Monitoring and Interference Impact in Multi-Robot Auction Methods *

José Guerrero and Gabriel Oliver
Universitat de les Illes Balears,
Mathematics and Computer Science Department
Cra. de Valldemossa, Km. 7,5, 07122
Palma de Mallorca, Spain
e-mail:{jose.guerrero, goliver}@uib.es

Abstract

Task allocation is one of the main problems in multi-robot systems. To get a good task allocation, we have to take into account, among other factors, the physical interference effect between robots, that is, when two or more robots want to access the same point at the same time. This paper analyzes the interference impact using auction methods. We will show how the performance of the auction utility function can be improved if the interference impact is included in it. We will also analyze the impact of the knowledge about the progress of the task on the auction process. It will be shown that, using the interference impact, the monitoring of the task process is not so necessary. This method has been tested using transport like tasks, where each object must be transported to a delivery point before a deadline. This is a simple task that allow us to isolate the interference effect under study.

Introduction

Multi-robot systems can provide several advantages over single-robot systems: robustness, flexibility and efficiency among others. To benefit from these potential aspects several problems have to be solved. Among all these problems, we focus on task allocation issues, that is, selecting the best robot or robots to execute a task. Some tasks require that two or more robots cooperate to execute them creating coalitions. In this case we have to find the best set of robots to execute the task and also the optimum number of these robots. As it has been demonstrated in different studies (Lerman & Galstyan 2002; Hayes 2002), the number of robots has an important impact on the system performance due to the physical interference effect, among other factors. Interference appears when two or more robots need to reach the same point at the same time. This factor has only been modeled and analyzed using very simple environments and using a specific architecture, like for example in (Lerman & Galstyan 2002). On the other hand, our method is based on external observations of the system behavior, and thus, it doesn't make any assumption about the architecture of the robots.

*This work has been partially supported by project CICYT-DPI2005-09001-C03-02 and FEDER fundings.

A lot of research has been done to solve the task allocation and coalition formation problems but they are still open problems. One of the most used and well studied task allocation solutions are auction methods (Dias & Stentz 2003; 2002; Gerkey & Mataric 2002; Kalra, Ferguson, & Stentz 2005). Only a few auction strategies, like (Vig & Adams 2005; Chaimowicz, Campos, & Kumar 2002), allow to allocate several robots to the same task, but these methods don't take into account the interference effect. Moreover, one of the main problems of all auction methods is to find or to learn a good utility function. The utility function depends on a lot of factors, and very specially on the interference effect. On the other hand, several work has been done to reduce the interference effect but without using auction mechanisms (Zuluaga & Vaughan 2005; Goldberg & Mataric 1997; Ostergaard, Sukhatme, & Mataric 2001; Agassounon & Martinoli 2002).

In this paper we analyze how to introduce the interference effect in the auction's utility functions, extending our previous work (Guerrero & Oliver 2006). We study the impact of this factor showing that the system performance can be improved when the interference is taken into account. Another problem that this paper tries to analyze is how does the monitoring of the task progress affects the system performance. The monitoring process can be a so complex task which requires sophisticated sensorial and communication capacities. The first experimental results show that using our interference model with the auction process, the robots don't need to monitor the task to know how to bid. This bid only depends on the initial conditions of the task. To our knowledge, this is the first time that this kind of analysis is made using auction strategies.

This paper also proposes a framework to reduce the complexity on finding utility functions when robots must create coalitions. This framework divides the learning process in three stages. During the first phase, only the knowledge of individual robots is included in the utility function. The second phase includes the knowledge about robots that form the coalition. Finally, the last stage includes the information of other coalitions. One of the most importat piece of information that should be included in the second phase is a measure of the physical interference produced between robots of the same group. 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. During this task, multiple robots can cooperate to transport the same object. In this case we have to decide how many robots and which ones we need to transport each object according to the object weight and to the robots characteristics. Each task, or object, must be executed before a deadline, and the goal of our auction method is to maximize the number of tasks executed before their deadline. The results show that including the interference information in the utility function, the system performance can be significantly improved.

The rest of this paper is organized as follows: the second section presents the task allocation algorithm used; the next section exposes the designed framework to simplify the search of good utility functions; section ”Group utility function: interference effect” analyzes the effect of the physical interference on the utility functions; section ”Task Allocation Experiments” shows the results of the task allocation system using the interference effect; finally, the last section exposes some conclusions and future work.

Task Allocation Method

Our task allocation mechanisms, including the groups' formation, membership policy and task assignment is briefly described in the following paragraphs. This new mechanism extends our previous work (Guerrero & Oliver 2004; 2006) to take into account deadlines and it also introduces the concept of partial knowledge about the task progress. Here we only expose in a very concise way the main aspects of our method to understand the interference model that will be explained in the ”Task Allocation Experiments” section.

A classical auction method has been modified to select which robots, and very specially, how many of them are needed to execute a task. In an initial stage, each robot is looking for a task. When a robot finds a new task, it will try to lead it. There is only one leader for each task. The details about how a robot can be promoted to leader, can be found in (Guerrero & Oliver 2004). If a robot is promoted to leader, it will create, if necessary, a work group; that is, a set of robots that will cooperate to execute this specific task. In that case, the leader must decide which the optimum group size is and what robots will be part of the group. To make this decision, the leader uses an auction like mechanism. During this process the leader will be the auctioneer and the other robots will bid using their work capacity. The work capacity is the amount of work that a robot can execute per time unit, thus, this value is the utility function of our auction method or the price that the robots want to pay to participate in the task. The leader selects the robots with the highest work capacity using a greedy algorithm, until it detects that the group is able to reach its deadline, that is, until this condition is verified:

$$DL_g = \frac{taskWorkLoad}{groupCapacity} \leq DL \quad (1)$$

Where $taskWorkLoad$ is the amount of work required to finish the assigned task that is calculated by the leader; $groupCapacity$ is the work capacity of the group, that is, the amount of work that the group can process each time unit and, finally, DL is the deadline of the task. As it can be seen, DL_g is the expected time required to finish the task. Therefore, the selection process is a very simple greedy method with a computational complexity of $O(n)$, where n is the number of robots.

If during the task execution the leader detects that the deadline (DL) can not be fulfilled, it starts a new auction process to get, if it's possible, new robots. This can happen if the leader monitors the task execution progress, that is, it knows at any time the current value of the $taskWorkLoad$ parameter, and, also, the initial $groupCapacity$ has not been correctly calculated. In general, it's not easy to get this value because the work capacity of the group is not the sum of the work capacity of each single robot, that is $groupCapacity \neq \sum_{1 \leq i \leq N} workCapacity_i$, where N is the number of robots of the group. This inequality is mainly due to the interference effect. During the following sections we will propose a method to calculate the individual utility of each robot and specially the group utility. From this point, we will consider the robot utility and the group utility synonymous of robot's work capacity and work capacity of the group.

A Framework to Get Utility Functions

This section describes a framework that will help us to get a good utility function for auction methods when robots must create coalitions. Getting a good utility function is a difficult process, very specially when the robots must form coalitions or when the utility of a robot depends on the utility of other robots. We can use learning algorithms to get this function, but these algorithms require a lot of time, and moreover, it is not clear what the robot has to learn. Also, in general, utility functions are not linear, so the learning process can be very hard. To simplify the process, some parameters can be previously analyzed, using an ideal environment, and then modified during the execution of the task. We will do this in 3 steps:

- Individual utility: during the first stage, we evaluate the characteristics of each single robot without taking into account the others. Here it will be include some characteristics like velocity, acceleration, etc.
- Group utility: in this step, the robot will take into account the other ones to create a coalition or working group. Here some parameters, like interference effect, will be included. That is, the robots will calculate the utility function of the group.
- Inter-Group utility: finally, just the leaders have to take into account that the decision of one group can affect other groups. This inter-group dependency must be included in the utility function during the final step.

In this paper we analyze the first and second step paying special attention to the interference effect. As an example of how to use the framework proposed, a transport like task will be used. 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

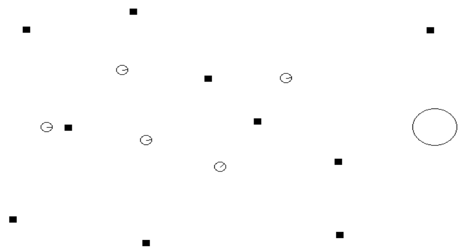

Figure 1: Example of initial situation of transport task

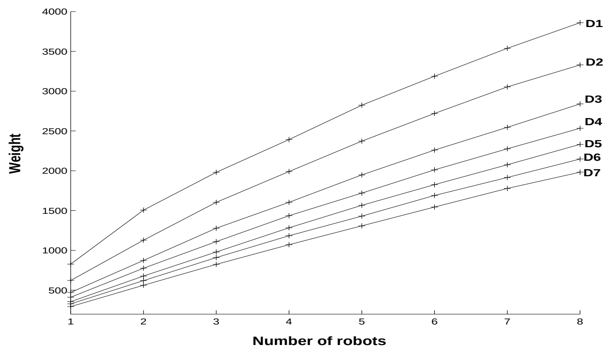

Figure 2: Total transported weight during the experiments to model interference effect

the squares represent the objects to collect, the delivery point is the big circle at the right 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 at once. 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. To maintain the initial conditions, when an object is fully transported to the delivery point, immediately appears another one, with identical characteristics in a random place. Of course, this is a very simple environment but it allows us to isolate the interference effect from other factors that can appear in more complex tasks. It is under study whether a similar reasoning can be made for different kind of tasks, like exploration, surface cleaning or mapping for example.

Individual utility function

We will now describe the first step of our framework, called individual utility, to find the utility function of each single robot. The transport task explained during the last section will be used as an example. The work capacity of a robot is the amount of object's weight that this robot can transport to the delivery point per time unit. Under ideal conditions, that is, assuming an open environment without any other obstacle or robot between the object and the delivery point, the robot's work capacity is easy to calculate. Let r_i be a robot and C_i the load capacity of the robot. V_i is the maximum velocity, d the distance between the object and the delivery point and C the weight of the object. The number of trips between the delivery point and the object that the robot must do to transport the full object is $2 * \frac{C}{C_i}$. If the acceleration and deceleration time is neglected, for each one of these trips the robots will need $\frac{d}{V_i}$ time units. We also consider that a robot needs one time unit to load and to unload each weight unit. For example, if the robot has to load 2 weight units it will require 2 time units to load all this weight and 2 time units more to unload it when it arrives to the delivery point. Thus, to load and to unload the full object, the robot will spend $2C$ time units. Therefore, the total time required to transport the full object is $T = 2\frac{C}{C_i}(C_i + \frac{d}{V_i})$ and the work capacity $\frac{C}{T}$ is:

$$workCapacity_i = \frac{C_i V_i}{2(C_i V_i + d)} \tag{2}$$

Group Utility Function: Interference Effect

During this section we will analyze the second step of our framework to get the utility function of the group. We will also use the transport task to show how to get this function value. The utility of the group will be defined as the expected amount of work that the set of robots can execute per time unit. This value is not the sum of each individual robots' work capacity because of the interference effect, among other factors. Thus, first of all, we will model the interference effect and then we will show how to use this information to get the utility function of the group. All the experiments carried out to model the interference have been executed using a multi-robot simulator called RoboCoT (Robot Colonies Tool). RoboCoT is a software tool developed by the authors at the University of Balearic Islands (Guerrero & Oliver 2001).

Interference Effect Analysis

To analyze the interference effect we have executed a task where several robots must transport a single object and the total weight transported by the robots after 40000 time units is calculated. All the robots have the same load capacity (2 weight units) and the same velocity (3 distance units/time unit), and therefore, they all have the same work capacity. Moreover, the environment doesn't have any obstacle but the robots, the object and the delivery point. Seven different distances between the object and the delivery point have been tested: $D_1 = 140$ units, $D_2 = 180$ units, $D_3 = 250$ units, $D4 = 280$ units, $D5 = 330$ units, $D_6 = 360$ units and $D_7 = 400$ units. Figure 2 shows the total transported weight during these experiments when the number of robots varies from 1 to 8. As it can be seen, and as has been pointed by other authors, the relation between the number of robots and the transported weight is not linear. The difference between the expected transported weight, calculated as the sum of the individual robot's work capacity, and the real transported weight can only be due to the interference. Figure 2 also shows that the interference effect increases as the distance between the object and the delivery point decreases. To analyze the interference effect, a polynomial fit model has been used.

The polynomial fit model suposes that the work capacity of the group follows this equation:

$$groupCapacity = \sum_{1 \leq i \leq N} workCapacity_i - I(N) \quad (3)$$

where $workCapacity_i$ is the individual work capacity of the ith robot of the group, calculated using equation 2; N is the number of robots of the group and $I(N)$ is a polynomial of degree 2 that fits the interference effect as a function of the number of robots. We have used this function because of its simplicity and because it fits with a very low error the experimental results. Moreover, due to its simplicity, only 3 parameters must be adjusted. We have also tested polynomials of higher degrees, but the results do not improve significantly the performance of the system. Thus, we assume that this polynomial models the difference between the expected work capacity without interference and the results of our simulations. Function $I(N)$ has the following form:

$$I(N) = \alpha N^2 + \beta N + \gamma \quad (4)$$

Table 1 shows the values of the parameters of function $I(N)$, and figure 3 shows the form of the $I(N)$ function that fits the real results. To improve the quality of the figure, only some distances have been represented (D_1, $D2$, $D3$ and $D7$). The crosses correspond to real data. The y axis represents the interference effect for every 1000 time units, that is, $1000 * I(N)$. The resulting parameters seem to be very low, but it should be pointed out that the utility of each robot is also very low because of the high values of the distance value (d) of the individual utility equation, as expressed in equation 2. However, the interference effect can modify very significantly the group utility. For example, when the distance is D_7 and there are 8 robots, the interference decreases the work capacity of the group down to a 58%. Moreover, as it can be seen in table 1, as the distance between the object and the delivery point increases, the values of α and β decrease. For the time being, a new function which relates the interference to the distance and the number of robots is under study. Finally, we have to note that the errors between the real results and the interference function fitted are, in general, very low but they increase as the number of robots is reduced. In fact, our equation is not suitable when the distance between the object and the delivery point is very low and $N = 1$.

-	α	β	γ
D_1	0.3589	7.5029	-12.3571
D_2	0.3476	2.964	-4.2857
D_3	0.2432	1.3347	-2.0107
D_4	0.2006	1.062	-1.6321
D_5	0.1619	0.4737	-0.6071
D_6	0.1494	0.37337	-0.5071
D_7	0.1071	0.405	-0.5

Table 1: Parameters of the interference function for each 1000 time units($I(N)$)

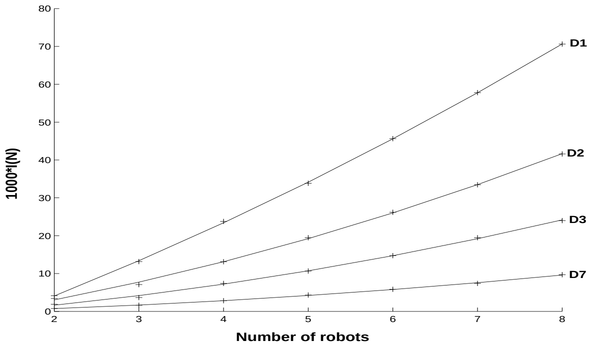

Figure 3: Polynomial fit of $I(N)$ for different values of distance between the object and the delivery point

Using all this information, the utility function of a robot, calculated in equation 2 must be modified. In fact, for a group with N robots, this new utility measure can be defined as the difference between the utility of the working group with and without the Nth robot. Therefore, this equation must be used:

$$utility_j = groupCapacity_t(N) - groupCapacity_{t-1}(N-1) \quad (5)$$

Where $groupCapacity_{t-1}(N-1)$ is the work capacity of the group before the new robot is added; $groupCapacity_t(N)$ is the group capacity with the new robot and N is the number of robots of the group including the new one. Therefore, if this equation is calculated, we can find that the utility of a new robot, $utility_j$, is:

$$utility_j = workCapacity_j - (\alpha(2N-1) - \beta) \quad (6)$$

Using the market based system vocabulary (Dias & Stentz 2003), we can say that the left side of the equation 6 is the benefit that the robot will get if it executes the task and the right side is the cost of this execution. We can also note that the utility of each single robot only depends on two values, α and β. Thus, future learning algorithms will only need to tune this two parameters, instead of making large searches in unknown state spaces.

Interference and Task Allocation

This section will show how to modify the auction process explained in ”Task Allocation Method” section to take into account the interference effect.

As it has been explained in the second section, the leader selects the best robots (robots with the highest bids) until the equation 1 is verified. Now, using the interference information, the leader of the group will only include a new robot if this operation increases the work capacity of the group. Therefore, the auction process will finish when the equation 1 or when this condition is verified. Thus, the bidding process will now consist on the evaluation of 6. The first term ($workCapacity_j$) will be sent by each robot and the final term will be calculated by the leader. Using this extra information about interference, and as it will be seen during the next section, the robot can find a better set of robots to verify the deadline.

The Monitoring Process

During the execution of a task the leader can periodically receive information about the remaining weight of the object ($taskWorkLoad$) to be transported. If available, the leader of the task uses this information to make a guess about the actual $taskWorkLoad$ using a simple linear equation like:

$$WL(t) = WL(t_m) - groupCapacity * (t - t_m) \quad (7)$$

Where t_m is the time when the leader receives information about the task process and $WL(t)$ is the expected $taskWorkLoad$ at instant t.

During a continual monitoring task progress execution, the leader knows in each moment the exact value of the $taskWorkLoad$, and therefore, it can start another auction process if inequality 1 is not verified. In a no monitoring task process execution, the leader only knows the $taskWorkLoad$ at the beginning of the task, and then it uses equation 7 to predict the $taskWorkLoad$, with a unique constant $WL(t_m)$ during the whole process.

Task Allocation Experiments

In this section we will show the results of several experiments performed to study the impact of the physical interference on our auction method. We will also analyze how the monitoring process affects to the system performance. During all the experiments RoboCoT has been used, which is the same simulator used in the last section. The robots must execute the transport task explained in the second section. The transport tasks have a deadline. The main objective is to transport each object before its deadline. If the fulfilment of this objective is not possible, the robots continue their execution until the object is fully transported. The time to deadline starts when the object appears in the environment. Thus, we give priority to the accomplishment of the tasks' deadline over the increment of the total transported weight. To simplify the analysis, the robots know the situation of each object in the environment.

During all the experiments we use 10 robots and 3 objects to gather. All the robots have the same characteristics as in the experiments of the last section, that is, their load capacity is 2 and the maximum velocity is 3. All the tasks have a weight equal to 40 weight units. Three different kind of experiments have been executed: greedy robot selection, continual monitoring task progress and no monitoring task progress. In all the cases the value of the deadline is equal to 1200 time units, other deadline values have been tested in our previous work (Guerrero & Oliver 2006). This deadline value is the same for all the tasks, so if there is a long distance between the object and the delivery point, this task will require more robots than a nearer one. In greedy robot selection experiments all the leaders try to create a working group as great as possible without taking into account the deadline value or the task characteristics, that is, the leader tries to get as many robots as possible. Robots carry out the mission during 30000 time units. After this period, we get the time required to transport each object and the number of object gathered. Despite having only 3 objects in the environment, when an object is fully transported to the delivery point, it immediately appears another one in a random place. Therefore, the number of objects gathered can be greater than 3. Each experiment has been repeated 4 times.

Figure 4 shows the percentage of tasks that fulfill a deadline equal to 1200 time units during the execution of the greedy robot selection experiments. The bar with a label 0.6 represents the percentage of tasks that its execution time exceeds a 60% of the deadline. The bar with a label 0.5 represents the percentage of tasks that require less than a 60% and more than 50% of the deadline time, etc. The negative numbers represent the tasks that have been fulfilled the deadline. For example, the bar with -0.2 represents the tasks that require to finish less than a 20% and more than a 30% of the deadline time. During these experiments 297 objects were fully transported to the delivery point. A 72,7% of the tasks were executed before the deadline.

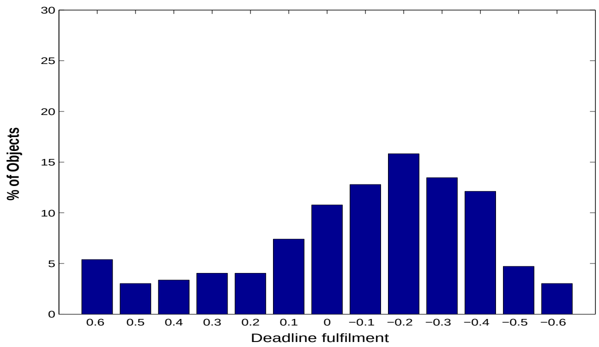

Figure 4: Deadline fulfilment during greedy robot selection experiments and with a deadline equal to 1200 time units

Figure 5 shows the results of the continual monitoring task progress experiments with a deadline equal to 1200 time units, taking into account the interference effect in the robot's work capacity. Thus, to calculate the work capacity of the group, equation 3 has been used. In this case a 98,8% of the tasks have been executed before the deadline, a 26,1% more objects than during the greedy robot selection experiments. Moreover, during the experiments, 323 objects were fully transported, a 8,8% more objects than with the last experiments. Another set of experiments has been executed only taking into account the continual monitoring of the task, but not using the interference effect model. The results obtained are not presented here but they are extremely similar to these previously shown in figure 5

The results of the experiments without monitoring the task progress can be seen in figures 6 and 7. As in the previous cases, the deadline is equal to 1200 time units. Figure 6 shows the results without modeling the interference effect, that is to say, to calculate the work capacity of the group equation 2 has been used. In this case only a 66,8% of the tasks fulfilled the deadline, less tasks than during the greedy experiments. On the other hand, the number of tasks that

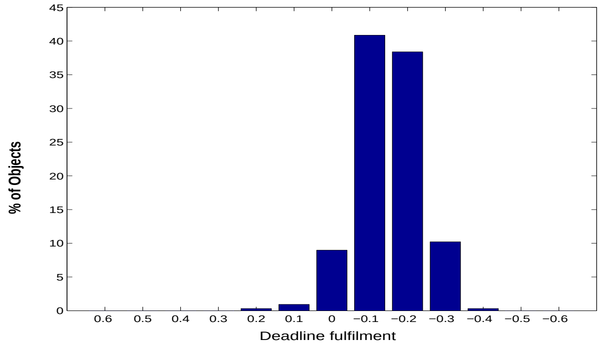

Figure 5: Deadline fulfilment during continual monitoring task progress experiments, using interference effect and with a deadline equal to 1200 time units

require a lot of time to finish has decreased with regard to greedy experiments. For example, now there are no tasks requiring more than a 50% of the deadline time to finish, but during the greedy experiments a 5,4% of tasks required this time. The total number of objects transported during these experiments was equal to 286. Finally, figure 7 shows the results of the no monitoring task progress experiments, but taking into account the interference effect. During these experiments the percentage of tasks that fulfill the deadline was equal to 93,2%. Thus, the number of tasks that fulfill the deadline has been increased a 26,4% with regard to the system that don't use interference effect. Therefore, the interference factor $I(N)$ seems to be useful. On the other hand, we can see that the monitoring process can improve the system performance, but it doesn't produce a great benefit. The number of tasks that fulfill the deadline has been increased a 5,6% using continual monitoring compared to the system that do not uses it. Therefore, using an interference model the monitoring process can be avoided. The reader should remember that for the continual monitoring the robots need to be continuously sensing the task state, while the interference effect can be modeled off-line.

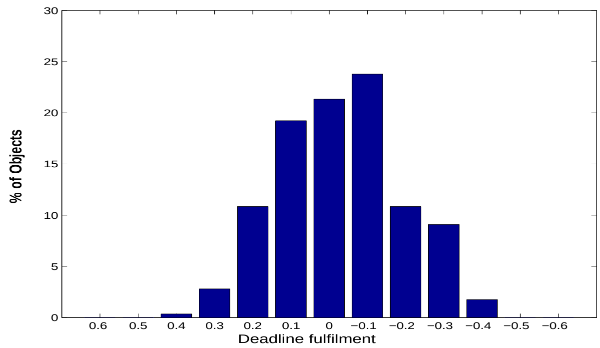

Figure 6: Deadline fulfilment during no monitoring the task progress without using interference effect and with a deadline equal to 1200 time units

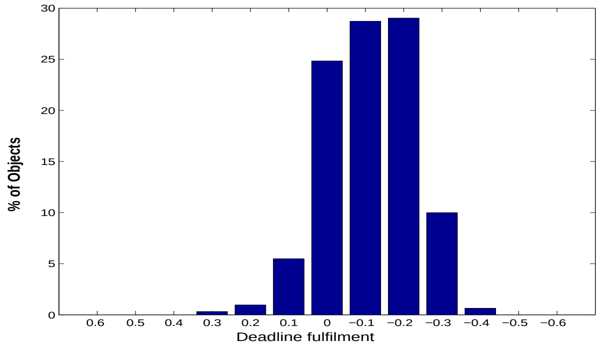

Figure 7: Deadline fulfilment during no monitoring the task progress, using interference effect and with a deadline equal to 1200 time units

Conclusion and Future Work

This paper analyzes the impact of the interference effect on the utility function used in an auction like system. It also studies how monitoring the task progress can affect to our method. First of all, an auction method has been presented that, unlike most of other auction methods, allows to assign multiple robots to the same task creating coalitions. Moreover, our method includes the concept of deadline, that is, the idea that a task should be executed, if possible, before a certain period of time. One of the main problems of all the auction systems is to find a good utility function. To simplify this problem when the robots must create coalitions, we propose a framework that divide the search in 3 steps. The first and the second step of this framework have been studied for the execution of transport like tasks. One of the main aspects that we have to take into account to calculate the utility function is the physical interference between robots. This influence has been analyzed and fitted using a polynomial function. The experiments carried out show that, including interference in the utility functions, the robots can better fulfil the tasks's deadline. Thus, the importance of interference has ben showed. Moreover, it seems that using the interference factor during the auction process, the leader can predict better the evolution of the task and, thus, a monitor system is not required. We have to take into account that monitoring the task progress can be a very hard process.

The work presented is in progress and has some challenging aspects to add and to improve. We are working to use a preemption auction method, that is a method that allows the exchange of robots between working groups. We will also study the interference effect between robots that belong to different groups, and thus, complete the last step of the framework presented in "A Framework to Get Utility Functions" section. Also, a deeper analysis of the monitoring effect over the system, using different deadline values, is necessary. Moreover, learning algorithms will be introduced to find out other parameters of our system. Finally, we will extend these experiments using real robots and other kind of tasks, like exploration and environments with obstacles. During these new experiments other factors, like the energy of the robot, will be taken into account to select the best robots for each task. We hope that some concepts from the classical real time systems literature will help us to formalize and to improve the system performance.

References

Agassounon, W., and Martinoli, A. 2002. Efficiency and robustness of threshold-based distributed allocation algorithms in multi-agent systems. In *Proceedings of First Int. Joint Conf. on Autonomous Agents and Multi-Agents Systems*, 1090–1097.

Chaimowicz, L.; Campos, M. F. M.; and Kumar, V. 2002. Dynamic role assignment for cooperative robots. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, 292–298.

Dias, M. B., and Stentz, A. 2002. Opportunistic optimization for market-based multirobot control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, volume 3, 2714–2720.

Dias, M. B., and Stentz, A. 2003. Traderbots: A market-based approach for resource, role, and task allocation in multirobot coordination. Technical Report CMU-RI-TR-03-19, Carnegie Mellon University, The Robotics Institute, Pittsburgh.

Gerkey, B. P., and Mataric, M. 2002. Sold!: Auction methods for multi-robot coordination. *IEEE Transactions on robotics and Automation, Special Issue on Multi-robot Systems* 18(5):758–768.

Goldberg, D., and Mataric, M. J. 1997. Interference as a tool for designing and evaluating multi-robot controllers. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, 637–642.

Guerrero, J., and Oliver, G. 2001. On simulating behaviour-based robot colonies in the classroom. In *First EURON Workshop on Robotics Education and Training*, 91–98.

Guerrero, J., and Oliver, G. 2004. Multi-robot task allocation method for heterogeneous tasks with priorities. In *7th. International Symposium on Distributed Autonomous Robotic Systems*.

Guerrero, J., and Oliver, G. 2006. Physical interference impact in multi-robot task allocation auction methods. In *IEEE workshop on distributed intelligent systems (accepted for presentation)*.

Hayes, A. T. 2002. How many robots? group size and efficiency in collective search tasks. In *Proceedings of the 6th Int. Symp. on Distributed Autonomous Robotic Systems DARS-02*, 289–298.

Kalra, N.; Ferguson, D.; and Stentz, A. T. 2005. Hoplites: A market-based framework for planned tight coordination in multirobot teams. In *Proceedings of the International Conference on Robotics and Automation*, 1170–1177.

Lerman, K., and Galstyan, A. 2002. Mathematical model of foraging in a group of robots: Effect of interference. *Autonomous Robots* 13(2):127–141.

Ostergaard, E. H.; Sukhatme, G. S.; and Mataric, M. J. 2001. Emergent bucket brigading - a simple mechanism for improving performance in multi-robot constrained-space foraging tasks. In *Proceedings of the 5th. International Conference on Autonomous Agents*, 29–30.

Vig, L., and Adams, J. A. 2005. A framework for multi-robot coalition formation. In *Proceedings of the 2nd Indian International Conference on Artificial Intelligence*.

Zuluaga, M., and Vaughan, R. T. 2005. Reducing spatial interference in robot teams by local-investment aggression. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2005*, 2798–2805.