Multi-Robot Task Allocation Strategies Using Auction-Like Mechanisms

José Guerrero, Gabriel Oliver
Dept. de Matemàtiques i Informàtica, Universitat de les Illes Balears (UIB), Cra. de Valldemossa, km 7.5, 07122, Palma de Mallorca (SPAIN)
$\{$*jose.guerrero,goliver*$\}$*@uib.es*

Abstract. Task allocation is a complex and open problem for multi-robot systems. In this paper, we present three strategies for dynamically allocating tasks in a colony of mobile robots. The system is partially inspired on auction and thresholds-based methods and tries to determine the optimum number of robots that are needed to solve specific tasks, or sub-goals, in order to minimize the execution time of a mission. The method has been extensively tested for a modification of the well-known foraging task. Initial situations with different combinations of workload and work capacity have been used to test the proposed solution on robot teams with different homogeneities. Experimental results are presented to show the benefits of the proposed methods.

1 Introduction

Multi-robot systems can provide several advantages over single-robot systems: robustness, flexibility and efficiency among others. To benefit from these potential aspects the robots must cooperate to carry out a common mission. It is well known that several problems have to be solved to archive this aim. Some tasks can be carried out by a single robot but if two or more robots can cooperate, the task can be archive faster. In this situation, the following question has to be answered: How many robots and which of them do we need to optimize the execution time of this task?. As it has been demonstrated in different studies, if the number of robots is increased with wrong criteria, the execution time can increase, among other causes, due to the interference effect. Interference is the result of competition for the shared resources, especially the physical space and the robots themselves. In [10] it is shown how the interference can affect the execution time of the foraging tasks and how increasing the number of robots also causes the time to finish the mission to increase. Besides, if a certain task captures the attention of an excessive number of robots, other tasks can be forsaken. Therefore, our challenge is to design a distributed system to coordinate a heterogeneous robot team working in an unknown and dynamic environment. We mainly focus on the task allocation, that is, selecting the best robot or robots to perform a specific task.

In this paper, we propose and evaluate a first version of a decentralized method of task allocation for groups of heterogeneous robots executing foraging-like tasks. Our method is partially inspired in both the auction-based works [7], [6] and the threshold models of the swarm intelligence approach [4]. Thus, strategy presented here are hybrid solutions between those approaches. Special attention will be focused on deciding the optimal number of robots to execute a specific task. This problem is closely connected with the diversity level of the team of robots, as will be shown later. To measure the heterogeneity of the robot collectivity we use, among others, the social entropy proposed by T. Balch [3]. We will also study the relation between this metric and some of our architecture parameters. Also, three different variations or strategies of our method are presented: ”no monitoring and no preemption”, ”monitoring and no preemption” and ”monitoring and preemption”. The differences among these three strategies lie in the capacity of monitoring the task progress and in the group preemption ability. The preemption ability is the capacity for change the task assigned to a robot.

In the system presented here, when a robot detects a non-assigned task tries to organize a working group. This group consists of all the robots that will cooperate to execute the specific task. One of the robots of this group will be the leader. The leader will decide how many robots and which of them will be needed to execute the task. To select these robots the leader starts an auction process, similar to [7]. The auction mechanism does not allow deciding how many robots will form the group. Thus, to take this decision, the leader together with the auction mechanism has to calculate the ratio between the work to do and the work capacity of the robots in the group. One of the objectives of the leader is to create a group as reduced as possible with the maximum work capacity. It is important to mention that the number of robots of the group can vary during the task execution. As it will be shown, during the mission, the leader can accept more robots as members of the group while other robots can abandon it too. In non preemption executions a robot can only abandon its task to became leader and create its own group. On the other hand, the preemption approach implements an inter-group or inter-leader negotiation mechanism. During this negotiation the leader can transfer one or more of its robots to another group. The experiments shown here are inspired on the so-called foraging or transportation tasks. In the present case, each robot has a limited load capacity (work capacity). If a robot can’t load the full object, it loads a portion of the weight, goes to the delivery point and comes back for more.

The rest of this paper is organized as follows: section 2 presents some relevant work in the field of multi-robot task allocation; section 3 describes our methods and their implementation; section 4 shows the experiments carried out to validate the different approaches; section 5 explains some conclusions and future work is stated.

2 Related work

The computer engineering community has done a lot of research to solve the task allocation problem. In recent years, some studies on multi-robot systems have used some similar ideas to solve the problem of how robot teams can distribute their individual work capacity to efficiently achieve a common task. This section shortly relates some of those researches that have inspired us. These approaches are the marked-based systems, inspired in the economy markets, and the threshold-based systems, inspired in the swarm intelligence paradigm.

The marked-based approaches are based in a negotiation between the robots to cooperate [7], [6]. The robots act as self-interest agents and try to maximize their own benefits and reduce the costs. When a robot executes a task it can obtain a benefit but it also has to pay an execution cost. The cost may be, for example, the distance between the robot and the target point, and the profit could be the reduction in the global work to be executed by the robots. If each robot, as self-interest agent, reduces its own cost and maximizes its benefit, the profit for the whole system will be maximized. The robots can subcontract part of the task execution to other robots. The subcontracted robots can carry out the task with a lower cost or higher benefit. Finally, the profit is distributed among all the robots which cooperated during the execution. Thus, the overall system profit is increased thanks to the robot cooperation.

Dias and Stenz [6] have proposed cooperation mechanisms based on the marked-based metaphor. In the same line, Gerkey and Mataric are working on auction-based mechanisms [7]. In these systems the robots bid for the tasks. The robot with the highest bid wins the auction process and gets the task. The bids are adjusted to the robots' interest (capacity) to carry out the goal. Thus, when a robot finds a task, it can start an auction process and subcontract its execution to other robots.

Other works can be found in the literature inspired in some biology metaphors. One of the best known are the so-called swarm intelligence paradigm systems, which are inspired in the collective behaviour of insect colonies like ants and wasps. To implement these systems some authors make use of the response thresholds systems [4], [9]. In these systems, each robot has a stimuli associated with each task to execute. The stimuli value depends on some intrinsic parameters of the robots and the tasks. When the level of the stimuli exceeds a threshold, the robot starts its execution. A pure threshold-based system dosen't requires communication mechanisms between robots and therefore, there are no negotiation protocols. Recently, some authors include a very simple communication mechanism to improve the system performance [1] . In this system the robots can exchange the information about the task location, but any negotiation protocol is implemented.

The pure threshold-based systems don't require any kind of communication mechanisms. Therefore, they are not limited by the bandwidth communication restrictions and they are very scalable. Nonetheless, a disadvantage of these systems is the absence of knowledge about the other robots. Thus, a robot can decide by itself to execute a task when other option could be better. Moreover, using a pure threshold-based system the execution of plans shared by several robots would be very difficult. The marked-based systems do not have this limitation but they require communication mechanisms. These communication requirements limit the system scalability. It has to be pointed out that both for threshold-methods and marked-based methods, a function has to be designed to quantify the cost and the profit values of a task. This function is task dependent, thus, when the robot team executes a different task, a new functions has to be used. In [4] a comparison between biology-based and marked-based task allocation algorithms is presented. The experiments shown in that paper are performed with industrial manipulators and they conclude that the performance of both, biology-based and marked-based systems is very similar.

Our approach is partially inspired in the auction mechanisms and, consequently, the best robot for a specific task can be chosen. However, while previous works cannot determine the optimal number of robots to execute a task, our method allows deciding this number as a function of both the amount of work required to complete the task and the work capacity of the robots involved in the auction process. Moreover, during the leader to leader negotiation the robots bid using a threshold-based like equation. Thus, this paper try to shows the initial works to create an hybrid system which will combine the advantages of both, threshold and market-based systems.

3 Working group leading, formation and updating

The coordination mechanisms description, including the groups' formation, the leading, the membership policy, the task assignment and the preemption strategies is described in the following paragraphs.

3.1 Single robot to leader negotiation

This section describes the negotiation that is carried out between each single robot and one group's leader. This negotiation level corresponds to a non preemption strategy, and therefore it doesn't allow the exchange of robots (work capacity) between groups.

In an initial stage, each robot is looking for a task. When a robot finds a new task by itself, it will try to lead it. As there can be only be one leader per task, the candidate will first check if this task is assigned to any other robot or not. If there are two or more robots requesting the leadership of the same task, it will be assigned to the 'best' robot. To determine their suitability, we evaluate each candidate using a measure that depends on the specific task to do. During our experiments this measure is the robots' load capacity.

When a robot is promoted to leader of a task, it evaluates the work needed to carry it out. Then, it will create, if necessary, the work group; that is, the set of robots that will cooperate to execute this specific task. In that case, the leader must decide which is the optimum group size. In this paper, we propose the leader to decide, as it will be exposed in the auction mechanism, using the following equation:

$$TH_g = \frac{taskWorkLoad}{\sum_{1\leq i\leq N} workCapacity_i} < TH \tag{1}$$

Where $taskWorkLoad$ is the amount of work required to finish the assigned task that is calculated by the leader. N is the number of robots of the group and $workCapacity_i$ is the individual work capacity of the ith robot. Finally, TH is the group threshold; this value is a parameter that will be used to compare the efficiency of the group formation policy. Using equation (1) the leader fixes the maximum ratio between work to do and available work capacity and, therefore, it fixes the maximum number of robots that will be part of the group. This concept is similar to the response threshold of the swarm intelligence algorithms, where the stimulus for each robot is a combination of its load capacity and the taskWorkLoad. In the present case, the decision process is also centralized on the leader.

The leader monitors the task execution, thus, in any time the leader knows the current value of the taskWorkLoad parameter of the equation (1). Obviously, the value must decrease as the execution progress. The practical problem is how can the leader monitor this progress?, and is it necessary to monitor the progress?. Section 4 shows the impact of the monitor process over the execution time.

To select the robots that will be part of its group the leader uses an auction process. Unlike other auction-based methods, we use the inequality (1) to select the robots. This process is as follows:

- *Auction beginning*: the leader sends a massage to inform that it has found a new task and needs to form a working group.
- *Bids*: each robot without any assigned task (robots that are not part of any group) sends to the leader its work capacity.

- *Auction finishing*: after a fixed time has passed, the leader closes the auction and doesn't accept more bids.
- *Selecting the robots*: the leader selects the robots with the highest work capacity provided the inequality (1) is verified.
- *Agreement and refuse negotiation*: The leader sends an agreement massage to the selected robots and, if the offer is accepted, they start to work for the group. The robots that do not accept to be part of the group reply with a refuse message and the leader subtracts their work capacity from the group capacity. A robot can refuse an offer if it joined another group or it became a leader task during the auction process.

If after this process the equation (1) is not fulfilled, the leader starts a new auction round. This new auction round will include a leader to leader negotiation as will be explained in section 3.2. When the task is finished, the working group is dissolved immediately.

Using only this kind of negotiation, the size of the group can vary during the execution of the task by means of two mechanisms. Group size can be reduced by a robot segregation process. If a non-leader robot, which is part of a group, finds a new task, it will try to leave its present group and become a leader of the new task. When this situation occurs, the segregated robot communicates this circumstance to its old leader which will recalculate equation (1) for the new situation of its group and, eventually, can initiate a new auction process. On the other hand, the size of the group can increase thanks to a robot aggregation process. This process takes place when after a certain time has passed a robot does not have any task to execute. Then, it sends a message asking for a task. If a leader receives one of these messages, it accepts the robot in its group if the group size is less than a fixed number. Using this mechanism, the inactive robots are avoided. A new mechanism to modify the group size will be included thanks to the leader to leader negotiation.

3.2 Leader to leader negotiation: Preemption

The leader to leader negotiation allows the exchange of robots between groups. If after the single robot to leader negotiation the equation (1) is not fulfilled, the group's leader tries to contract the robots which are working in another task. To select these robots, each leader bids for the task using both its robots load capacity, as in the previous negotiation, and its working group energy. Using the operating systems vocabulary, during the experiments this kind of algorithm will be called strategies with preemption.

The working group energy is a measure to indicate the group's tendency to send its robots to another group. A group with a high energy value is a potential sender of robots to other tasks and, on the other hand, a low energy value indicates that this group is a receptor of new robots. The energy is calculated using the following equation:

$$E_g = \frac{GroupSize}{TH_g} \tag{2}$$

Where $GroupSize$ is the number of group robots and TH_g is obtained from equation (1). A high value of TH_g indicates that the group has a low value of work capacity compared with the task to carry out. In this case the group needs more robots and, therefore, the energy is low. Moreover, the leader has to try to create a group with the minimum number of robots to reduce the interference effect between robots. Thus, a group with a lot of robots needs to reduce this number, and this means a higher energy value.

The energy concept is similar to the stimulus intensity value used by some threshold based algorithms [4]. Using stimulus the robots have to belong to a specific cast or group with its particular characteristics. Nevertheless, using the energy concept, each robot has its own load capacity and it is not necessary a preliminary division in groups. Finally, the main difference between the threshold and energy strategies lies in the fact that the energy value will be used as the bid value in an action process, and therefore, the decisions will be taken knowing its impact over the others groups. In a pure threshold based system this decision is reactive and it is taken in independence from the others groups.

To select robots from other groups the leader uses another auction process. This process is as follow:

- *Auction beginning*: if the equation (1) is not fulfilled after the previous auction process, the leader sends a 'leader to leader' negotiation request.

- *Bids*: each leader sends the value of its energy and the information about the robots of the group; this is, the identification of each robot and its load capacity. A robot only can abandon a group after a fixed time passed since its incorporation. During the experiments this time is equal to 85 units. As in the previous auction process, if a robot without any assigned task received this message, it sends its load capacity.

- *Close Auction*: after a fixed period of time the leader closes the action and doesn't accept more bids.

- *Selecting robots without assigned task*: first of all the leader selects the robots without any assigned task and with the higher load capacity.

- *Selecting robots associated to a previous group*: if the equation (1) is not fulfilled, the leader selects the robots sent by the others leaders. The goal of this selection is create a more stable system. By the way, a system where the groups have low energy will be more stable than a system made up of groups with high energy. A more appropriate definition of system stability is now under study.

The leader, before starting the auction process, pre-selects the robots that will participate in the auction process. A robot from another group is pre-selected if reassigning it to the new group produces a reduction of the maximum value of the global energy; that is, if the following condition is verified:

$$MAX(E_{g1}, E_{g2}) > MAX(E'_{g1}, E'_{g2}) \tag{3}$$

Where $MAX(v1, v2)$ returns the maximum value of v1 and v2. E_{g1} is the energy of the acceptant group and E_{g2} is the energy of the requesting group before the transaction. E'_{g1} and E'_{g2} are the predicted energies after the transaction. Therefore, the left side of the equation (3) is the current state of the system and the right side represents the situation after the robot selection. Finally, O is the perceptual overhead produced by the group change. This factor avoids the robot selection when the benefit obtained by the group is very low. During the experiments the overhead is equal to 50% and, therefore, the O value is equal to 1,5. The selected robots bid during the auction process not only using its work capacity, like in the single robot to leader negotiation, but using the following value:

$$B = workCapacity * E'_{g2} \tag{4}$$

Where $WorkCapacity$ is the individual work capacity of the robot and E'_{g2} is the energy of the robot's group if this robot is selected. The robot with a higher bid is selected. The action process finished when the equation (1) is fulfilled or when no more robots validate the equation (3). This auction process allows selecting the best robot between all the pre-selected candidates. If after this process the equation (1) is not fulfilled another auction process is started.

4 Experiments and validation

This section explains the experiments carried out to validate our approach. It is also shown how the value of the group threshold and the social entropy of the group affect the execution time. We have also evaluated the impact of the task work load monitoring and the effects of the leader to leader negotiation over the team. All the experiments have been done using the RoboCoT environment. The mission to be carried out by the robots is a modification of the classical foraging task.

4.1 Measure of the homogeneity of the collective

It has to be emphasized that each robot can have a different work capacity. Thus, we have focused part of our study on the measure of the performance of the collective while its homogeneity is changed. To characterize the homogeneity of the collective of robots we use two entropy measures. On the one hand, the simple entropy measure based on the Shannon's information entropy:

$$H = \sum_{1 \leq i \leq M} p_i \log_2(\frac{1}{p_i}) \tag{5}$$

where M is the number of classes of robots and p_i is the proportion of robots in the ith class. On the other hand, social entropy as formulated by Balch [3] extends simple entropy to take into account the quantitative differences between groups. In our experiments, these differences are the work capacity of each robot; therefore, two robots belong to the same group if they have the same work capacity.

4.2 Platform and task description

We use as test bed a multi-robot simulator called RoboCoT (Robot Colonies Tool). RoboCoT is a software tool developed and used by the authors at the University of the Balearic Islands that allows testing the performance of individual or colonies of mobile robots in a very flexible, fast and non-expensive way. This simulator implements robots that work according to a control architecture based on behaviours, as they were introduced by Ronald C. Arkin [2]. It has been widely proved that efficient robot navigation capabilities can be obtained at

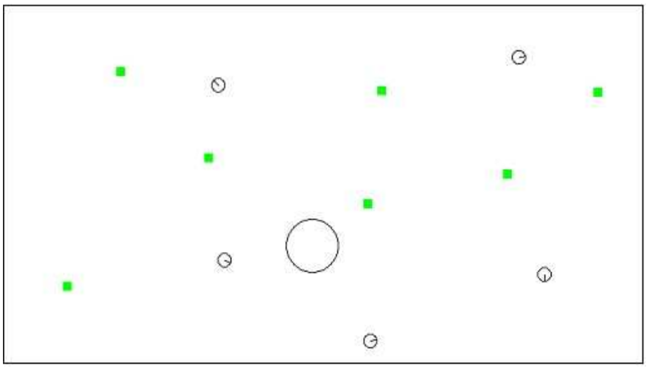

Figure 1: Example of initial situation of the experiments.

Table 1: Robots' load capacities used during the experiments. R1..R5 represent the robots' load capacities. The H represents the simple entropy of the configuration and SH the social entropy.

Configuration	R1	R2	R3	R4	R5	H	HS
1 (Homogeneous)	3	3	3	3	3	0	0
2	1	1	1	1	11	0,72	7,22
3	1	1	3	5	5	1,52	4,49
4	1	2	3	4	5	2,32	2,32

a reasonable low computational cost with such architectures. A detailed discussion about the RoboCoT architecture can be found in [8].

The task to be carried out by the robots is described as follows: some randomly placed robots must locate objects, randomly placed too, and carry them to a common delivery point. Figure 1 shows a typical initial situation, where the squares represent the objects to collect, the delivery point is the big circle in the middle of the image and the robots are the little circles. Each object to gather has a weight and each robot has a load capacity. The robot load capacity is the amount of weight that it can carry. If a robot cannot carry the entire object at once, it takes a part of it, goes to the delivery point and comes back to the object for more bits. Therefore, unlike a classical foraging task, the objects can be transported by the robots in little transportable bits. It is obvious that in this kind of experiments, a robot can always complete any specific task, carrying bit by bit all the objects to the delivery point. However, if several robots cooperate, the task can be done faster. During the experiments, the taskWorkLoad value is the object weight and the workCapacity of a robot is its load capacity.

In all the experiments presented here, we have used seven objects to load and five robots. The objects' weights are: 1, 3, 5, 5, 8, 10 and 15 units. All the robots have the same sensorial, behavioral and communication capabilities, and they only differ in their load capacity. To study the impact of the robots' homogeneity over the execution time we have used four different configurations or combinations for the load capabilities of the robots. Table 4.2 shows these configurations, the load capacities of the robots, and the value of simple (H) and social (SH) entropy of the robot community. For each configuration, we have used as values for the group threshold (TH): 0, 1, 2, 3. In addition, we have executed the experiments without communication between robots. In the case TH=0, equation (1) has not been used, and therefore the number of robots per group is not limited. We have repeated each experiment 56 times using 14 different environments. The environments are different because of the initial placement

Table 2: Mean of the execution time for the experiments using a no monitoring and no preemption strategy. The "No Comm." row represents the experiments executed without communication. C1..C5 represent the robots configurations

-	C1	C2	C3	C4
No Comm.	996,6	1384,3	1002,3	1079,8
TH=0	934,4	1060,6	948,3	947,5
TH=1	919,2	1047,3	964,1	886,5
TH=2	878,7	1043,4	852,7	970,3
TH=3	892,4	1081,5	886,2	893

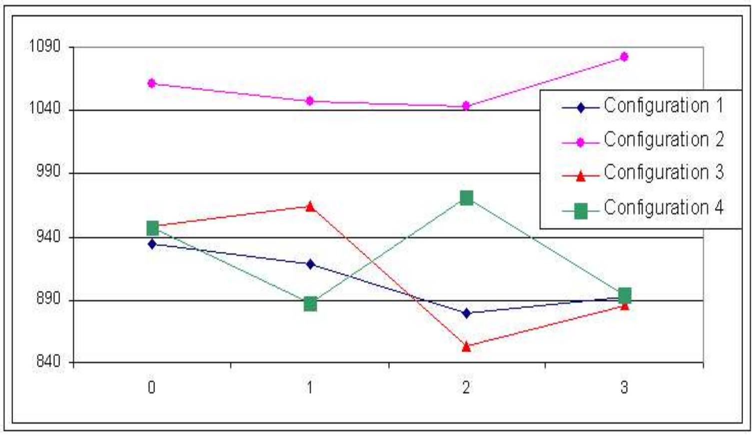

Figure 2: Execution time as a function of robot configuration for a no monitoring and no preemption strategy.

of the robots and the objects. To evaluate the impact of both, work load monitoring and leader to leader negotiation (preemption), three different strategies have been tested: no monitoring and no preemption, monitoring and no preemption and monitoring and preemption.

4.3 No monitoring and no preemption strategy

Using a no monitoring and no preemption strategy the group's leader can not monitor the progress of the task. It only can detect when the task is finished. Therefore, the value of the task work load of equation (1) is fixed during all the execution. Also, the leader to leader negotiation, presented in section 3.2 is not implemented. As it can be seen, this is the simplest strategy presented in this paper.

Table 4.3 and figure 2 show the mean of the execution time of the experiments. As it can be seen in this table, in most cases equation (1) involves the reduction of the execution time compared to the system with null threshold (TH=0). The configuration with the highest social entropy (SH) presents the lowest benefits due to the use of threshold. In this configuration the highest benefit reaches the 1,6% (TH=2), but when TH=3 the time is increased a 2%. On the another hand, the configuration 3 presents the highest benefit (10,1%) when TH=2 and the configuration 4 presents the highest increase (2,4%) when TH=2. As it can be seen, the use of communication produces a considerable reduction of the execution time in all the cases, and this time increases as the social entropy (SH) grows. The figure 2 also shows that the worst case is always the configuration with the highest social entropy value (configuration 2).

Table 3: Mean of the execution time for the experiments using a monitoring and no preemption strategy. C1..C5 represent the robots configurations

-	C1	C2	C3	C4
TH=0	934,4	1060,6	948,3	947,5
TH=1	869,9	1019,4	930,3	863,7
TH=2	905,8	996,8	880,8	950,5
TH=3	904,1	1044,3	922,8	910,1

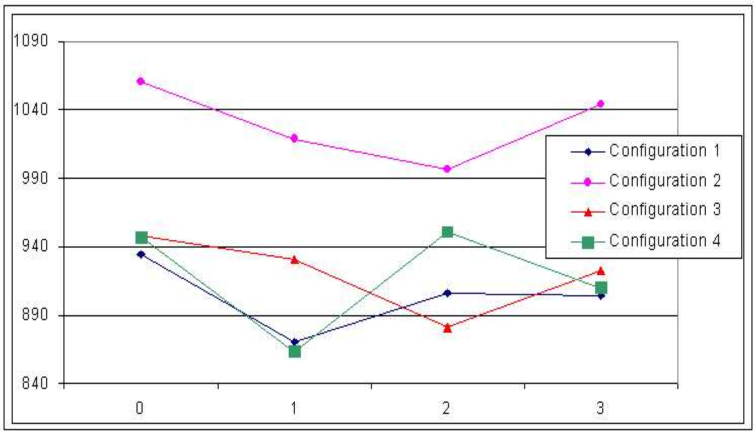

Figure 3: Execution time as a function of robot configuration for a monitoring and no preemption strategy.

4.4 Monitoring and no preemption strategy

Figure 3 and table 4.4 show the mean of the execution time of our monitoring and no preemption strategy experiments. If these results are compared with the no monitoring and no preemption strategy results, it can be seen that for all configurations the worst execution time is reduced. This effect is more evident in the configuration 2. Nonetheless, the better execution time is increased in some cases. For example, for the configuration 3 the better case is 880,8 but during the no monitoring experiments this time was 852,7. Therefore, the monitoring task produces a ”smoothing” effect over the execution time, but, in general, it seems that a little reduction of the execution time is produced.

4.5 Monitoring and no preemption strategy

Monitoring and preemption strategy implements both the leader monitoring capacity and the leader to leader negotiation process. Thus, this is the most sophisticate strategy that we present here. Figure 4 and table 4.5 show the execution time for the experiments and, as it can be seen, the results are similar to those of the previous strategy. The main difference appears during the configuration with the highest simple entropy (configuration 4) where, in general, the execution time is reduced. Also, a reduction of the execution time is produced when the threshold is equal to 2. We expect that the leader to leader negotiation is especially useful if a priority is associated to each task. Thus, if a robot finds a high priority task, it can negotiate the temporary cession of some robots that are executing lower priority tasks using the leader to leader negotiation mechanism.

Table 4: Mean of the execution time for the experiments using a monitoring and preemption strategy. C1..C5 represent the robots configurations

-	C1	C2	C3	C4
TH=0	944,4	1099,8	957,9	955,6
TH=1	930,1	1026,6	969,4	874,8
TH=2	917,6	996,5	891,1	885,8
TH=3	890,9	1066,9	910,2	881,9

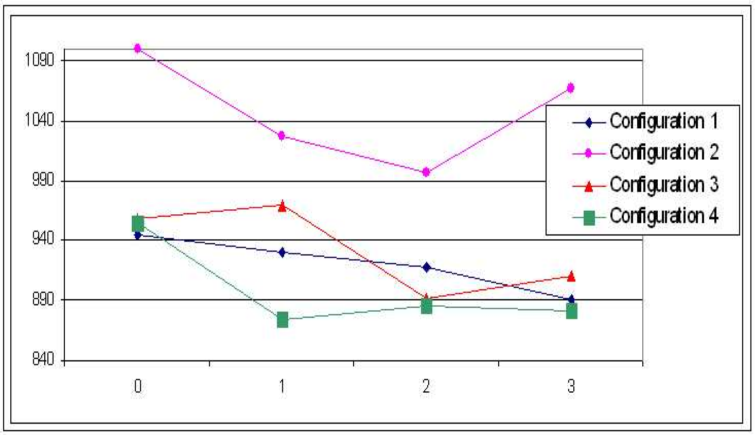

Figure 4: Execution time as a function of robot configuration for a monitoring and preemption strategy.

5 Conclusion and future work

This paper presents a simple and efficient way to solve the task allocation problem and, more specially, to decide how many robots are needed to execute a specific task. Our algorithms adapt the classical auction using some threshold-based systems concepts to find the optimal number of robots. In addition, our method allows both changing the robots assigned to a task as new objectives are found and interchanging robots between working groups. Thus, we have provided a faster and flexible way to regulate the optimal number robots as a function of the kind of the task and the available robots. Three versions of our mechanism have been proposed and tested using several environments. All the cases have been tested using a modification of the classical foraging task to allow a higher cooperation degree between robots. The execution results of this task prove that our mechanism reduces the mission's execution time.

This paper presents a work in progress that has some challenging aspects to add and to improve. For the time being we are focused on a deep analysis of the data available, obtained from a huge set of experiments that should take us to a precise understanding of the relations between heterogeneity and group thresholding as well as robot segregation, aggregation and preemption processes or auction algorithms. We are working on developing a better definition of system stability. As it can bee seen in section 4, the execution time of the mission is not modified by the leader to leader negotiation. The next kind of mission to implement by our system will include tasks with priorities. We expect that in this kind of environment the preemption strategy will be very useful. Thus, if a leader finds a high priority task, it could ask for help to other groups with lower priority tasks and some of their robots could temporarily be transferred to the demanding leader. Another aspect we are working on is the replacement of the leader mechanisms. These strategies should be activated when the leader robot faults and should improve the robustness of the system. Finally, another aspect of the systems that should be improved is the use of a non fixed threshold. Dynamic thresholding techniques in the line of those exposed in [5] will be tested.

Acknowledgments

This work has been partially supported by project CICYT-DPI2001-2311-C03-02 and FEDER fundings.

References

[1] W. Agassounon, A. Martinoli, ”Efficiency and Robustness of Threshold-Based Distributed Allocation Algorithms in Multi-Agent Systems”, First Int. Joint Conf. on Autonomous Agents and Multi-Agents Systems, (2002), 1090–1097.

[2] R.C. Arkin, *”Behaviour-Based Robotics”*, The MIT Press, (1998).

[3] T. Balch, ”Social Entropy: A New Metric for Learning Multi-Robot Teams”, *10th international Florida Artificial Intelligence Research Society Conference*, CA:AAAI Press, (1997).

[4] E. Bonabeau, A. Sobkowski, G. Theraulaz and J. L. Deneubourg, ”Adaptive Task Allocation Inspired by a Model of Division of Labor in Social Insects”, *Bio Computation and Emergent Computing*, edited by D. Lundh, B. Olsson, Narayanan, Singapore: World Scientific, (1997), 36-45.

[5] M. Campos, E. Bonabeau, G. Théraulaz and J. L. Deneubourg, ”Dynamic Scheduling and Division of Labor in Social Insects”, *Adaptive Behavior*, Vol. 8-2, (2001), 83–92.

[6] M.B. Dias and A. Stentz, ”A Free Market Architecture for Distributed Control of a Multirobot System”, *6th International Conference on Intelligent Autonomous Systems*, Venice (Italy), (2000), 115–122.

[7] B. P. Gerkey and M. J. Matarić, ”Sold!: Auction methods for multi-robot coordination”, *IEEE Transactions on Robotics and Automation, Special Issue on Multi-robot Systems*, Vol. 18, No. 5, (2002), 758–768.

[8] J. Guerrero, G. Oliver and A. Ortiz, ”On Simulating Behaviour-based Robot Colonies in the Clasroom”, *First EURON Workshop on Robotics Education and Training*, Weingarden (Germany), (2001), 91–98 .

[9] M. J. Krieger, J. G. Billeter and L. Keller, ”Ant-like task allocation and recruitment in cooperative robots”, *Nature 406*, (2000), 992–995, .

[10] Kristina Lerman, Aram Galstyan, ”Mathematical Model of Foraging in a Group of Robots: Effect of Interference”, *Autonomous Robots 13 (2)*, (2002), 127–141.