INTERFERENCE MODELIZATION IN MULTI-ROBOT AUCTION METHODS [1]

José Guerrero and Gabriel Oliver

Universitat de les Illes Balears, Mathematics and Computer Science Department, Palma de Mallorca, Spain, e-mail:{jose.guerrero, goliver}@uib.es

Abstract: Task allocation is one of the main problems in multi-robot systems, very specially when the robots form coalitions. To get a good task allocation, we have to take into account, among other factors, the interference effect or the interaction between robots. Interference is not easy to modelize because it depends on a lot of dynamic factors. This paper models the interference impact using auction methods and support vector regression. We will show how the performance of the auction utility function can be improved if the interference impact is included in it. This method has been tested using transport like tasks taking into account an specific interference called physical interference.

Keywords: multi-robot, task allocation, coalition, auction, transport, learning

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 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 calculate the utility of the coalition. Moreover, when the tasks must be executed before a deadline we also need to know, how much time will the coalition need to finish the task. If we assign too few resources (robots) to a task, then the deadline can not be verified. On the other hand, if a certain task captures the attention of an excessive number of robots, other tasks can be forsaken.

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 and Stentz, 2003). Only a few auction strategies allow to allocate several robots to the same task, but these methods don't take into account the interference effect. Some other authors, that don't use auction, like (Fua and Ge, 2005), also suppose that the utility of a coalition of robots is the sum of the utilities of its members, which in general is not true because of the interference. Therefore, our challenge is to model the interference to get a coalition utility function as good as possible. In our case this utility function will be the time required by the set of robots to finish the task. Thus, we can assign the appropriate number of robots to fulfil the task before the deadline. 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

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

sensorial and communication resources. To our knowledge this is the first time that a study like this has been made in a multi-robot environment.

Extending our previous work (Guerrero and Oliver, 2006), we use support vector regression (SVR)to tune the interference of a coalition. This learning method has been used by other authors (Jones *et al.*, 2006) with an auction process, but in a very different environment. There are different kinds of interferences, among all of them we focus on physical interference effect, produced when two or more robots need to reach the same point at the same time. As it has been demonstrated in different studies (Lerman and Galstyan, 2002), the physical interference has an important impact on the system performance. To test our system we use a foraging like task, where multiple robots can cooperate to transport the same object.

The rest of this paper is organized as follows: the section 2 formalizes this problem; section 3 presents the task allocation algorithm used; section 4 explains the SVR technique that has been used; section 5 presents how to use SVR to model the physical interference effect; section 6 shows the results of the experiments; section 7 explains some conclusions and future work is stated.

2. PROBLEM STATEMENT

During this section we will formalize the task allocation problem explained in the previous section and will explain the main problems that it presents.

The task allocation is defined as follows: we want to allocate a set of tasks to a set of robots. Each task t_i has a workload ($taskWorkLoad_i$) that represents the amount of work required to finish the task. For example, if the robots must transport an object, this value will be the weight of that object, or if the robots must clean a surface, the workload will be the area to clean. Moreover, each task must be executed, if it is possible, before a deadline DL_i. Each robot (r_i) has an individual work capacity ($workCapacity_i$) that represents the amount of task work load that the robot can process per time unit. The tasks can be executed by a group of robots, which form a coalition. Thus, we have to know if the group of robots can fulfil the deadline. To that end, we need to calculate the work capacity of the group as a whole ($groupCapacity$), that is, the amount of work that the group can perform per time unit. In general, it is not true that the work capacity of the group is the sum of the work capacity of each single robot. Thus, the real value of the group capacity is:

$$groupCapacity = idealModel + I \qquad (1)$$

Where $idealModel$ is the model of the group capacity without interference and I is the interference factor. This additive model is very simple and efficient. The $idealModel$ can be easily represented when the individual capabilities of the robots are known. As it has been explained before, the sum of the individual utilities of each robot is often used. Usually, these utilities are independent of the interference factor and therefore they can be added. On the other hand, I is not easy to calculate because it depends on a lot of dynamic factors: the number of robots, the environment, etc.

3. TASK ALLOCATION

The 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 and Oliver, 2004) 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 the 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 and 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 a 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 take 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 the 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:

$$DL_g = \frac{taskWorkLoad}{groupCapacity} \leq DL \qquad (2)$$

Where DL is the deadline of the task. As it can be seen, DL_g is the expected time required to finish the task. If during the task execution the leader detects that the deadline (DL_g) can not be fulfilled, it starts a new auction process to get new robots, if it's possible. From now on, we will consider the robot utility and the group utility synonymous of robot's work capacity and work capacity of the group.

4. INTERFERENCE ESTIMATION: SUPPORT VECTOR REGRESSION

This section describes the process to predict or to learn the value of the interference using support vector regression (SVR) (C.Chang and Lin, 2001). The goal of this method is to find a function as flat as possible $f(x)$ that fits a given set of training data (x_i, y_i), $i = 1..l$ where $x_i \in R^n$ represents the input data of $f(x)$ and $y_i \in R$ are the results. In our case, the x_i vectors represent the coalition characteristics including both coalition robot and environment features. From this point forward, we will call to this coalition vectors. Of course, the dimension of these vectors can be very large, but as it will be explained, this dimension can be reduced if we include some previous knowledge about the interference characteristics. To implement the support vector regression we have used the *libsvm* library developed by National Taiwan University (C.Chang and Lin, 2001). This library creates a model file from a given set of training data. We have use the $\varepsilon - SVR$ method and and a radial basis function as the kernel function.

E. Jones in (Jones *et al.*, 2006) also applies the same SVR library to allocate tasks in the Robocup Rescue Simulation League but in that case, only a single robot can be assigned to the same task. Moreover, Jones' method doesn't want to get the expected time to finish a task, it only tries to find the individual utility function for each agent taking into account the expected incoming tasks and the their deadline.

5. PHYSICAL INTERFERENCE: THE TRANSPORT TASK

In this section we will analyze the specific kind of interference that we want to solve that is, the physical interference. This interference appears when two or more robots want to reach the same point at the same time. As it has been explained earlier, this situation has a great impact on the system performance. As an example of how to model the interference using SVR, a transport like task is used. This task 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 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. This weight is the $taskWorkLoad$ of equation 2. 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. 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.

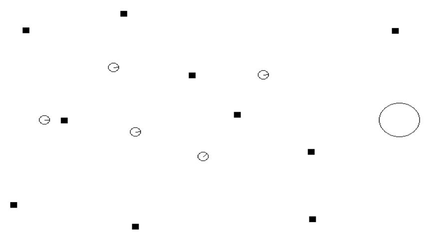

Fig. 1. Example of initial situation of transport task

In the next subsections we will describe how to get the individual utility for each robot, how the $idealModel$ of equation 1 is calculated and, finally, how to model the physical interference in these transport tasks.

5.1 Individual Utility Function

We will now describe how to find the individual utility or the individual work capacity of each single robot. The transport task explained earlier 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 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 robot needs $\frac{d}{V_i}$ 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:

$$workCapacity_i = \frac{C_i V_i}{2(C_i V_i + d)} \quad (3)$$

Thus, the $idealModel$ of equation 1 will be equal to $\sum_{1 \leq i \leq N} workCapacity_i$, where N is the number of robots of the group. Of course, this value of work capacity is only an estimation with several errors, but these errors will be included into the interference effects.

5.2 Group utility: Interference Effect

During this section we will analyze how to calculate or to learn the physical interference between robots, the I value of equation 1. 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. Ten different distances between the object and the delivery point have been tested and the number of robots varied from 1 to 8. All these experiments 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.

Figure 2 shows the total transported weight during these experiments when the number of robots varies from 1 to 8 and for the following values of distance between object and delivery point: $D_1 = 140$ units, $D_2 = 180$ units, $D_3 = 250$ units, $D4 = 280$ units, $D5 = 330$ units, $D_6 = 360$ units and $D_7 = 400$ units. 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. The interference effect can modify 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%.

Using the data of figure 2 and knowing the $idealModel$ we can calculate the value of interference I for each situation. This value only take into account the interference between robots of the same working group, as it will be explained

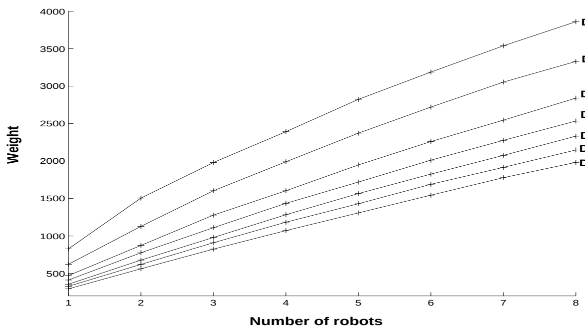

Fig. 2. Total transported weight during the experiments to model interference effect

later, in future works we will include the interference between groups. All this information is now included in the training data of the SVR method. The coalitions vectors (x_i values) will include 2 features: the $idealModel$ value, calculated using equation 3, and the number of robots of the group. The calculated values of I are used as the expected results of $f(x)$ (y_i). A total of 80 training data pairs $(x_i,\ y_i)$ has been used. All the information is used by the *libsvm* library to create the model of the system.

As it has been explained, the leader selects the best robots (robots with the highest bids) until the equation 2 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 2 or when this condition is verified. Thus, the bidding process will now consist on the evaluation of $workCapacity_i + (I' - I)$ for each robot r_i. Where I' is the predicted interference if this robot is selected by the leader and I is the current interference. Using this extra information about interference, and as it will be seen during the next section, the leader can find a better set of robots.

5.3 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:

$$WL(t) = WL(t_m) - groupCapacity * (t - t_m) \quad (4)$$

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 2 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 4 to predict the $taskWorkLoad$, with a unique constant $WL(t_m)$ during the whole process. Similar experiments have been carry out with different monitoring periods but their results are not included in this paper.

6. 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 using SVR. We will also analyze how the monitoring process affects to the system performance. During all the experiments RoboCoT has been used. The robots must execute the transport task explained in the second section. 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. Moreover, when a robot has no task to execute it stops.

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. 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 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. 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. Three different values of deadline have been tested: 1500 time units, 1200 time units, and 900 time units. All the tasks have the same deadline value.

Figure 3 shows the percentage of tasks that fulfill a deadline equal to 900 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 a 43,3% of the tasks were executed before the deadline.

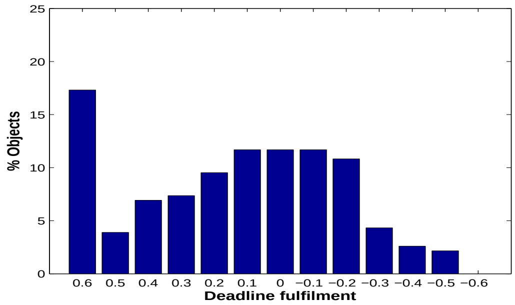

Fig. 3. Deadline fulfilment during greedy robot selection experiments and with a deadline equal to 900 time units

The results of the experiments without monitoring the task progress can be seen in figures 4 and 5. As in the previous cases, the deadline is equal to 900 time units. Figure 4 shows the results without modeling the interference effect, that is to say, only taking into account the $idealModel$ of the task. In this case only a 23,9% of the tasks fulfilled the deadline, less tasks than during the greedy experiments. On the other hand, the number of tasks that require a lot of time to finish has decreased with regard to greedy experiments. For example, now there are about a 7% of the task that require more than a 50% of the deadline time to finish, but during the greedy experiments a 21,2% of tasks required this time. Moreover, the total distance covered by all the robots has decreased a 10.8%. Figure 5 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 54,3%. Thus, the number of tasks that fulfill the deadline has been increased a 30,5% with regard to the system that don't use interference model. Therefore, the interference factor I seems to be useful. Also, the total distance covered by the robots decreases a 18,9% with regard to the greedy strategy.

Figure 6 shows the results of the continual monitoring task progress experiments with a deadline equal to 900 time units, taking into account the interference effect. In this case there are less tasks that fulfil the deadline, a 30,3%, with regard to greedy experiments. This effect may be because the task progress evolution is not constant over the time and can exist local minimums. This effect

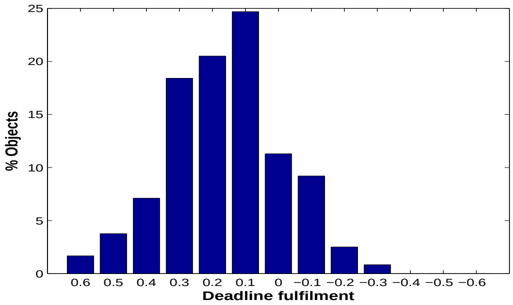

Fig. 4. Deadline fulfilment during no monitoring the task progress without using interference effect and with a deadline equal to 900 time units

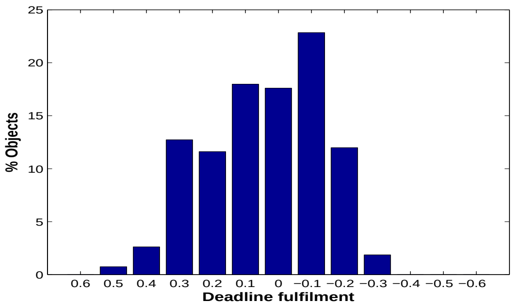

Fig. 5. Deadline fulfilment during no monitoring the task progress, using interference effect and with a deadline equal to 900 time units

make the problems of the monitoring processes clear.

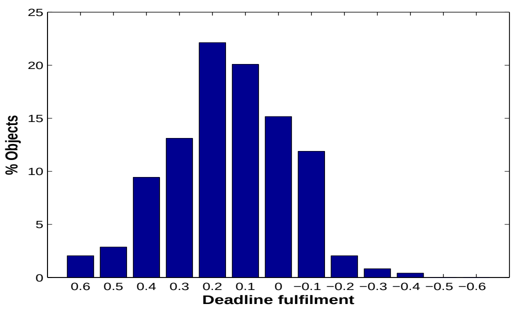

Fig. 6. Deadline fulfilment during continual monitoring task progress experiments, using interference effect and with a deadline equal to 900 time units

During the execution of the experiments with a deadline equal to 1200 time units, the interference value only increases a 10,1% the number of tasks that fulfil the deadline. Finally, we have to note that when we a deadline equal to 1500 was used, there was no significant differences between the experiments executed with and without interference, or between with or without monitoring. The main result is that with regard to a greedy algorithm our method can decrease the total distance covered by the robots up to a 19,3%.

7. 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. We model the interference between the robots in a coalition using support vector regression. The experiments show that with this off-line learning method the robots can better fulfil the tasks's deadline. Our method has been tested with transport task modeling the physical interference effect.

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 we will modify our system to allow on-line learning. Moreover, we will analyze other regression techniques without parameters to improve our results. Finally, we will extend these experiments using real robots and other kind of tasks, like exploration, cleaning, etc.

REFERENCES

C.Chang and C. Lin (2001). *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Dias, M. and A. Stentz (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.

Fua, Cheng-Heng and Shuzhi Sam Ge (2005). Cobos: Cooperative backoff adaptive scheme for multirobot task allocation. *IEEE Transactions on Robotics* **21**(6), 1168–1178.

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

Guerrero, J. and G. Oliver (2006). Physical interference impact in multi-robot task allocation auction methods. In: *IEEE workshop on distributed intelligent systems*. PRAGUE, (Czech Republic). pp. 19–24.

Jones, E., M. Dias and A. Stentz (2006). Learning-enhanced market-based task allocation for disaster response. Technical Report CMU-RI-TR-06-48. Carnegie Mellon University.

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