Multi-Robot Task Allocation Method for Heterogeneous Tasks with Priorities

José Guerrero and Gabriel Oliver

Mathematics and Computer Science Department,
Universitat de les Illes Balears (UIB), Cra. de Valldemossa, km 7.5,
07122, Palma de Mallorca (SPAIN)
{jose.guerrero, goliver}@uib.es

Summary. Task allocation is a complex and open problem for multi-robot systems and very especially if a priority is associated to each task. In this paper, we present a method to allocate tasks with priorities in a team of heterogeneous 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 taking into account their priorities and characteristics. Thus, we can minimize the interference effect between robots and increase the system performance. The method has been extensively tested for a modification of the well-known foraging task, using different kinds of robots. Experimental results are presented to show the benefits of the proposed method.

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 achieve this aim. Among these problems, we focused on the task allocation aspects, that is, selecting the best robot or robots to carry out a task. As it has been demonstrated in different studies [1, 12], the number of robots has an important impact on the system performance, among other factors, due to the interference effect. Interference is the result of competition for the shared resources, especially the physical space. That is, two or more robots need to reach the same point at the same time. Besides, if a certain task captures the attention of an excessive number of robots, other tasks can be forsaken. This effect can increase if a priority is associated to each task. Therefore, a good task allocation mechanism must decide on the 'optimal' number of robots needed to carry out each task.

In this paper, we extend our decentralized method of task allocation for groups of heterogeneous robots. We mainly focus on deciding the optimal number of robots to execute each task when priority is associated to each one. Our method is inspired in both swarm systems, and, very especially auction-like methods. In most cases multi-robot researchers studied the interference effect only using methods based on the swarm intelligence paradigm. In these systems each robot decides the task to execute using only its own information. As will be shown later, the pure swarm based systems has some limitation. Our task allocation method mitigates some of these problems.

To test our system we use a foraging like task, where the robots must find a set of objects and carry them to a delivery point. A priority and a weight are associated to each object and each robot has a load capacity. Unlike the classical foraging task, multiple robots can cooperate to transport the same object. In this case we have to decide how many robots and which ones do we need to transport each object according to its priority and weight. This is a new task that has not been tested from the interference point of view.

The performance of a task allocation mechanism is closely connected to the diversity level of the team of robots, as will be shown later. To measure the heterogeneity of the group collectivity we use, among others, the social entropy proposed by T. Blach [3]. We will also study the relation between this metric and some of our architecture parameters. Finally, two different strategies or variations of our method are tested: 'preemption' and 'no preemption'. The preemption ability is the capacity of changing the task assigned to a robot. As will be exposed, among other advantages, if we use a 'preemption' strategy when a robot finds a high priority task it can ask for help other robots with a lower priority task.

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; finally, section 5 exposes 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.

Dias and Stenz [7] have proposed cooperation mechanisms based on explicit coordination between robots in the so-called market-based mechanisms. In the same line, Gerkey and Matari'c are working on auction-based mechanisms [8]. In this kind of systems, the robots act as self-interest agents and they bid for 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, the best robot for a specific task can be chosen, but they need communication mechanisms between robots.

Other papers proposed swarm intelligence inspired solutions. To implement these systems some authors make use of the response thresholds systems [4, 5, 11]. In these systems, each robot has a stimuli associated with each task to execute. When the level of the stimuli exceeds a threshold, the robot starts its execution. The pure threshold-based systems don't require any kind of communication mechanisms.

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.

Many authors [1, 12] try to study and solve the interference problem using swarm methods. For example, K. Lerman [12] studies mathematically the interference effect using this kind of systems. This work shows how system performance decreases as number of robots is incremented. Few systems have studied auction methods to solve the interference problem. These systems use the 'classical' foraging mission, where a single robot is assigned to each object and the set of robots and tasks are to be supposed homogeneous and without priorities. Other authors [6] use an auction like system, similar to our method, but the number of robots assigned to each task is predefined.

Our approach is partially inspired in the auction mechanisms and, consequently the best for a specific task can be chosen. However, while previous work cannot determine the optimal number of robots to execute a task, our method allows deciding this number as a function of the amount of work required to complete the task, the priority of this task and the work capacity of the robots involved in the auction process.

3 Working group leading, formation and updating

The coordination mechanisms description, including the groups' formation, the membership policy, the task assignment and the preemption strategies is briefly described in the following paragraphs. In this paper we focus on the issues related to the priorities. A more detailed description of our method without priorities can be found in [9].

3.1 Single robot to leader negotiation

This section describes the initial negotiation that corresponds to a non preemption strategy, and therefore it doesn't allow the exchange of robots (work capacity) between tasks.

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 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. 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 the optimum group size is. We propose the leader to decide using the following equation:

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

Where N is the number of robots of the group and $workCapacity_i$ is the individual work capacity of the ith robot. TH is the group threshold; this value is a parameter that will be used to compare the efficiency of the group formation policy.

$taskWorkLoad$ is the amount of work required to finish the assigned task that is calculated by the leader. And finally *priority* is the priority of the task. A high value of priority represents a task with a high priority. 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. As it can be seen, a task with a high priority will require more robots and, therefore, more work capacity than other with lower priority.

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. Initially, the leader informs to other robots that it needs to form a work group. Then, each robot without any assigned task sends to the leader its work capacity. Finally, the leader selects the robots with the highest work capacity provided inequality (1) is verified. 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. During this process a non-leader robot finds a new task and leaves its group. On the other hand, the size of the group can increase thanks to a robot aggregation process. Using this process a new member can be accepted in a group if, after a certain time has passed, this robot hasn't any task to execute. Thus, 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 tasks with equal or lower priority. To select robots from another group, 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 other groups. 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 to 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 is shown by a high energy value. The energy concept is similar to the stimulus intensity value used by some threshold based algorithms [4].

To select robots from other groups the leader uses another auction process. The robots from other groups bid during the auction process not only using its work capacity, like in the single robot to leader negotiation, but using the following value:

$$B = workCapacity * E_{g2}^{'} \tag{3}$$

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, B, is selected.

The goal of this selection is to 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. To get this objective, the leader only selects a robot 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:

$$MAX(E_{g1}, E_{g2}) > O * MAX(E_{g1}^{'}, E_{g2}^{'}) \tag{4}$$

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 new energies of the groups in the case that the transaction would take place. Finally, O is the percentual 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 action process finished when the equation (1) is fulfilled or when no more robots validate the equation (4). 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 and how the value of the group threshold, the social entropy of the group and the preemption strategy affect to the mission. We also evaluate the suitability of our system to carry out tasks with priorities.

4.1 Measure of the homogeneity of the collectivity

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, we use the simple entropy measure based on the Shannon's information entropy. On the other hand, we use the hierarchic social entropy as formulated by Balch [3]. This measure 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. This simulator implements robots that work according to a control architecture based on behaviours, as they were introduced by Ronald C. Arkin [2]. A detailed discussion about the RoboCoT architecture can be found in [10].

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. To maintain the initial conditions, when an object is transported to the delivery point immediately appears, randomly placed, another one, with identical characteristics. Figure 1 shows a typical 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. Thus, 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. Moreover, a priority is associated to each task. This priority is an integer value between 1 and 5, where 1 is the lowest priority and 5 is the highest. This priority value is the priority value, the $taskWorkLoad$ value is the object weight and the $workCapacity$ of a robot is its load capacity.

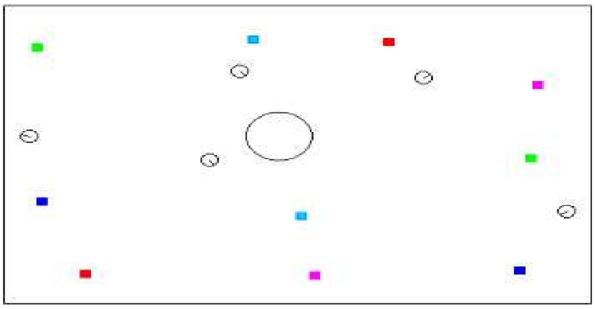

Fig. 1. Example of initial situation of the experiments

In all the experiments presented here, we have used ten objects to load and five robots. The weight of the objects is 30 units in all cases. During the experiments with priorities, we use two object of each priority, that is, two object with priority 1, two objects with priority 2, etc. 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 we have used four different configurations or combinations for the load capabilities of the robots, as it can be seen in table 1. For each configuration, we have used as values for the group threshold (TH): 0, 2, 4, 6 and 8. In the case TH=0, equation (1) has not been used, and therefore the number of robots per group is not limited. The robots carry out the mission during 35000 time units. After this period, we get the total weight transported, the average time required to finish each task, etc.

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	SH
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

4.3 Results without priorities.

During the first set of experiments we evaluate our system using a set of homogeneous objects. These objects have no priority associated. Thus, we can study the impact of the threshold value on the system. Figure 2 shows the total amount of weight transported by the robots using a 'non preemption strategy', that is, using only the leader to robot negotiation. Figure 3 shows the weight transported when the leader to leader negotiation is used.

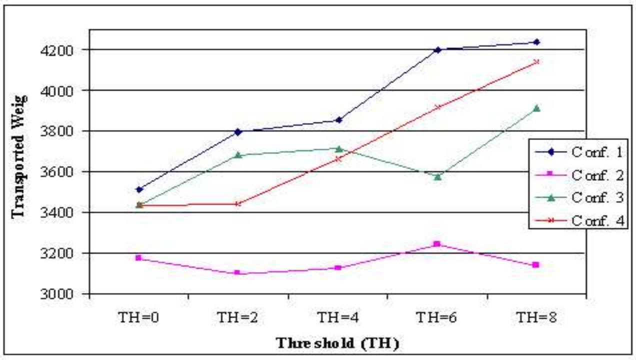

Fig. 2. Transported weight by the robots using a 'no preemption strategy'

As it can be seen in figure 2, in most cases equation (1) involves the reduction of the execution time compared to the system with null threshold. Moreover, for the most part, this benefit is increased as the threshold value is higher. A high value of TH reduces the number of robots assigned to each task and therefore reduces the interference effect. The configuration with the highest social entropy seems not to be affected by the threshold value and it presents the worst results. It can also be seen that preemption strategy leads to poorer results than non-preemption one. Therefore, the 'preemption strategy' seems to produce a self-defeating effect over the system.

Finally, figure 4 shows the average time required by the set of robots to gather single object without preemption. This time is computed from the moment a robot finds the object to the time the object is fully transported. The results using pre-

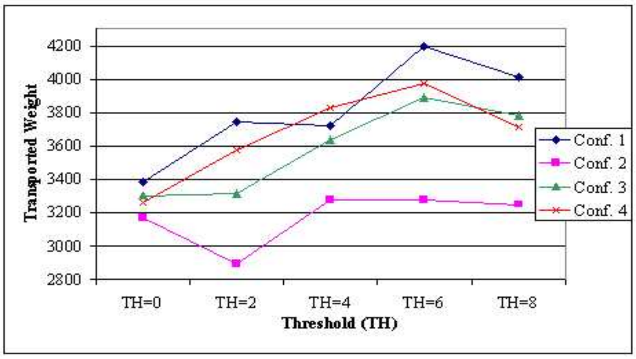

Fig. 3. Transported weight by the robots using a 'preemption strategy'.

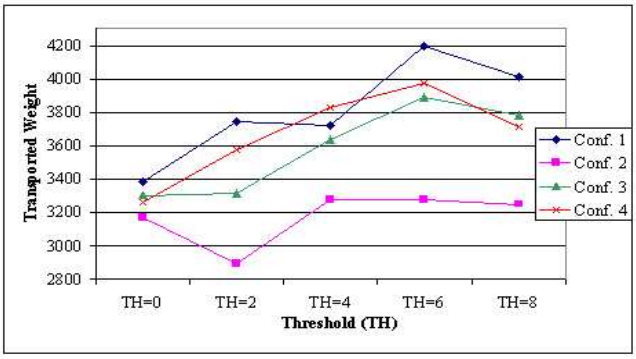

Fig. 4. Average time required to transport a full object

emption are similar. As it can be seen, the average time is increased as the threshold increases. If a priority is assigned to each task, this time could be very important.

4.4 Results of tasks with priorities.

We evaluated our method when a priority is associated to each task, as described in section 4.2. Figure 5 shows the average time required to completely gather an object as a function of its priority for different values of threshold without preemption. Figure 6 shows the same set of experiments but with preemption. In both cases we used the robots of the configuration 3. The graphic with the total weight transported is not shown because it is very similar to figure 3. As it can be seen, when we use a preemption strategy, the tasks with a high priority require less time than the tasks with a low priority. This benefit is increased as the threshold decreases. But, as it can be seen in figure 3, as the threshold increases the total weight transported is reduced. Therefore, we must find a balance between these factors to improve the performance of the system. In addition, when a non preemption strategy is used, this effect is not produced Sometimes tasks with low priority requires less time than the tasks with a higher priority.

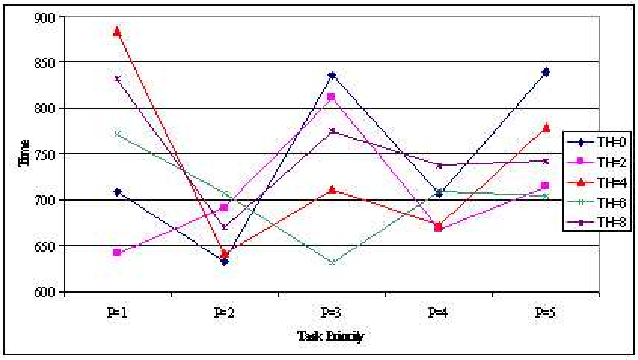

Fig. 5. Average time required to fully transport an object without preemption and with priorities

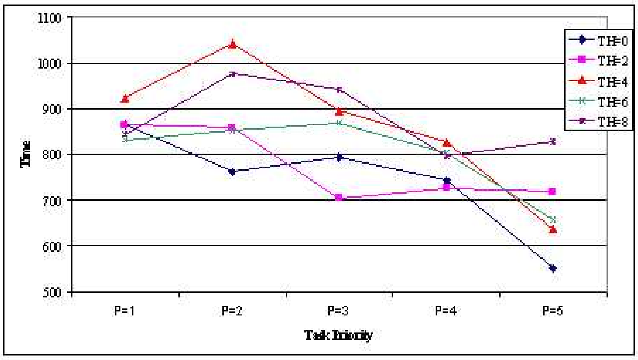

Fig. 6. Average time required to fully transport an object with preemption and with priorities

5 Conclusion and future work

This paper presents a simple and efficient way to solve the task allocation problem and, more specifically, to decide how many robots are needed to execute a specific task. Our algorithm adapts the classical auction one using some threshold-based systems concepts to find the optimal number of robots when a priority is associated to each task. 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 of robots as a function of the kind of the task, the priority of the task and the available robots. The execution results of our task prove that our mechanism increments the amount if objects transported during a foraging-like mission, very specially if a non preemption strategy is used. On the other hand, these results show that our system can reduce the average time required to transport the objects with a high priority, only if the preemption strategy is used.

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 the parameters of our architecture and the heterogeneity level of the robots. We are also working on developing a better definition of system stability. Finally, another aspect of the systems that should be improved is the use of a non fixed threshold like [5].

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

References

1. Agassounon W., Martinoli A. (2002). Efficiency and Robustness of Threshold-Based Distributed Allocation Algorithms in Multi-Agent Systems. 1st Int. Joint Conf. on Autonomous Agents and Multi-Agents Systems, 1090-1097.
2. Arkin R.C. (1998), Behaviour-Based Robotics, The MIT Press.
3. Balch T. (1997) Social Entropy: A New Metric for Learning Multi-Robot Teams, 10th international Florida Artificial Intelligent Research Society Conference, CA:AAAI Press.
4. Bonabeau E., Sobkowski A., Theraulaz G. and Deneubourg J. L. (1997) Adaptive Task Allocation Inspired by a Model of Division of Labour in Social Insects, Bio Computation and Emergent Computing.
5. Campos M., Bonabeau E. and Thraulaz G., Deneubourg J. L. (2001) Dynamic Scheduling and Division of Labour in Social Insects, Adaptive Behaviour Vol. 8-2.
6. Chamowicz L., Campos M., Kumar C. (2002) Dynamic Role Assignment for Cooperative Robots. IEEE International Conference on Robotics and Automation.
7. Dias M.B. and Stentz A. (2000) A Free Market Architecture for Distributed Control of a Multirobot System, 6th International Conference on Intelligent Autonomous Systems.
8. Gerkey B. P. and Mataric M. J. (2002) Sold!: Auction methods for multi-robot coordination, IEEE Transactions on Robotics and Automation, Special Issue on Multi-robot Systems, Vol. 18 No. 5
9. Guerrero J. and Oliver G. (2003) Multi-robot Task Allocation Strategies Using Auction-Like Mechanisms, 6th Catalan Conference on Artificial Intelligence (CCIA'03)
10. Guerrero J., Oliver G. and Ortiz A. (2001) On Simulating Behaviour-based Robot Colonies in the Classroom, First EURON Workshop on Robotics Education and Training.
11. Krieger M. J., Billeter J. G. and Keller L. (2000) Ant-like task allocation and recruitment in cooperative robots, Nature 406.
12. Lerman K., Galstyan A. (2002) Mathematical Model of Foraging in a Group of Robots: Effect of Interference, Autonomous Robots 13 (2)