \tau_2, \dots, \tau_n\}, \forall n \in \mathbb{N}$, is a set of tasks to be solved in the instant t ;
- Φ is the social context of the environment;
- $C(\alpha_i) = \{c_1, c_2, \dots, c_x\}, \forall x \in \mathbb{N}$, is the set of knowledge of the robot α_i ;
- $C(\tau_j) = \{c_1, c_2, \dots, c_x\}, \forall x \in \mathbb{N}$, is the set of knowledge necessary to perform the task τ_j ;

- $F(\alpha_i) = \{f_1, f_2, \dots, f_y\}, \forall y \in \mathbb{N}$, is the set of physical capabilities of the robot α_i ;
- $F(\tau_j) = \{f_1, f_2, \dots, f_y\}, \forall y \in \mathbb{N}$, is the set of physical capabilities necessary for a robot to perform the task τ_j .

The objective function is described by Eq. 9, such that $C(\alpha_i)$ is the set of knowledge of the robot α_i , t is the evaluated instant and $m \in \mathbb{N}$.

$$\max \left(\sum_{i=1}^m \frac{C(\alpha_i)}{t} \right) \quad (9)$$

However, the restrictions are:

- 1) $f_zpd(\alpha_i, \tau_j) = f_lrd(\alpha_i, \tau_j) - f_lpd(\alpha_i, \tau_j)$
- 2) $f_lrd(\alpha_i, \tau_j) = 1 \Rightarrow f_e(\tau_j, \alpha_i) = 1$
- 3) $f_e(\tau_j, \alpha_i) = 0 \Rightarrow f_lrd(\alpha_i, \tau_j) = 0$
- 4) $\rho(A'_i, \tau_j) = \{\bigcup_{\alpha_i \in A'_i} C(\alpha_i)\} \cup \{\bigcup_{\alpha_i \in A'_i} F(\alpha_i)\}$

The indicators of i) Level of Real Development, ii) Level of Potential Development, iii) Zone of Proximal Development, and iv) Stimulus of a task for an agent, are:

- i) $f_lrd(\alpha_i, \tau_j) = 0$, if α_i did not know how to solve τ_j alone; 1, otherwise;
- ii) $f_lpd(\alpha_i, \tau_j) = -1$, if α_i is able to learn τ_j ; 1, otherwise;
- iii) $f_zpd(\alpha_i, \tau_j) = -1$, if α_i does not know and is unable to learn τ_j ; 0, if α_i knows τ_j , but cannot develop its solution; 1, if α_i does not know but is able to learn τ_j ; and 2, if α_i already knows τ_j and is even able to develop its solution;
- iv) $f_e(\tau_j, \alpha_i) = 1$, if α_i knows how to solve τ_j ; 0, otherwise.

5.1 Cooperation Rules for the IDeM-MRS

By analyzing the states that the robots can assume and before the solicitation of a solution for a given task of the Mission, some functionality models are modeled. However, for practical applications, it is not always possible to obtain real solutions [15] due to several reasons, like for example incorrect modeling of the target problem or insufficient time to find an ideal solution. The solution method for the Cooperative Execution of Task Problems (CETP) proposed by the Intellectual Development Model for Multi-Robot

Systems - IDeM-MRS, deals with these limitations through the execution of its cooperative rules allowing the robots to cooperate in order to accomplish different parts of the Mission until finalizing its complete execution.

The IDeM-MRS combines routines for updating the multi-robot system, on-line task allocation and, finally, executing tasks of the mission. These components are integrated, driven to execution of efficient solutions to multi-robot systems in solving CTEP, associating the benefit of knowledge acquisition by the robots. The modules that are part of these components as well as the information flow among them are described in Fig. 6. This model, called Framework, is abridged here from Talay's work [21], adapted in this research in order for the inclusion of specific elements inherent to the IDeM-MRS proposal.

The framework combines functional modules useful for updating the instances of the multi-robot environment, for verifying data consistency, for applying coexistence rules to select the tasks and assigning them to the robots, and for executing the mission.

Information about the MRS can be obtained from some external agent (through the *Communication Layer*), from a robot of the environment (through the *Perception Layer*), or through modifications performed by the *Task Selector* and *Execution* modules (through the *Communication Layer*).

The *Consistence Verification* module is responsible for starting warning procedures to inform environment data inconsistency, if necessary. This module also performs the correction of inconsistency, possibly. In a situation where all data of the environment is consistent, this module reports, to the *Task Selector*, that its execution can start.

The *Task Selector* is the module responsible for the execution of cooperation rules, obtained by evaluating the social learning models (SLM) in order to allocate the tasks of the mission to the most appropriate robots. Hence, it is in this module that the IDeM-MRS is implemented.

Finally, the *Execution* module implements the execution of the Mission, in a synchronized and cooperative manner, according to definitions established by the *Task Selector*.

The components *Communication Layer*, *Perception Layer* and *Operation Layer* have the following

objectives, respectively: to allow communication among modules, to receive the configuration of the environment through perceptions, and to execute the mission through commanding the physical elements responsible for such functions in the robots.

An example of information flow for an MRS solving a CTEP, when exposed to the proposal of this research, is illustrated in Fig. 6, expressly following a numerical order, and described below:

1. Initially, the *Update* module receives the specification of the MRS in terms of defining the environment, the robots and the general objective (mission).
2. The specification of the environment is sent to the *Consistence Verification* module, which evaluates it, trying to detect inconsistent data.
3. If there is no data inconsistency, the *Task Selector* module performs, according to rules defined by the IDeM-MRS, the selection of the ideal robot to solve each expected task. At this point, conflicts are not always solved through rules imposed by

the IDeM-MRS and the modification of knowledge in the robots (acquisition and assimilation) is made effective, and sent to update the environment, through the *Communication Layer*.

4. The *Execution* module finishes the mission, following the robot/task selection defined by the IDeM-MRS in the *Task Selection* module. It reports an execution list to the *Operation Layer*, besides informing to the environment about any update and confirming that the mission is completely executed.

The IDeM-MRS design is capable of dealing online and in real-time with task solving situations. The modeling can efficiently show each interaction among modules, as well as the information flow going through the system. The objective of a team of robots $A = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$ is to execute a list of tasks $T = \{\tau_1, \tau_2, \dots, \tau_n\}$ (tasks making part of a specific M mission), always trying to maximize the set of knowledge of the robots $C(\alpha_i)$ ($\forall 0 < x \leq m$) and minimizing the total execution time of M .

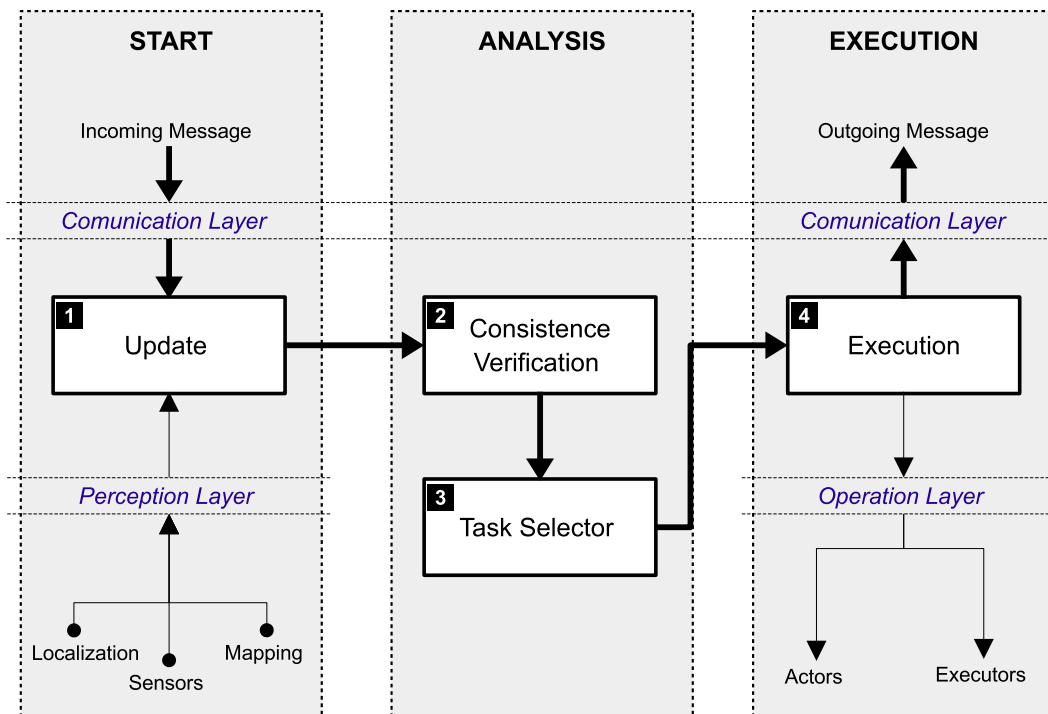


Fig. 6 Framework with functional modules *Update*, *Consistence Verification*, *Task Selector* and *Execution*, and information flow for a multi-robot system solving a cooperative execution of task problem

5.1.1 Important Definitions

Our formalism is developed to be adaptable to any multi-robot environment. Ideally, the robots in the environment can assume any configuration ranging from a set of robots without external interference and without specification of coordination to a robot with features of leader or provider of permission referring to the remaining robots of the group. Each robot, at the moment of the request of a task, is active in some state. This state characterizes it before the group, being able to assume values distinguishing it in terms of aptitude, execution and instruction. These states are specified as:

- *Able*: it possesses all the knowledge and physical capabilities that are needed to solve a given task;
- *Strong Executor*: it has all the physical capabilities needed to solve the task and even some knowledge, but not all;
- *Weak Executor*: it has all the physical capabilities needed for solution, but no knowledge;
- *Strong Instructor*: it has all the knowledge needed to solve the task, but has no physical capability to execute it;
- *Weak Instructor*: it has some knowledge about solving the task, but not all, but has no physical capability to execute it;
- *Unable*: it does not have any knowledge nor physical capability to execute the task.

In order to formalize these concepts, the definitions of these possible states are presented below considering that the robot $\alpha_i = \{C(\alpha_i), F(\alpha_i)\}$ is being requested to solve a task $\tau_j = \{C(\tau_j), F(\tau_j)\}$.

Definition “*Able*”: the robot α_i is considered *Able* to solve task τ_j if, and only if, $C(\tau_j) \subset C(\alpha_i)$ and $F(\tau_j) \subset F(\alpha_i)$.

Definition “*Strong Executor*”: the robot α_i is considered a *Strong Executor* of task τ_j if, and only if, $C(\tau_j) \setminus C(\alpha_i) = X$, such that $X \neq \emptyset$, $X \neq C(\tau_j)$ and $F(\tau_j) \subset F(\alpha_i)$.

Definition “*Weak Executor*”: the robot α_i is considered a *Weak Executor* of task τ_j if, and only if, $C(\tau_j) \subset C(\alpha_i) \setminus C(\tau_j) = C(\tau_j)$, or otherwise, $C(\tau_j) \cap C(\alpha_i) = \emptyset$, and $F(\tau_j) \subset F(\alpha_i)$.

Definition “*Strong Instructor*”: the robot α_i is considered a *Strong Instructor* of task τ_j if, and only if, $C(\tau_j) \subset C(\alpha_i)$ and $F(\tau_j) \not\subset F(\alpha_i)$.

Definition “*Weak Instructor*”: the robot α_i is considered a *Weak Instructor* of task τ_j if, and only if, $C(\tau_j) \setminus C(\alpha_i) = X$, such that $X \neq \emptyset$, $X \neq C(\tau_j)$ and $F(\tau_j) \not\subset F(\alpha_i)$.

Definition “*Unable*”: the robot α_i is considered *Unable* to solve task τ_j if, and only if, $C(\tau_j) \setminus C(\alpha_i) = C(\tau_j)$, or otherwise, $C(\tau_j) \cap C(\alpha_i) = \emptyset$ and $F(\tau_j) \not\subset F(\alpha_i)$.

The differences of the robots belonging, at a given time, to each of those states can be better visualized in Table 1 that specifies the existence or not, of the concepts related to the execution of a task: previous knowledge (total or partial) and physical capability to execute a task. These states are not static. The robot may evolve to another state depending on the flow of information (previous knowledge) which is passed to it at each moment.

Figure 7 shows a machine of states X describing the state transitions, which the robot may be subjected to depending on the needs of the moment. For each effective transition, the robot bears a new state. At start, all of the robots are in state A . After the evaluation in terms of necessary partial or total knowledge and physical capability, the robots reach the final states D , E , H , I , J or L .

The operations happen in the following way: at the moment when the environment is invoked to solve a task from the list, it automatically evaluates the robots and drives them to the states described above. After this phase, the environment decides which robot should be executor of the task. This choice makes the robots to pass to the *Executor* state. The flux of this transformation is shown in Fig. 8.

To be defined as an *Executor*, the robot needs to be in one of three possible conditions:

1. To be an *Able* robot, available in the environment (i.e. being free from executing any other task);
2. To be a *Strong Executor* robot, to then receive some subset of knowledge from others, being them a *Weak Instructor* or a *Strong Instructor*;
3. To be a *Weak Executor* robot, to then receive the total knowledge from others, being them a

Table 1 Aptitudes of the robots when asked to solve a task

State	Total knowledge	Partial knowledge	Being capable
Able	x	-	x
Strong Executor	-	x	x
Weak Executor	x	-	x
Strong Instructor	-	-	-
Weak Instructor	-	x	-
Unable	-	-	-

“Total knowledge” reports that the robot has all the necessary knowledge about how to solve the task. “Partial knowledge” specifies that the robot has some knowledge, but not all. And “Being capable” indicates that the robot has the physical capability required to solve the task

Strong Instructor or an *Able* robot occupied at the moment. In these two last cases, after the *Strong Executor* or *Weak Executor* robot receives the proper amount of knowledge (partial or total, respectively), it may become an *Able* robot, available to solve the aforesaid task. This way, he can,

without any exchange of knowledge, assume the state of *Executor* and execute the given task.

Algorithm 1 explains how to obtain the state of the robot if required by some task of the mission. An important fact to be verified is that a situation

Fig. 7 Machine of states X for robots during a request to execute a given task. X has an initial state A and final states D, E, H, I, J and L . The specification of this machine is $X = \langle \{A, B, C, D, E, F, G, H, I, J\}, A, \{yes, no\}, \{D, E, H, I, J, L\}, \delta \rangle$, with transitions δ as specified in the diagram

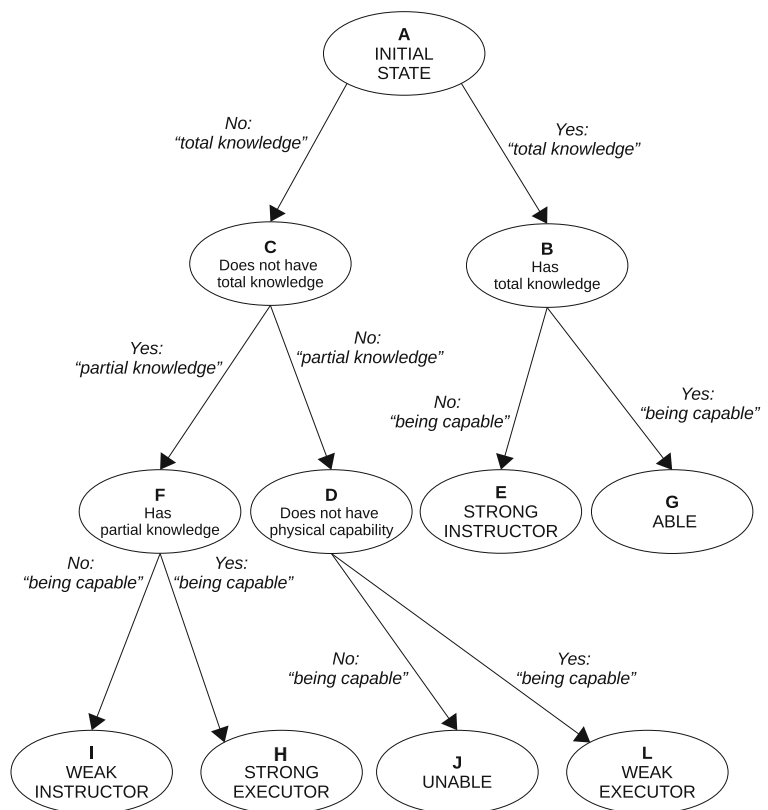
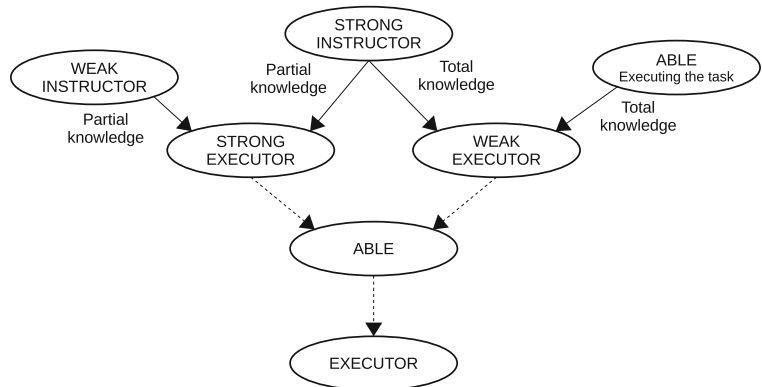


Fig. 8 Data flow applied to modify the states and to end with the state of *Executor*



where the task needs some physical feature that any of the robots have may occur. In this case, this task is called impossible, because no robot is able to execute it. Considering an environment, with a group of robots $A = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$ such that $\alpha_i = \{C(\alpha_i), F(\alpha_i)\}$, the definition for verifying the existence of impossible tasks is described below.

Definition Impossible Task: task τ_j is considered *Impossible* if, and only if, $F(\tau_j) \not\subset \bigcup_{i=1}^n F(\alpha_i)$, where the number of robots of the group is n .

Algorithm 1: Get robot state.
Input: τ_j of M and R
Output: R

```

1: begin
2:   Able = 0;
3:   Strong_instructor = 0;
4:   Weak_instructor = 0;
5:   Strong_executor = 0;
6:   Weak_executor = 0;
7:   Unable = 0;
8:   robots = R;
9:   if  $f_{\text{egr}}(R, \tau_j) \neq 0$  then
10:    return Impossible_task;
11:  else
12:    while robot  $\neq 0$  do
13:       $r = \text{Select\_robot}(robots)$ ;
14:      if  $f_{\text{Ird}}(r, \tau_j) == 1$  then
15:        Able = Able  $\cup$  {r};
16:      else
17:        if  $f_{\text{Ipd}}(r, \tau_j) == 1$  then
18:          if  $f_{\text{er}}(\tau_j, r) == 0$  then
19:            Weak_executor = Weak_executor  $\cup$  {r};
20:          else
21:            Strong_executor = Strong_executor  $\cup$  {r};
22:          end
23:        else
24:          if  $f_{\text{er}}(\tau_j, r) == 0$  then
25:            Unable = Unable  $\cup$  {r};
26:          else
27:            if  $f_{\text{ec}}(\tau_j, r) \supset C(\tau_j)$  then
28:              Strong_instructor = Strong_instructor  $\cup$  {r};
29:            else
30:              Weak_instructor = Weak_instructor  $\cup$  {r};
31:            end
32:          end
33:        end
34:      end
35:      robots = robots - {r};
36:    end
37:  end
38:  return 1;
39: end
    
```

5.1.2 Representation of the Environment

The mission in the IDeM-MRS is represented by an ordered list of tasks, $M = \{\tau_1, \tau_2, \dots, \tau_m\}$. The tasks are represented as quartets with the necessary information for its execution and for the perception of their status in the environment: $\langle id, reqc, recf, info \rangle$. The field *id* is a unique value identifying the task; *reqc* represents the set of knowledge required to execute the task; *recf* represents the set of physical features required to execute the task; *info* indicates the status of the task at a given moment, in the case that it is being executed by a robot or in the case that it is in standby. Table 2 shows an example with two instances of this representation.

The representation of a group of robots is not different from the previous one. It is represented as a set of robots that must work in the environment, namely $R = A = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$, thus robots are also represented as quartets $\langle id, reqc, recf, info \rangle$, were: *id* is a unique value identifying the robot; *reqc* represents a set of knowledge belonging to the robot; *recf* represents the set of physical features of the robot; *info* indicates the status of the robot at a moment, if he is

Table 2 An example with some instances of task representations

id	reqc	reqf	info
T58	{c5}	{f1, f2}	0
T14	{c4, c6}	{f2, f5, f8}	1

Task $T14$ being performed by a robot ($info=1$). $C(T14)=\{c4, c6\}$ is the set of skills needed and $F(T14) = \{f2, f5, f8\}$ is the set of required physical characteristics

executing a task (dedicated) or not. Table 3 shows an instantiation example.

5.1.3 Task Selector

Initially, a set of tasks $T = \{\tau_1, \tau_2, \dots, \tau_m\}$ (and their respective features $\tau_j = \{C(\tau_j), F(\tau_j)\}$) are ordered in a list to form the mission M of the environment. Some issues are important:

1. There is a group of robots working, $R = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$, also with their features $\alpha_i = \{C(\alpha_i), F(\alpha_i)\}$.
2. Each task is verified, ordered, as the mission is reported to the environment. That is, the task τ_i will be attended previously to τ_j if and only if $i < j$.
3. For each instance of a group of robots, different or not of a new mission, the task execution list, called *list_exec*, will be different.
4. Meanwhile, the *list_exec* will also show a different solution sequence for the same mission and the same group of robots since it is possible that there are robots with the same knowledge or the same physical capability.
5. Following the ideas of this research that are based on the investigation of social learning models, the execution of Algorithm 2 reports the list of task selection by the robots (*list_exec*) from the mission offered to an environment containing a group of robots.
6. The list of task selection by the robots (*list_exec*) contains each task t_j associated to the respective robot r_i robot which will execute it, in the specified order.
7. M is the mission accomplished by the tasks $T = \{\tau_1, \tau_2, \dots, \tau_n\}$ in the sequence that must be invoked for execution, where $\tau_j = \{C(\tau_j), F(\tau_j)\}, \forall 0 < j \leq \|n\|$.

Table 3 An example of representations of robots

id	reqc	reqf	info
R08	{c ₁ , c ₂ , c ₆ }	{f ₁ , f ₂ }	0
R05	{c ₅ }	{f ₂ , f ₅ }	1

Robot R08 is idle (*info*=0). $C(R08) = \{c_1, c_2, c_6\}$ is the set of basic knowledge needed and $F(R08) = \{f_1, f_2\}$ is the set of physical characteristics

8. R is a group of robots working in the environment ($R = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$, such that $m \in \mathbb{N}$), where $\alpha_i = \{C(\alpha_i), F(\alpha_i)\}, \forall 0 < i \leq \|m\|$.

The function *Get_robot_state()* refers to Algorithm 1. In this Algorithm, the function *Select_robot()* refers to the choice methodology of a robot among the set specified as parameter. At this point, this choice is set randomly but some heuristics may be applied to objectively enhance the choice.

Algorithm 2: Task Selector of IDEM-MRS.

```

Input: M, R
Output: list_exec
1: begin
2:   list_exec = 0;
3:   tasks = M;
4:   while tasks ≠ 0 do
5:     t = Select_task(tasks);
6:     Get_robot_state(R,t);
7:     if Able ≠ 0 then
8:       r = Select_robot(Apto);
9:       list_exec = list_exec ∪ {(t,r)};
10:    else
11:      if Strong_executor ≠ 0 then
12:        r = Select_robot(Strong_executor);
13:        if Strong_instructor ≠ 0 then
14:          r_inst = Select_robot(Strong_instructor);
15:          f_assimilation(r,r_inst,t);
16:          list_exec = list_exec ∪ {(t,r)};
17:        else
18:          if Weak_instructor ≠ 0 then
19:            A' = get_robot_set();
20:            if f_egr(t,A') == 1 then
21:              f_assimilation(r,A',t);
22:              list_exec = list_exec ∪ {(t,r)};
23:            end
24:          end
25:        end
26:      else
27:        if Weak_executor ≠ 0 then
28:          r = Select_robot(Weak_executor);
29:          if Strong_executor ≠ 0 then
30:            r_inst = Select_robot(Strong_instructor);
31:            f_assimilation(r,r_inst,t);
32:            list_exec = list_exec ∪ {(t,r)};
33:          else
34:            if Weak_instructor ≠ 0 then
35:              A' = Get_robot_set();
36:              if f_egr(t,A') == 1 then
37:                f_assimilation(r,A',t);
38:                list_exec = list_exec ∪ {(t,r)};
39:              end
40:            end
41:          end
42:        end
43:      end
44:    end
45:    tasks = tasks - {t};
46:  end
47:  return list_exec;
48: end

```

6 Experiments and Results

A set of experiments was planned and implemented for verifying the consistence of the proposed mathematical formalism and also to test performance and

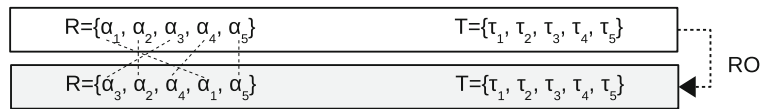


Fig. 9 Example of the exploratory criterion RO. *Input* list of robots $R = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}$. *Output* (after the operation): list of robots modified available for communication $R = \{\alpha_3, \alpha_2, \alpha_4, \alpha_1, \alpha_5\}$

other related issues. The first experiment includes the simulation of an environment with 10 heterogeneous robots in it, which are required to solve a set of 10 tasks, here called a Mission. This experiment was implemented using the software Scilab [4], which is a very nice tool for solving numerical problems. Simulated in an environment with 10 heterogeneous robots (r_1, \dots, r_{10}) which may be required to execute Missions (groups of 10 different tasks, like t_1, \dots, t_{10}), the mathematical presuppositions are evaluated through different tests, which will be reported later. For all of the 6 tests, 5 samples were obtained from 100 executions, making a total of 3.000 evaluated results.

6.1 Exploratory Criteria

As a way to evaluate all the parameters of the instance, three exploratory criteria were adopted from the initial configuration Environment/Robots/Mission, which randomly modifies some parts integrating the environment. In this case, the effect is like making a rearrangement of the working space.

The first one modifies the list of robots available for communication (Fig. 9); the second one modifies the line of tasks of the mission given to be accomplished in the environment at the beginning of the execution (Fig. 10); and the third one transforms both the list of robots and the line of tasks of the mission (Fig. 11).

6.2 Nomenclature of Test Instances

As we deal with a large number of evaluated instances, we define a way of identifying each test coding in a map, a nomenclature for each one as $\langle \text{robot, task, executions, exploratory criterion, methodology} \rangle$. Figure 12 shows the map. To better understand, the example 10R10T_100_MI_N1 is an instance evaluated by the features such that:

- 10R: 10 heterogeneous robots;
- 10T: the robots being required to execute a mission containing 10 different tasks;
- 100: in a cycle of 100 executions;
- MI: using the exploratory criterion of randomly modifying the mission in each cycle. In this case, the exploratory criteria are MI, RO or MR;
- N1: for the algorithm without learning in its first sample. Other methodology is Sx that indicates the execution using the IDeM-MRS model, in its x-th sample;

6.3 Experiments Without Cooperation

The first test is a simulation of the environment without any action policy among the robots inside it. The Algorithm 3 named *Selector_tasks_without_cooperation* defines the main steps for the experiment, and defines a list of robots to execute

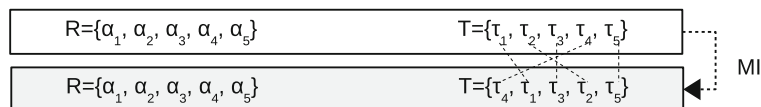


Fig. 10 Example of the exploratory criterion MI. *Input* list of tasks $T = \{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}$. *Output* (after the operation) list of tasks of the mission modified given to the environment $T = \{\tau_4, \tau_1, \tau_3, \tau_2, \tau_5\}$

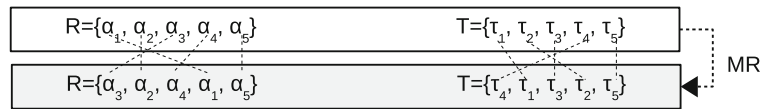


Fig. 11 Example of the exploratory criterion MR. *Input* list of robots $R = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}$ and list of tasks $T = \{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}$. *Output (after the operation)* list of robots

modified available for communication $R = \{\alpha_3, \alpha_2, \alpha_4, \alpha_1, \alpha_5\}$ and list of tasks of the mission modified given to the environment $T = \{\tau_4, \tau_1, \tau_3, \tau_2, \tau_5\}$

the tasks without the use of any improvement in the assignment of responsibilities for each task.

```

Algorithm 3: Selector_tasks_without_cooperation.
Input:  $M, R$ 
Output:  $list\_exec$ 
1: begin
2:    $list\_exec = 0;$ 
3:    $tasks = M;$ 
4:   while  $tasks \neq \emptyset$  do
5:      $t = Select\_task(tasks);$ 
6:      $Aptos = Get\_able(R, t);$ 
7:     if  $Aptos \neq \emptyset$  then
8:        $r = Select\_robot(Aptos);$ 
9:        $list\_exec = list\_exec \cup \{(t, r)\};$ 
10:    else
11:      return  $Impossible\_task;$ 
12:    end
13:     $tasks = tasks - \{t\};$ 
14:  end
15:  return  $list\_exec;$ 
16: end
    
```

A mission M and a set of robots R are required as input parameters, and the output is a list $list_exec$, which specifies exactly which robots must perform the tasks of the mission. The result is the same for all the samples, of all the instances following the 3 exploratory criteria (RO, MI and MR). We observe in the 1500 results obtained that there is not any kind of cooperation and, at the same time, any learning during the execution of the tasks. The results obtained with the implementation of this algorithm are shown in Tables 4 and 5.

As it can be seen, the approach without cooperation does not allow to acquire knowledge in the robots that begin with an amount of knowledge. After execution, each instance analyzed offers no changing in this setting. At the end, the robots have exactly the same knowledge of the beginning of the experiment. All the 1500 execution show no changes.

6.4 Experiments with IDeM-MRS

The second test is a simulation of the environment with the rules of the model IDeM-MRS for the same instances. Algorithm 4 defines the main steps for the experiment. The list of robots engaged to execution of the tasks agree with the model in the assignment of responsibilities for each task. Similarly to the previous experiment, a mission M and a set of robots R are required as input parameters, and the output is a list $list_exec$, which specifies exactly which robots must perform the tasks of the mission.

```

Algorithm 4: IDeM-MRS main.
Input:  $M, R$ 
Output:  $list\_exec$ 
1: begin
2:    $Update\_environment(M, R, T);$ 
3:   if  $Check\_inconsistencies(M, R, T) == 0$  then
4:      $h = Get\_nomenclature();$ 
5:      $Apply\_nomenclature(h);$ 
6:      $list\_exec = Selector\_tasks(M, R);$ 
7:      $Execute(list\_exec);$ 
8:     return 1;
9:   else
10:    return 0;
11:  end
12: end
    
```

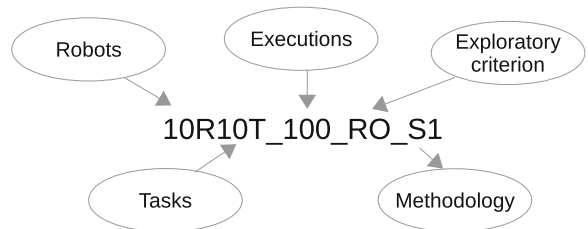


Fig. 12 An example of the map of the nomenclature with 10 heterogeneous robots being required to execute a mission containing 10 different tasks in a cycle of 100 executions. The experiment is done using the exploratory criterion of randomly modifying the list of robots in each cycle, for the algorithm with learning, in its first sample

Table 4 Result considering the tasks execution and the exchange of messages between the environment, without IDeM-MRS

Tests		Tasks		Messages	
Nomenclature	Instances	Of mission (number)	Performed (%)	To the environment (number)	By answers (second)
10R10T_100_RO	N1	1000	40,0	600	34
	N2	1000	40,0	600	35
	N3	1000	40,0	600	33
	N4	1000	40,0	600	33
	N5	1000	40,0	600	33
10R10T_100_TA	N1	1000	40,0	600	37
	N2	1000	40,0	600	37
	N3	1000	40,0	600	35
	N4	1000	40,0	600	36
	N5	1000	40,0	600	33
10R10T_100_RT	N1	1000	40,0	600	45
	N2	1000	40,0	600	43
	N3	1000	40,0	600	42
	N4	1000	40,0	600	42
	N5	1000	40,0	600	40

Table 5 Result considering the acquisition of knowledge, without IDEM-MRS

Tests		Knowledge total			
Nomenclature	Instances	Of environment (number)	Inicial (number)	Final (number)	acquired (%)
10R10T_100_RO	N1	40	13	13,0	0,0
	N2	40	13	13,0	0,0
	N3	40	13	13,0	0,0
	N4	40	13	13,0	0,0
	N5	40	13	13,0	0,0
10R10T_100_TA	N1	40	13	13,0	0,0
	N2	40	13	13,0	0,0
	N3	40	13	13,0	0,0
	N4	40	13	13,0	0,0
	N5	40	13	13,0	0,0
10R10T_100_RT	N1	40	13	13,0	0,0
	N2	40	13	13,0	0,0
	N3	40	13	13,0	0,0
	N4	40	13	13,0	0,0
	N5	40	13	13,0	0,0

Table 6 Result considering the tasks execution and the exchange of messages between the environment, with IDeM-MRS

Tests		Tasks	Messages		
Nomenclature	Instances	Of mission (number)	Performed (%)	To the environment (number)	By answers (second)
10R10T_100_RO	S1	1000	90,0	100	69
	S2	1000	90,0	100	76
	S3	1000	90,0	100	87
	S4	1000	90,0	100	69
	S5	1000	90,0	100	72
10R10T_100_TA	S1	1000	93,1	69	81
	S2	1000	93,0	70	92
	S3	1000	93,4	66	94
	S4	1000	93,0	70	85
	S5	1000	93,8	62	87
10R10T_100_RT	S1	1000	94,6	54	81
	S2	1000	94,3	57	87
	S3	1000	93,9	61	89
	S4	1000	94,6	54	90
	S5	1000	93,9	61	94

Table 7 Results considering the acquisition of knowledge, with IDEM-MRS

Tests		Knowledge total			
Nomenclature	Instances	Of environment (number)	Initial (number)	Final (number)	Acquired (%)
10R10T_100_RO	S1	40	13	17,6	17,0
	S2	40	13	17,5	16,7
	S3	40	13	17,4	16,3
	S4	40	13	17,1	15,2
	S5	40	13	17,5	16,7
10R10T_100_TA	S1	40	13	19,7	24,8
	S2	40	13	19,6	24,4
	S3	40	13	19,6	24,4
	S4	40	13	19,7	24,8
	S5	40	13	19,7	24,8
10R10T_100_RT	S1	40	13	17,6	17,0
	S2	40	13	17,7	17,4
	S3	40	13	17,9	18,1
	S4	40	13	17,6	17,0
	S5	40	13	18,0	18,5

Table 8 Average knowledge gain of each robot which completed the process execution of the experiments with the rules of IDeM-MRS, according to each instance

Identification of robot	Knowledge Initial	Knowledge acquired		
		Test 1	Test 2	Test 3
r1	1	1.0	1.0	1.0
r2	0	1.4	4.0	1.2
r3	1	1.1	1.0	1.2
r4	2	2.0	2.0	2.0
r5	1	2.2	1.0	2.3
r6	2	2.0	2.3	2.1
r7	1	1.1	2.3	1.2
r8	0	1.5	1.0	1.6
r9	2	2.1	2.0	2.1
r10	3	3.1	3.0	3.2

The initial configuration of the environment is defined as in the experiment without our model. The results obtained with the implementation of this algorithm (with IDeM-MRS) are shown in Tables 6 and 7.

6.4.1 Evaluation of the Results Compared to the Gain of Knowledge for the Environment

Compared to the gain of knowledge acquired by the environment, when using IDeM-MRS, there was knowledge acquisition, which characterizes learning. At the end of the tests, the robots have more knowledge than at the beginning of the process.

Table 8 shows the average of knowledge with which each robot finishes the process. Each instance has

improved its initial condition. The tests performed allow to obtain objective data in terms of the knowledge acquired by the environment when using the IDeM-MRS.

This experiment is the first step towards proofing the effectiveness of the model. It works in practice in this performed test. Instances with the formalism show gains of up to 24.7 % in knowledge. In comparison, in the environment without communication among the robots, the result is the same for all the samples of all the instances. Following the exploratory methods described above, there is no cooperation nor learning during the execution of tasks in the results. Table 9 reports data calculated from the results of each instance (seen in Fig. 13).

Table 9 Gain, expressed as percentage of knowledge units, obtained by the mean of the execution of the evaluated instances

Instance	Gain	
Without IDeM-MRS	10R10T_100_RO_N	0,0 %
	10R10T_100_TA_N	0,0 %
	10R10T_100_RT_N	0,0 %
With IDeM-MRS	10R10T_100_RO_S	16,4 %
	10R10T_100_TA_S	24,7 %
	10R10T_100_RT_S	17,6 %

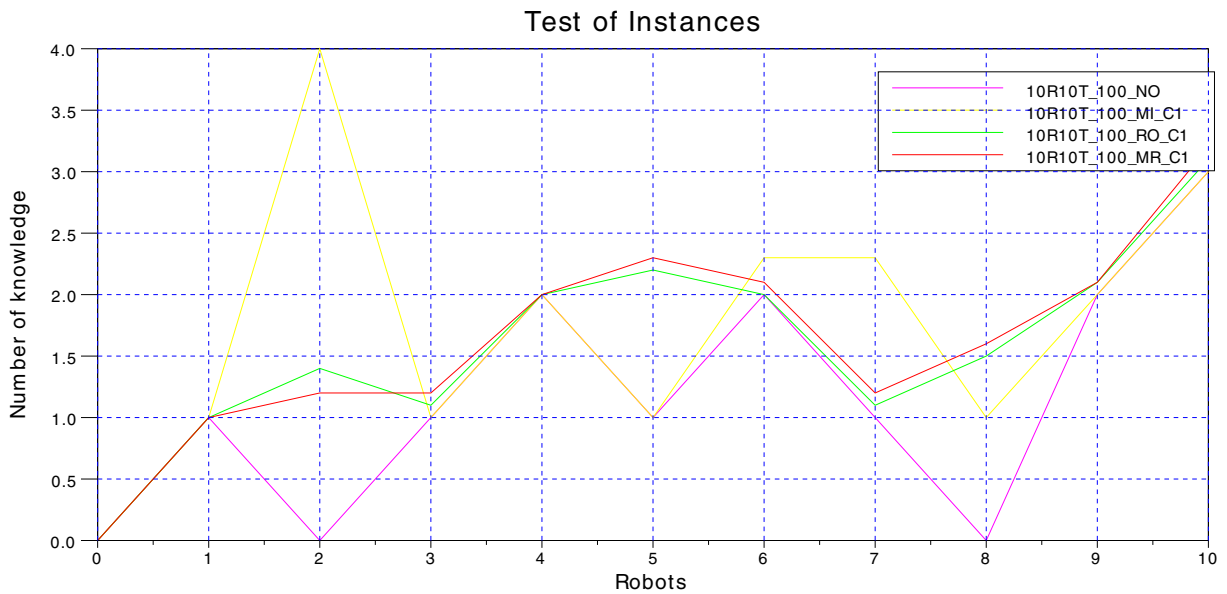


Fig. 13 Average of the 5 execution samples. At the end, the robots (r) have value of knowledge as follow. 10R10T_100_NO: r1=1, r2=0, r3=1, r4=2, r5=1, r6=2, r7=1, r8=0, r9=2, r10=3; 10R10T_100_MI_C1: r1=1, r2=4, r3=1, r4=2, r5=1, r6=2.3, r7=2.3, r8=1, r9=2, r10=3;

10R10T_100_RO_C1: r1=1, r2=1.4, r3=1.1, r4=2, r5=2.2, r6=2, r7=1.1, r8=1.5, r9=2.1, r10=3.1; 10R10T_100_MR_C1: r1=1, r2=1.2, r3=1.2, r4=2, r5=2.3, r6=2.1, r7=1.2, r8=1.6, r9=2.1, r10=3.2

6.4.2 Evaluation of the Results Comparing the Execution of Tasks

When not using the formalism, the system shows a fix percentage of 40 % in the achievement of tasks required for the environment mission. In opposition,

when the IDeM-MRS is applied the system shows a mean of 92,5 % of achievement. Table 10 reports data calculated from the results of each instance (seen in Fig. 14). Both were inserted into the simulated environment, without interaction with external agents.

Table 10 Tasks performed, expressed as percentage of knowledge units, obtained by the mean of the execution of the evaluated instances

Instance	Task performed	
Without IDeM-MRS	10R10T_100_RO_N	40,0 %
	10R10T_100_TA_N	40,0 %
	10R10T_100_RT_N	40,0 %
With IDeM-MRS	10R10T_100_RO_S	90,0 %
	10R10T_100_TA_S	93,3 %
	10R10T_100_RT_S	94,3 %

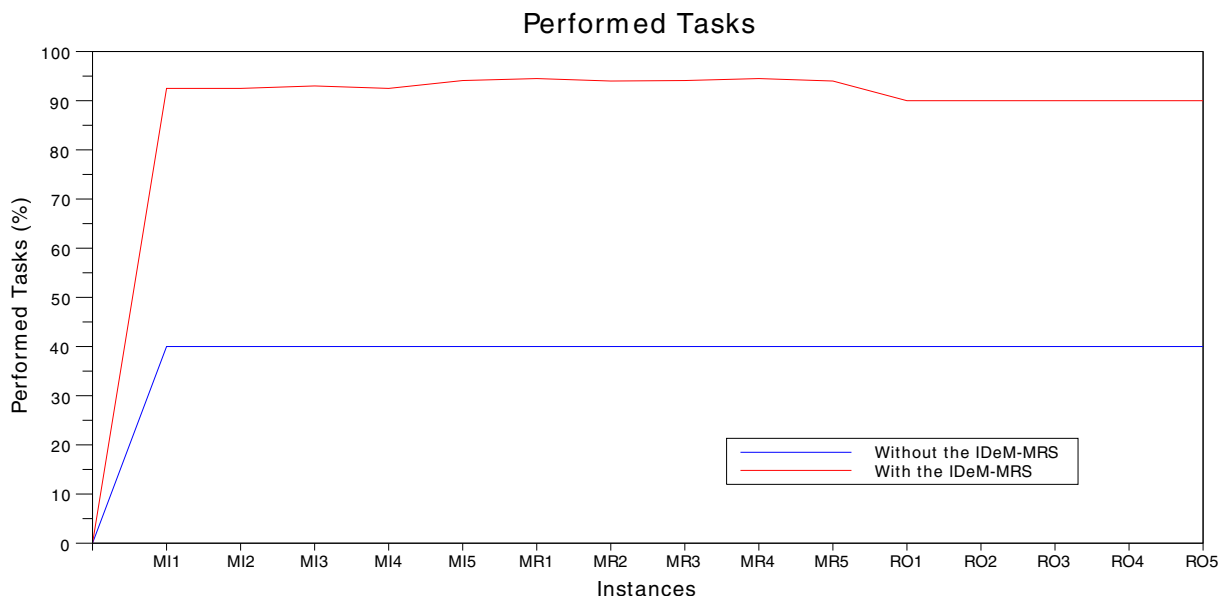


Fig. 14 Percentage of executed tasks by the environment, with the formalism and without the need of requirements external to the environment. When not using the formalism, there is a fix percentage of 40 % of achievement in the tasks required to

the environment mission, opposed when the proposed formalism is applied, which shows a mean of 92,5 % of achievement. Both were inserted into the simulated environment, without interaction with external environments

7 Conclusion

We propose a learning formalism that can be used for multi-robot systems in cooperative solution of tasks. We adopted a social approach and a communication process of the robotic agents that allows learning by the group. As seen in the results, evaluated in terms of units of knowledge acquired, we can conclude that the formalism allows the exchange of experience. And, with this, to positively influence the cooperation of the group in function of the execution of the tasks.

By evaluating the results in terms of the number of tasks executed autonomously (without external interference) by the system, it is possible to see that the formalism directs the environment, favoring freedom, knowledge and autonomy to the robots, independing of the actions of the external agents. After executing the formalism, without any constructive heuristic for choosing the most able robots to pass their experience to the rest or those deserving priority to execute a given task, the result shows knowledge gain with simple message exchange among the robots.

Even having many peculiar features of a multi-robot environment, with this research, we foresee to make possible a mathematical formalism defining

learning rules for any group of robots inserted in a closed and dynamic environment. These rules must be responsible for turning cooperation the most efficient as possible inside the group.

At the present time, constructive heuristics are being evaluated to be added for implementing a mathematical formalization. With this addition, and with some optimization techniques, we believe that decision taking will be enhanced during the communication of the group and, with this, the gain of knowledge units with experience.

This research will naturally evolve with the simulation of a real team of robots, based on a predefined space known through 3D occupation maps, restricting executions in case studies. The tests must validate the portability to any environment. In fact, in the short term, this model will be experimented with robots of our Robotics Lab in order to provide autonomy to them in the execution of cooperative tasks, as mapping using visual information. A team of 4 Turtle-bots is planned to use this approach.

Conflict of interests This article does not contain any studies with human or animal subjects. The authors declare that they have no conflict of interest.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

- Botelho, S., Alami, R.: Robots that cooperatively enhance their plans. *Proceeding of 5th International Symposium on Distributed Autonomous Robotic Systems DARS2000*, Lecture notes in Computer Science. Springer Verlag (2000)
- Brucker, P.: Scheduling and constraint propagation. *Discrete Applied Mathematics* **123**(1–3), 227–256 (2002)
- Brucker, P.: *Scheduling algorithms*. Springer (2004)
- Campbell, S.L., Chancelier, J.P., Nikoukhah, R.: *Modeling and simulation in Scilab/Scicos*. Springer Science and Business Media, New York (2006)
- Carrascosa, C., Bajo, J., Julian, V., Corchado, J.M., Botti, V.: Hybrid multi-agent architecture as a real-time problem-solving model. *Expert Syst. Appl.* **34**(1), 2–17 (2008)
- Chaimowicz, L., Sugar, T., Kumar, V., Campos, M.F.M.: An architecture for tightly coupled multi-robot cooperation. *IEEE ICRA*, 2992–2997 (2001)
- Chen, D.N., Jeng, B., Lee, W.P., Chuang, C.H.: An agent-based model for consumer-to-business electronic commerce. *Expert Syst. Appl.* **34**(1), 469–481 (2008)
- Engelberger, J.F. Health-care robotics goes commercial: the helpmate experience **11**, 517–523 (1993)
- Piaget, J.: *The equilibration of cognitive structures: the central problem of intellectual development*. University of Chicago Press, Chicago (1985)
- Kalmar, Z., Szepesvari, C., Loerincz, A.: Module-based reinforcement learning: Experiments with a real robot. *Mach. Learn.* **31**, 55 (1998)
- Kambayashi, Y., Tsujimura, Y., Yamachi, H., Takimoto, M., Yamamoto, H.: Design of a multi-robot system using mobile agents with ant colony clustering. *System Sciences, 2009. HICSS '09.42nd Hawaii International Conference on*, pp. 1–10 (2009)
- Maia, R.S., Souza, A.A.S., Gonçalves, L.M.G.: A learning process model proposal for a multi-robot environment (in portuguese). *XI Congreso de La Sociedad Peruana de Computacion* (2010)
- Mes, M., Heijden, M.V.D., Harten, A.V.: Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems **181**, 59–75 (2007)
- Piaget, J.: *The creation of cognitive structures* (in Portuguese). Zahar, Rio de Janeiro (1975)
- Reinelt, G.: *The traveling salesman: computational solutions for TSP applications*, vol. 840. Springer, Berlin (1994)
- Graf, R., Weckesser, P.: Autonomous roomservice in a hotel, pp. 641–647 (1998)
- Rooker, M.N., Birk, A.: Combining exploration and ad-hoc networking in robocup rescue **3276**, 236–246 (2005)
- Serment, L., Grabowski, R., Paredis, C., Khosla, P.: Multi-robots - the development of a framework and algorithms for a heterogeneous robot team, pp. 31–40 (2002)
- Surmann, H., Morales, A.: Scheduling tasks to a team of autonomous mobile service robots in indoor environments **8**(8), 809–833 (2002)
- Talay, S.S., Balch, T.R., Erdogan, N.: Incremental multi-robot task selection for resource constrained and interrelated tasks. *IEEE IROS*, 2314–2319 (2007)
- Talay, S.S., Balch, T.R., Erdogan, N.: A generic framework for distributed multirobot cooperation. *J. Intell. Robot. Syst.* **63**(2), 323–358 (2011)
- Toksari, M.D.: Ant colony optimization approach to estimate energy demand of turkey **35**, 3984–3990 (2007)
- Vestli, S., Tschichold-Gurman, N.: Mops, a system for mail distribution in office type buildings. *Proceedings of the first euromicro workshop on advanced mobile robot* (1996)
- Vygotsky, L.S.: *Play and its roles in mental development of child* (1967)
- Vygotsky, L.S.: *Mind and society the development of higher psychological processes*. Harvard University Press, Cambridge (1978)
- Weglarz, J., Józefowska, J., Mika, M., Waligóra, G.: Project scheduling with finite or infinite number of activity processing modes - a survey. *Eur. J. Oper. Res.* **208**(3), 177–205 (2011)