Response threshold models and stochastic learning automata for self-coordination of heterogeneous multi-task distribution in multi-robot systems

Javier de Lope [a,b], Darío Maravall [a], Yadira Quiñonez [a]

[a] Computational Cognitive Robotics Group, Department of Artificial Intelligence, Universidad Politécnica de Madrid, Madrid, Spain
[b] Department of Applied Intelligent Systems, Universidad Politécnica de Madrid, Madrid, Spain

ABSTRACT

This paper focuses on the general problem of coordinating multiple robots. More specifically, it addresses the self-selection of heterogeneous specialized tasks by autonomous robots. In this paper we focus on a specifically distributed or decentralized approach as we are particularly interested in a decentralized solution where the robots themselves autonomously and in an individual manner, are responsible for selecting a particular task so that all the existing tasks are optimally distributed and executed. In this regard, we have established an experimental scenario to solve the corresponding multi-task distribution problem and we propose a solution using two different approaches by applying Response Threshold Models as well as Learning Automata-based probabilistic algorithms. We have evaluated the robustness of the algorithms, perturbing the number of pending loads to simulate the robot's error in estimating the real number of pending tasks and also the dynamic generation of loads through time. The paper ends with a critical discussion of experimental results.

1. Introduction

Autonomous Multi-robot Systems are an outstanding applied area of Artificial Intelligence that has witnessed remarkable growth since its inception and that has developed significant progress in several applications [1]. More specifically, within Multi-robot Systems, optimal task/job allocation or assignment is an active research problem [2], in which several global or central allocation methods have been proposed so far [3,4]. Some authors have also introduced autonomous or decentralized solutions, in particular inspired in the social labor observed in some species of social insects [5,6].

In this paper we take a specifically distributed or decentralized approach as we are particularly interested in experimenting with truly autonomous and decentralized techniques in which the robots themselves are responsible for choosing the tasks in an autonomous and individual manner. Under this approach we can speak of multi-task selection instead of multi-task allocation as the agents or robots select the tasks instead of being assigned a task by a central controller.

In previous work we have already experimented with different techniques. First, we applied the well-known threshold models inspired by the labor division of social insects [7]. Afterwards, in a recently published paper [8] we applied ant colony optimization for coordinating multiple robots in the above mentioned problem of multi-task selection and compared its performance with the threshold model-based approach.

In this paper we introduce a novel approach by means of stochastic reinforcement learning algorithms based on Learning Automata theory [9] and we also compare their performance in comparison with the threshold models approach. The remainder of the paper is organized as follows: Section 2 presents the formal description of the general problem of decentralized distribution of multi-tasks/jobs in a multi-robot system, as well as a formal description of the experimental scenario. Section 3 briefly describes the response threshold models approach. Section 4 presents a brief introduction and basic definitions concerning the stochastic reinforcement learning algorithms based on Learning Automata theory. Afterwards, in Section 5 we discuss the comparative experimental results obtained in applying both methods. The paper ends with the conclusions and some remarks on future work.

2. Formal definitions

2.1. Formal description of the problem

The optimal multi-task allocation problem in multi-robot systems can be formally defined as follows: "Given a robot team

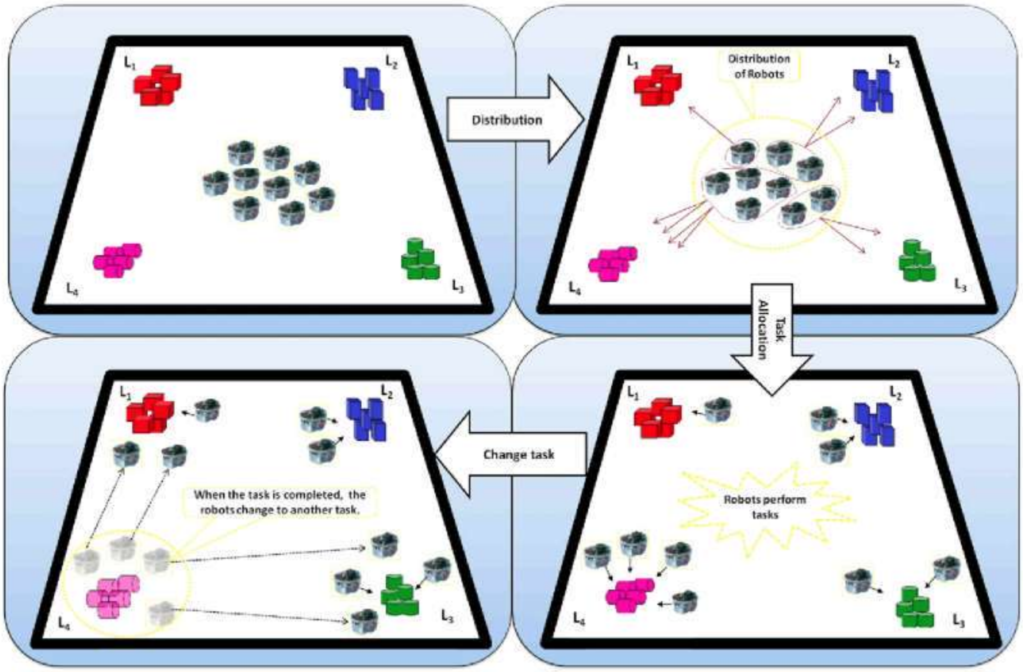

Fig. 1. Experimental scenario.

formed by N heterogeneous robots, and given K different types of heterogeneous specialized tasks or equivalently, given K different robot's roles or robot's jobs and given a particular time-dependent load or number of tasks to be executed $L = \{l_1(t), l_2(t), \ldots, l_K(t)\}$, obtain an optimal distribution of the K tasks among the N robots in such a way that the robots themselves, autonomously and in an individual manner, select a particular task such that all the existing tasks are optimally executed".

Let $L = \{l_1(t), l_2(t), \ldots, l_K(t)\}$ be the different specialized tasks. Each $l_j \in L$ has a number of j sub-tasks or pending loads. Let $R = \{r_1, r_2, \ldots, r_N\}$ be the set of N heterogeneous mobile robots. To solve the problem, we have supposed that all members $R = \{r_1, r_2, \ldots, r_N\}$ are able to participate in any sub-task l_j.

2.2. Experimental scenario

We have established the following experimental scenario (Fig. 1) in order to analyze a particular strategy or solution for the coordination of multi-robot systems as regards the optimal distribution of the existing tasks. Given a set of N heterogeneous mobile robots in a region, achieve an optimal distribution for different types of tasks. The set of N robots will form sub-teams for each type of task l_j. The sub-teams are dynamic over time, i.e. the same robots will not always be part of the same sub-team, but the components of each sub-team can vary depending on the situation.

Most of the proposed solutions in the technical literature are of a centralized nature, in the sense that an external controller is in charge of distributing the tasks among the robots by means of conventional optimization methods and based on global information about the system state [10]. However, we are mainly interested in truly decentralized solutions in which the robots themselves, autonomously and in an individual and local manner, select a particular task so that all the tasks are optimally distributed and executed. In this regard, we have experimented with response threshold models and stochastic reinforcement learning algorithms based on Learning Automata theory to tackle this hard self-coordination problem as described in the following.

3. Response threshold models

3.1. A brief introduction

Insect societies are characterized by the division of labor, communication between individuals and the ability to solve complex problems [11], and these characteristics have long been a source of inspiration and the subject of numerous studies, acquiring great relevance for many researchers both in the field of robotics as in biology. On the one hand, the biologists are trying to prove their theories of social insects on robots, and on the other hand, researchers in the discipline of robotics seek solutions to problems that cannot be solved by a single robot.

Seeley et al. [12] have considered the following experiment to study the collective behavior in a colony of insects, focusing on the work performed by bees to get honey. Two food sources are presented to the colony at 8:00 A.M. at the same distance from the hive: source A is characterized by a sugar concentration of 1.0 mol/1 and source B by a concentration of 2.5 mol/1. Between 8:00 A.M. and noon, source A has been visited 12 times and source B, 91 times. At noon, the sources are modified: source A is now characterized by a sugar concentration of 2.5 mol/1 and source B 0.75 mol/1. Between noon and 4:00 P.M., source A has been visited 121 times and source B only 10 times. It has been shown that a bee has a relatively high probability of going to a good food source and abandoning a poor food source.

3.2. Model

Based on these observations, these simple rules of behaviors allow the bees to select the best quality source; Bonabeau et al. have proposed a simple mathematical model of response thresholds for the regulation of the division of labor in insect societies [13]. In this model we assume that each task is associated with a stimulus or set of stimuli, so that individuals can detect information on each of the different stimulus intensities, and therefore, can assess the demand for a particular task when they are in contact with the associated stimulus.

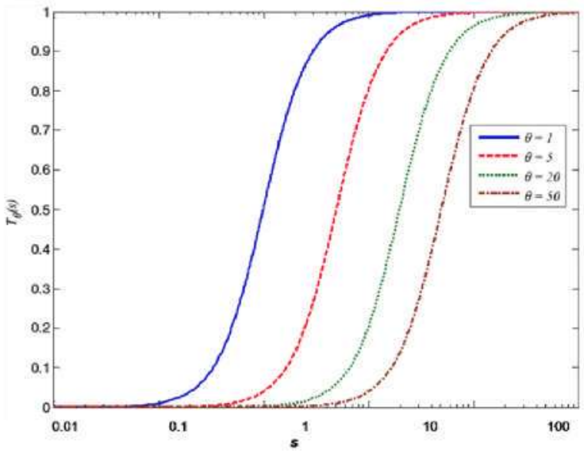

Fig. 2. Semi-logarithmic plot with different thresholds ($\theta = 1, 5, 20, 50$) and with $n = 2$.

Let s be the intensity of a stimulus associated with a particular task; s can be a number of encounters, a chemical concentration, or any quantitative cue sensed by individuals. A response threshold θ, expressed in units of stimulus intensity, is an internal variable that determines the tendency of an individual to respond to the stimulus s and perform the associated task. More precisely, θ is such that the probability of response is low for $s < \theta$ and high for $s > \theta$. This mathematical model that satisfies this requirement is given by:

$$T_{\theta ij}(s_j) = \frac{s_j^n}{s_j^n + \theta_{ij}^n}(n > 1) \tag{1}$$

where $n > 1$ determines the steepness of the threshold. Fig. 2 show several such response curves with $n = 2$, for different values of θ. More clearly: for $s < \theta$, the probability of engaging task performance is close to 0, and for $s > \theta$, this probability is close to 1. Then, the probability that an individual will perform a task depends on s.

The underlying idea is very simple, when a stimulus exceeds the threshold of response of an individual, that individual is likely to respond to stimuli, and engage in the task because the level of the stimulus associated with that task exceeds its threshold. The intensity of a stimulus decreases as the individual performs the task; therefore, individuals with high thresholds are unlikely to perform the task when other individuals, with lower thresholds, maintain the stimulus intensity below their thresholds. However, when individuals with low thresholds do not perform the task, individuals that have high thresholds may engage in the task performance because the stimulus intensity exceeds their thresholds.

The tasks can be constant or can be a time-dependent variable. Stimuli associated with each task can vary considerably from one task to another depending on the nature of tasks, task demand and by the number of robots that are executing the task. Each task is associated with the demand expressed in the form of a stimulus, as when a robot performs a task it tends to reduce the intensity of the associated stimulus, and as a result, modifies the intensity of the stimuli for tasks that are not running. Each robot $\{r\}$ has a set response threshold $\theta_r = \{\theta_1, \theta_2, \ldots, \theta_L\}$. Each threshold $\theta_{r,l}$ corresponds to a task type $l_j = \{l_1, l_2, \ldots, l_J\}$ that the robot is capable of. The initial values of the threshold are randomized to ensure that their roles are not predetermined; the performance of a given task induces a decrease in threshold of the robots:

$$\theta_{r,l}^{new} = \theta_{r,l}^{old} - \sigma. \tag{2}$$

And conversely, the lack of performance of a given task induces:

$$\theta_{r,l}^{new} = \theta_{r,l}^{old} + \sigma \tag{3}$$

where $\sigma > 0$.

4. Learning automata methods

4.1. A brief introduction

Learning automata have made a significant impact and have attracted considerable interest in the last few years [14]. The first research on learning automata models were developed in Mathematical Psychology, that describe the use of stochastic automata with updating of action probabilities which result in a reduction in the number of states in comparison with deterministic automata. They can be applied to a broad range of modeling and control problems, control of manufacturing plants, pattern recognition, and path planning for manipulators, among others. An important point to note is that the decisions must be made with very little knowledge concerning the environment, to guarantee robust behavior without the complete knowledge of the system. In a purely mathematical context, the goal of a learning system is the optimization of a function not known explicitly [15].

Learning is defined as any permanent change in behavior as a result of past experience, and an automata is a machine or control mechanism designed to automatically follow a predetermined sequence of operations or respond to encoded instructions [16]. The objective of stochastic learning automata is to determine how the choice of the action at any stage should be guided by past actions and responses, so when a specific action is performed the environment provides a random response which is either favorable or unfavorable [17].

4.2. Basic definitions

A learning automaton is a sextuple $\langle x, Q, u, \vec{P}(t), G, \mathcal{R}\rangle$, where x is the finite set of inputs, $Q = \{q_1, q_2, \ldots, q_m\}$ is a finite set of internal states, u is the set of outputs, $\vec{P}(t) = \{p_1(t), p_2(t), \ldots, p_m(t)\}$ is the state probability vector at time instant t, $G : Q \rightarrow u$ is the output function (normally considered as deterministic and one-to-one), and $\mathcal{R}$ is an algorithm called the reinforcement scheme, which generates $\vec{P}(t + 1)$ from $\vec{P}(t)$ and the particular input at a discrete instant t.

The automaton operates in a random environment and chooses its current state according to the input received from the environment. The new state probabilities distribution $\vec{P}(t + 1)$ reflects the information obtained from the environment. The random environment has a set of inputs u and its set of outputs is frequently binary {0, 1}, with '0' corresponding to the reward response and '1' to the penalty response. If the input to the environment is u_i the environment produces a penalty response with probability c_i.

Fig. 3 shows the feedback configuration of a learning automaton operating in a random environment. At each instant t the environment evaluates the action of the automaton by either a penalty '1' or reward '0'. The performance of the automaton's behaviors is the average penalty

$$I(t) = \frac{1}{m}\sum_{i=1}^{m} p_i(t)c_i \tag{4}$$

which must be minimized. In order to minimize the expectation of penalty (1), the reinforcement scheme modifies the state probability vector $\vec{P}$. The basic idea is to increase p_i if state q_i generates a reward and to decrease p_i when the same state has

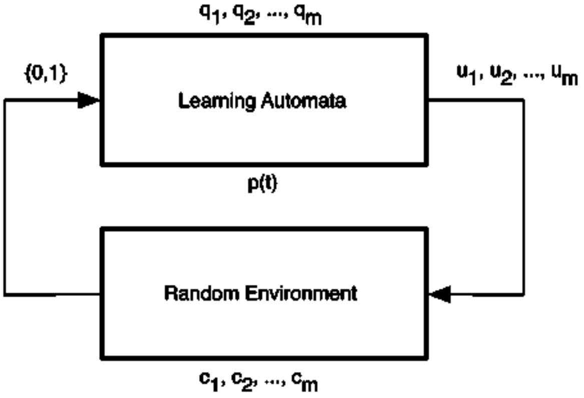

Fig. 3. Interaction of learning automaton with random environment.

produced a penalty. A great number of reinforcement schemes for minimizing the expected value of the penalty have been studied and compared. One of the most serious difficulties that arise in learning automata is the dichotomy between learning speed and accuracy. If the speed of convergence is increased in any particular reinforcement scheme, this action is almost invariably accompanied by an increase of convergence to the undesired state [18,19].

4.3. *Stochastic reinforcement algorithms in learning automata theory*

In the technical literature a widely used stochastic reinforcement algorithm is L_{R-I}, which stands for the Linear Reward-Inaction algorithm.

Let us suppose that the action chosen by the automaton at instant t is ϕ_i, then for the L_{R-I} the updating of the action probabilities is as follows:

$$p_i(t+1) = p_i(t) + \lambda\beta(t)\,[1 - p_i(t)] \tag{5}$$

$$p_j(t+1) = p_j(t) - \lambda\beta(t)p_j(t) \quad \forall j \neq i,\ 1 \leq j \leq N \tag{6}$$

where $0 < \lambda < 1$ is the learning rate and $\beta(t)$ is the environment's response: $\beta = 1$ (favorable response or reward) or $\beta = 0$ (unfavorable response or penalty in which case the algorithm do not change the probability, i.e. inaction).

Let's suppose that there are K different specialized tasks, then we designate by $p_{ij}(t)$, the probability at instant t that robot r_i selects task l_j these probabilities hold:

$$0 \leq p_{ij}(t) \leq 1; \quad \sum_{i=1}^{N} p_{ij}(t) = 1;$$
$$i = 1, 2, \ldots, N \text{ robots; } j = 1, 2, \ldots, K \text{ tasks.} \tag{7}$$

Initially, without the previous robot's experience these probabilities are initialized at the "indifference" position as follows:

$$p_{ij}(0) = \frac{1}{K} \quad \text{for } i = 1, 2, \ldots, N$$
$$\text{robots and } j = 1, 2, \ldots, K \text{ tasks.} \tag{8}$$

Afterwards it starts the learning process in which each robot updates its selection probabilities according to the following conventional updating rule:

$$p_{ij}(t+1) = p_{ij}(t) + \lambda\beta(t)\left[1 - p_{ij}(t)\right] \tag{9}$$

where $0 < \lambda < 1$ is the learning rate with a fixed value of 0.2; $\beta(t)$ is the usual reward signal generated by the environment of the learning automata with the following interpretation: $\beta(t) = 1$; reward if and only if for the corresponding task l_j at instant t it holds that $\#R_j(t) \leq \#L_j(t)$, i.e. the number of robots performing task l_j is lower than the number of tasks l_j to be executed; $\beta(t) = 0$; penalty if and only if $\#R_j(t) > \#L_j(t)$; i.e. when the number of robots performing task l_j is greater than the number of tasks l_j or whenever there are no pending tasks to be executed the automata receives a penalty signal. In other words: at each instant t the environment evaluates the action of the automata; when the response generated by the environment is 1 it means that the action is "favorable" and if the response value is 0 it corresponds to "unfavorable" as follows:

$$\beta_{L_j}(t) = \frac{\#R_j}{\#L_j} = \begin{cases} \text{If} \leq 1 \text{ then reward } \beta = 1 \\ \text{If} > 1 \text{ then penalty } \beta = 0. \end{cases} \tag{10}$$

5. Experimental results

We have conducted several experiments to evaluate the system performance index by applying response threshold models as well as Learning Automata-based probabilistic algorithms to solve the optimal distribution of the tasks among the N robots, so that all of them are executed by means of the minimum number of robots. The ideal objective is that the performance index or learning curve corresponding to the load $l_j(t)$ of each task tend asymptotically to zero for all curves in the minimum time and using the minimal possible number of robots for task execution.

In the simulations we have considered some variants such as: the multi-robot system size, different loads $l_j(t)$ for each type of task, two different ways to carry out the task selection, the additive noise generation to simulate the robot's error and the dynamic generation of tasks $l_j(t)$ over time. According to the results obtained with Eqs. (1) and (9) we have employed for response threshold models and for the learning automata-based probabilistic algorithms two different mechanisms for the selection of tasks:

1. Maximum principle: at each instant t choose the task that has the highest probability for all $T_{\theta ij}(s_j)$ and $p_{ij}(t)$.
2. The strictly random method: using the probabilities $T_{\theta ij}(s_j)$ and $p_{ij}(t)$ in the strict sense of the word, it generates a random number with uniform distribution $(0 - 1)$ and it selects the appropriate task for the value obtained by the method of inversion of discrete probability distributions.

5.1. *Evaluation of the performance index: by noise or error estimation*

To evaluate the evolution of the performance index we have introduced additive noise, perturbing the number of pending loads to simulate the robot's error in estimating the real number of pending tasks. The noise generated is modeled using a normal distribution ("white noise") as follows:

$$Noise = R + R * S = R(1 + S) \tag{11}$$

where *Noise* is the noise generated by the number of pending loads $l_i(t)$, which is proportional to the amplitude of the noise R without perturbing, and S is a Gaussian distribution with a mean of '0' and a typical deviation '0.005': $N(0, 0.005)$.

Figs. 4 and 7 show the evolution of the system performance index obtained for self-selection of heterogeneous specialized tasks through response threshold models as well as learning automata-based probabilistic algorithms, using both mechanisms: maximum principle and the strictly random method, with a team of robots formed by 20–30 heterogeneous robots and 4 types of heterogeneous specialized tasks with different loads. Each experiment has been run 10 times and the results shown are the mean of all of them.

Fig. 4 shows the performance index through threshold response models for the two task selection mechanisms mentioned above

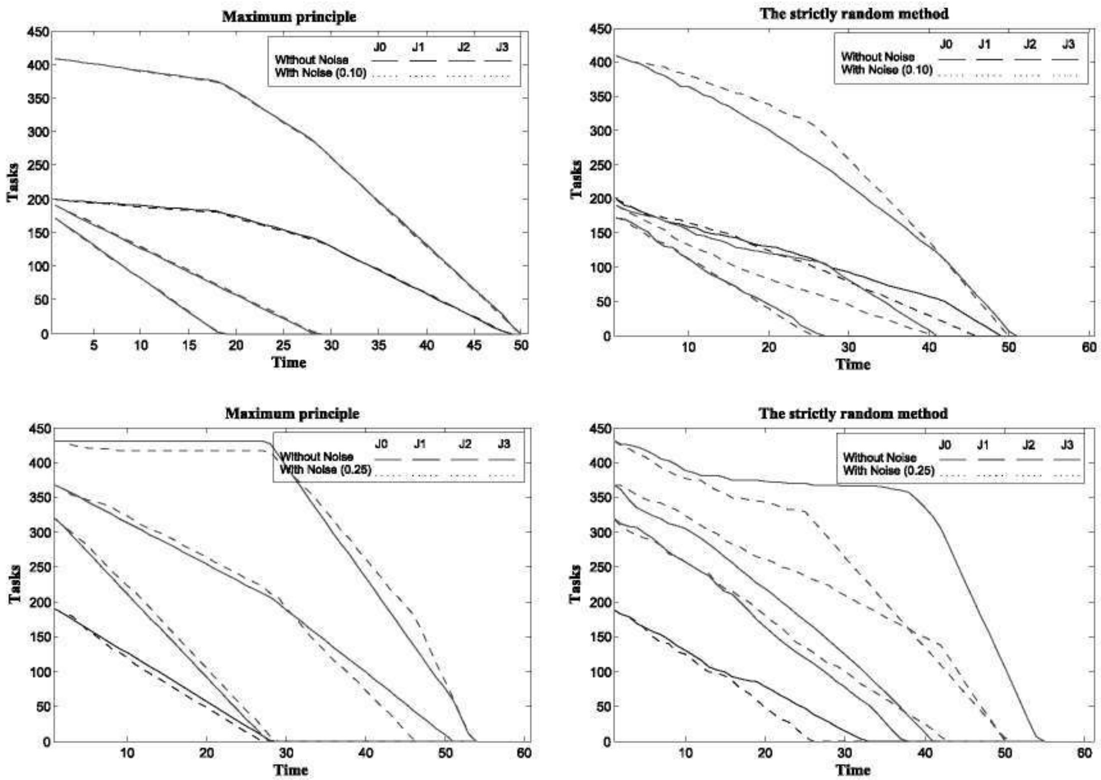

Fig. 4. Learning curves with the evolution of the system performance index for self-selection of tasks using Response Threshold Models for different values of noise.

and for different values of noise, it can be noted that in all cases the generation of additive noise does not affect the performance of the approach; on the contrary, in most cases better results are obtained with the generation of noise.

Similarly, Fig. 7 shows the performance index using Learning Automata-based probabilistic algorithms for both mechanisms and for different values of noise, it can be observed that learning curves corresponding to the load $l_j(t)$ of each task tend asymptotically to zero for both methods. However, when additive noise in this approach is introduced it can be clearly seen that in some cases more time is required for the execution of tasks.

According to previous results it can be observed that system performance with the learning automata approach is more affected with the introduction of noise versus the results shown in the response threshold models approach.

5.2. Evaluation of the performance index: by dynamic tasks generation

In the previous experiments, the number of loads for each type of task is determined from the beginning of the simulation and there is no change until the end of the execution. To evaluate the performance of the algorithm we have generated dynamic tasks. This idea was rescued from classical models of queue simulation, so we have used a Poisson distribution to determine the probability of generating a number of tasks through time:

$$f(k;\lambda) = \frac{e^{\lambda}\lambda^{k}}{k!}. \qquad (12)$$

Specifically we will have a different distribution for $k = 1$ to 100. Each λ is a positive real number representing the number of tasks expected to be generated during a time interval. For that expected number of tasks generated to be decreasing, and therefore the system is stable, we have parameterized this constant λ as follows:

$$\lambda(t) = \sigma - \alpha * t \qquad (13)$$

where σ is the initial value (for example, 10 or 20) and α is a factor of “reduction tasks” that initially we have defined as 1. Finally, t corresponds to the time of execution at each instant.

Figs. 5 and 6 show the evolution of the system performance index with dynamic tasks generation through time using the Poisson distribution. Experiments have been performed 10 times and the results shown are the mean of all of them. We have also additive noise generated in the loads with the maximum principle and the strictly random method. In the results it can be observed for dynamic tasks generation, the task number generated is decreasing over time. All learning curves tend to zero in both mechanisms and do not affect the performance for any approaches.

6. Conclusions and further research work

In this paper we have proposed an experimental scenario for the self-coordination problem of multi-robot systems in the heterogeneous multi-task distribution to be executed by a team of heterogeneous mobile robots and we have applied two different approaches for this problem by applying Response Threshold Models as well as Learning Automata-based probabilistic algorithms. To carry out the selection of tasks in both approaches we used two mechanisms: maximum principle and the strictly random method and, in most experiments the best results are obtained with maximum principle instead of the strictly random method. We have generated additive noise to evaluate the robustness of both approaches, perturbing the number of the pending load, to simulate the robot's error in estimating the real number of pending tasks. According to the results obtained the noise generated does not affect the performance of the response threshold models approach since the best result are obtained by generating noise in the pending loads. However, by applying learning automata-based probabilistic algorithms in some cases more time is required for the execution of tasks. We have also studied the performance index with dynamic generation of loads through time and the results

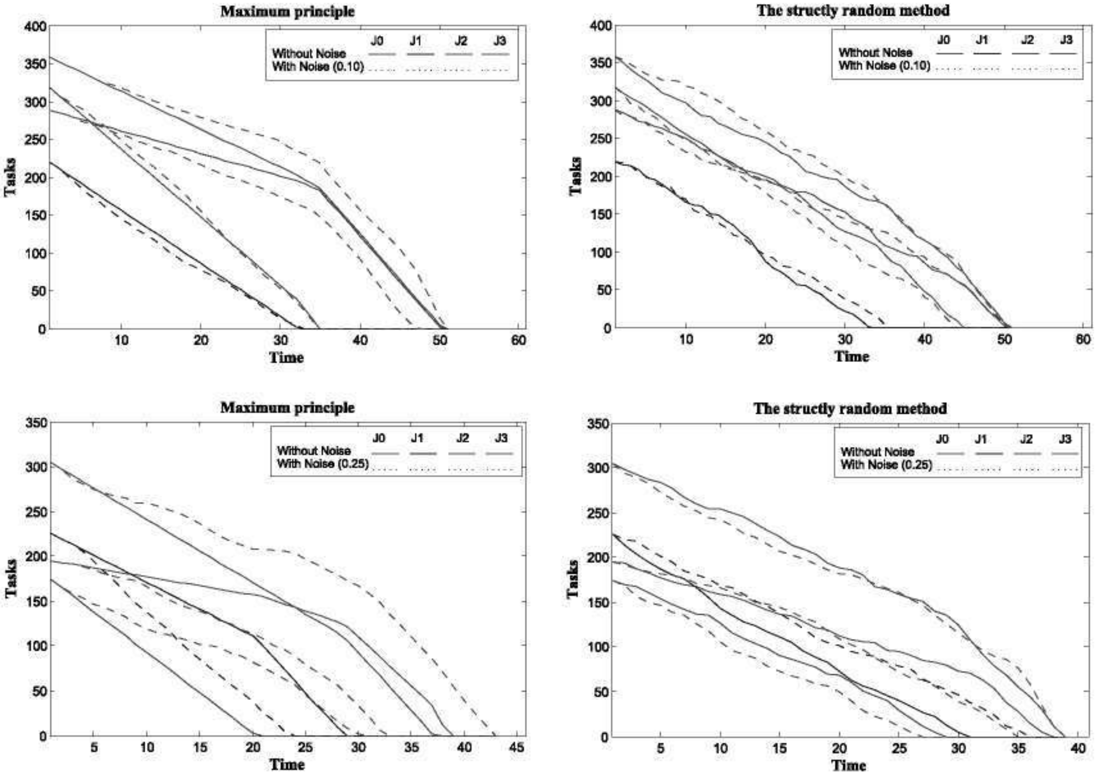

Fig. 5. Learning curves with the evolution of the system performance index for self-selection of tasks using Learning Automata-based probabilistic algorithms for different values of noise.

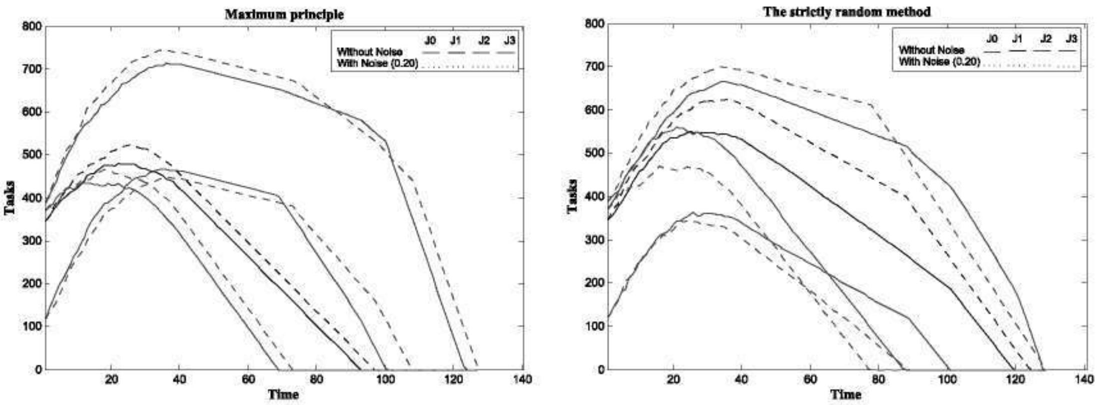

Fig. 6. Dynamic tasks generation: learning curves with the evolution of the system performance index for self-selection of tasks using Response Threshold Models.

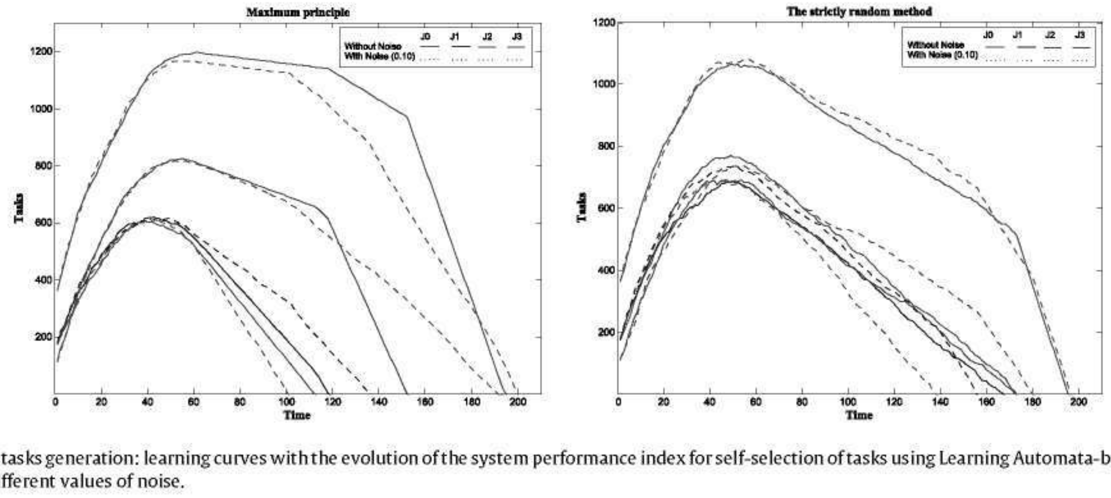

Fig. 7. Dynamic tasks generation: learning curves with the evolution of the system performance index for self-selection of tasks using Learning Automata-based probabilistic algorithms for different values of noise.

confirm that the robots are capable of selecting in an autonomous and individual manner the existing tasks without the intervention of any global and central task scheduler. We have shown that both approaches can be efficiently applied to solve this self-coordination problem in multi-robot systems obtaining truly decentralized solutions.

References

[1] L. Parker, Multiple mobile robot systems, in: S. Bruno, K. Oussama (Eds.), Handbook of Robotics, Springer, 2008.

[2] B. Gerkey, M. Mataric, Cooperative multi-robot box-pushing, in: IEEE International Conference on Robotics and Automation, 2003, pp. 3862–3868.

[3] B. Gerkey, M. Mataric, A formal analysis and taxonomy of task allocation in multi-robot systems, International Journal of Robotics Research (2004) 939–954.

[4] A. Farinelli, L. Locchi, D. Nardi, Multirobot systems: a classification focused on coordination, IEEE Transactions on Systems, Man and Cybernetics (2004) 2015–2028.

[5] G. Oster, E. Wilson, Caste and ecology in the social insects, in: Monographs in Population Biology, Princeton Univ. Press, 1978.

[6] G. Robinson, Regulation of division of labor in insect societies, Annual Review Entomology (1992) 637–665.

[7] Y. Quiñonez, J. De Lope, D. Maravall, Bio-inspired decentralized self-coordination algorithms for multi-heterogeneous specialized tasks distribution in multi-robot systems, Foundations on Natural and Artificial Computation (2011) 30–39.

[8] J. De Lope, D. Maravall, Y. Quiñonez, Decentralized multi-tasks distribution in heterogeneous robot teams by means of ant colony optimization and learning automata, in: International Conference on Hybrid Artificial Intelligence Systems, 2012, pp. 103–114.

[9] Y. Quiñonez, J. De Lope, D. Maravall, Stochastic learning automata for self-coordination in heterogeneous multi-tasks selection in multi-robot systems, in: Mexican International Conference on Artificial Intelligence, 2011, pp. 443–453.

[10] B. Gerkey, M. Mataric, Multi-robot task allocation: analyzing the complexity and optimality of key architectures, in: IEEE International Conference on Robotics and Automation, 2003, pp. 3862–3868.

[11] E. Bonabeau, G. Theraulaz, J. Deneuborurg, Quantitative study of the fixed threshold model for the regulation of division of labour in insects societies, in: Proceedings Biological Science, 1996, pp. 1565–1569.

[12] T. Seeley, S. Camazine, J. Sneyd, Collective decision-making in honey bees: how colonies choose among nectar sources. behavioral ecology and sociobiology, Behavioral Ecology and Sociobiology (1991) 277–290.

[13] E. Bonabeau, G. Theraulaz, J. Deneubourg, Fixed response thresholds and the regulation of division of labor in insect societies, Bulletin of Mathematical Biology (1998) 753–807.

[14] K. Narendra, M. Thathachar, Learning Automata: An Introduction, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[15] K. Narendra, M. Thathachar, Learning automata: a survey, IEEE Transactions on Systems, Man and Cybernetics (1974) 323–334.

[16] M. Obaidat, G. Papadimitriou, A. Pomportsis, Guest editorial learning automata: theory, paradigms, and applications, IEEE Transactions on Systems, Man and Cybernetics (2002) 706–709.

[17] D. Maravall, J. De Lope, Fusion of learning automata theory and granular inference systems: anlagis, Applications to Pattern Recognition and Machine Learning (2011) 1237–1242.

[18] K. Narendra, E. Wright, L. Mason, Applications of learning automata to telephone traffic routing and control, IEEE Transactions on Systems, Man and Cybernetics (1977) 785–792.

[19] K. Narendra, R. Viswanathan, A two-level system of stochastic automata for periodic random environments, IEEE Transactions on Systems, Man and Cybernetics (1972) 285–289.