Auction like Task Allocation and Motion Coordination Strategies for Multi-Robot Transport Tasks

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

Abstract. In this paper we present a task allocation method based on auction mechanisms that allows to find how many robots are needed to execute a task. This number is unknown and depends on several factors. There are also different types of tasks that must be executed using different skills of the robots. It is very difficult to find a correct allocation under this conditions and at present it is an open problem. We also propose two motion coordination methods to reduce the interference effect between robots. To test our system a modification of the well know foraging task has been used. This task introduces special characteristics, not directly studied in previous work, that our method try to solve.

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 these problems, our work tries to minimize the physical interference between robots. Interference is the result of competition for the shared resources, especially the physical space. It's well known that interference has an important impact on the system performance. To reduce this impact, first of all, we have to decide the optimum number of robots to execute each task, using a task allocation method. Moreover, during the execution of the task the robots must coordinate their motion.

During this paper we extend our previous task allocation method, proposed in [5, 6] to allow more complex tasks. Now the robots have different skills to carry out the tasks with different characteristics. Our method is inspired in both swarm systems, and, very especially auction-like methods. Moreover, it extends the current auction mechanisms to find the number of robots needed to execute a task. This number is unknown a priori and not fixed. The current auction mechanisms doesn't take into account this characteristic. In this paper we also analyze some very simple methods to coordinate the motion of robots in order to reduce the interference between them. Two coordinate methods has been proposed: *follow the header* and *stay on the side*. These methods try to take into account the special characteristics of the foraging task used, that will be explained later. To our knowledge no other work studies the impact of the motion coordination strategies over the interference using the transportation tasks that will be used during this paper. To test our system we use a foraging like task, where the robots must find a set

Guerrero J. and Oliver G. (2005).
Auction like Task Allocation and Motion Coordination Strategies for Multi-Robot Transport Tasks.
In *Proceedings of the 1st International Workshop on Multi-Agent Robotic Systems*, pages 80-87
DOI: 10.5220/0001195000800087
Copyright © SciTePress

of objects and carry them to a delivery point. The robots have different load capacities (or skills) for each type of object. For exemple, one robot can have a high load capacity for tasks of type 1 and very low for other kind of tasks. 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, weight and type.

The rest of this paper is organized as follows: section 2 presents some relevant previous work; sections 3 and 4 describe our task allocation and motion coordination methods; section 5 shows the experiments carried out to validate the different approaches; finally, section 6 exposes some conclusions and future work.

2 Related Work

Our task allocation mechanism is inspired in both the auction processes, that use explicit communication between robots, and response threshold systems. Classical auction mechanism can only assign a robot to each task [3]. Other authors like L. Chamowicz in [2] use an auction like system, similar to our method, but the number of robots assigned to each task is predefined. Swarm systems [1] can use a response threshold mechanisms where multiple robots can be assigned to each task. A disadvantage of this kind of 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. Our task allocation mechanism try to solve these two problems. On the other hand, to reduce the interference effect we can use motion coordination strategies. Mataric explained in [8] some potential advantages of flocking in classical foraging tasks, but no experimental results were given. Finally, reference [4] demonstrated that during a classical foraging if the number of robots that simultaneously could access to the delivery point is limited the interference is also reduced.

3 Task Allocation

Our task allocation mechanism has been detailed in [5, 6]. In the following paragraphs it is briefly presented just to clarify the main concepts that are used in the rest of the paper. Moreover, this previous work has been extended to accept robots with multiple skills and different types of tasks. Our mechanism modifies the classical auction methods 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. 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 take this decision the leader uses an auction like mechanism. During this process only robots without any assigned task can be selected and they bid using only their work capacity. This work capacity depends on the type of the task that will be carried out. The leader selects the robots with the highest work capacity, until the ratio between the weight of the object and the load capacities of all the robots in the work group is greater than a fixed threshold. That is, the leader selects the best robots until this condition is verified:

$$TH_g = \frac{priority * taskWorkLoad}{\sum_{1 \le i \le N} workCapacity_i} < TH \qquad (1)$$

Where N is the number of robots of the group and $workCapacity_i$ is the individual work capacity of the ith robot for this kind of task. 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. Finally, $priority$ is the priority of the task. Thus, it limits the maximum number of robots that will be part of the group. If after this process equation 1 is not verified, the exchange of robots between groups is allowed using a new auction process. It's important to note that a leader substitution protocol has been implemented when the leader must change its work group or when it fails. The details of this processes can be found in our previous work [5, 6].

4 Motion Coordination Strategies

This section describes the motion coordination strategies implemented to reduce the interference effect. These strategies are: *follow the header* and *stay on the side*. These coordination strategies are executed after the task allocation process, described in the previous section.

4.1 Follow the Header

The first motion coordination strategy implemented is called *follow the header*. In this strategy the robots of the same group and with the same objective (object or delivery point) are attracted, instead of avoiding between them. The follow the header is implemented such that, if proper parameters are used, the robots form a line. To implement the follow the header strategy, a new behavior, called follow the header, has been used. This behavior creates an attraction force from the robot to the nearest one in the same group and with the same objective. A robot is only attracted by another one if this one is nearer to the objective. The objective can be the object to gather or the delivery point.

We suppose that the robots have no sensors to know if the detected object is another robot or it is an obstacle. Moreover, when a robot detects another one, it needs to know its group and its objective (object or delivery point). Thus, the robots periodically send a broadcast message to all the other robots with all this information. If a robot fails it can't send any more messages. When a follower doesn't receive these messages from its leader, it supposes that the leader is broken and tries to follow another robot. Thus, our system is robust and it can continue the execution of the task after the failure of some robots.

As demonstrated in [4], the highest interference during the execution of a classical foraging task is produced around the delivery point. During our foraging-like task, the highest interference is produced in points nearby the delivery point and also around the object to gather. To reduce the interference, the number of robots that can access at the same time to the neighborhood of the object or delivery point is limited. Thus, the robots wait to access to the objects or to the delivery point using physical queues.

4.2 Stay on the side

The stay on the side coordination method focuses on reducing the interference produced when a robot finds another one in the same group but with the opposite direction. This situation is produced when a robot, which goes to the object, finds another one, of the same group, which goes to the delivery point, or vice versa. The interference degree in this case is grater than in other cases, like, for example, when one robot finds another one but with the same objective or direction. The stay on the side mechanism is inspired by the behavior of the cars when they drive in a two ways road. In these roads, the cars which drive in opposite direction also use opposite sides of the road. Thus, in our case, the robots that have to go to the object will drive by one side of a 'virtual road', and the robots that go to the delivery point will use the opposite side of this road. Figure 1 shows an example of this situation. The line in the middle represents the virtual path, the big circle is the delivery point and the little square is the object to gather.

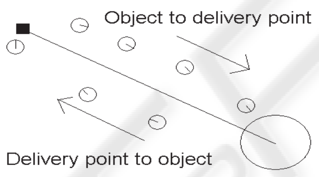


Fig. 1. Example of stay on the side strategy execution.

First of all, if the robots execute the stay on the side strategy, the leader of the work group decides which will be the path ('virtual road') that the robots of its group will use. During our experiment, this path will be a strength line from the object to the delivery point. It is under study the use of more complex paths. To implement the stay on the side strategy we use a new behavior. This behavior generates a force that moves the robots to its correct side of the path. When the robot is in its correct side, this force will continue affecting its movements until the distance to the path is grater than a fixed value. This value is a security distance between the two ways of the path. The magnitude of this force, $|V_s|$, is calculated using the following equation:

$$|V_s| = \begin{cases} 1 & \text{if } C \text{ or } d \leq D_{min} \\ \frac{(D_{max}-d)}{(D_{max}-D_{min})} & \text{if } D_{min} < d \leq D_{max} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Where D_{max} and D_{min} are two distance, such that, $D_{max} > D_{min}$. C is a preposition which indicates if the robot is in its correct side, and d is the distance between the robot position and the line which indicates the path. As during the follow the header strategy, the number of robot that can access to the object or to the delivery point at the same time can be limited.

5 Experiments and validation

This section explains the experiments carried out to evaluate both our task allocation method when diferent types of task are used and the coordination methods explained above. All the experiments has been tested using a multi-robot simulator called RoboCoT (Robot Colonies Tool). RobotCoT is a software tool developed by the authors at the University of Balearic Islands [7].

5.1 Task Allocation Experiments

Our previous works [6, 5] show the correctness of our task allocation mechanism. Now we will focus the experiments on showing the performance of the system when we use diferent type of tasks. 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. Figure 2 shows a typical situation, where the squares represent the objects to collect and the delivery point is the big circle in the middle of the image. Each object belongs to a type that defines the robot's characteristics needed to execute the task. During the experiments 5 different kind of tasks has been used. The object to gather has also a weight and each robot has a set of load capacities. The robot load capacity is the amount of weight that it can carry at once. This load capacity depends on the task type and a robot can have diferents load capacities, one for each type of task. 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. The $taskWorkLoad$ value is the object weight and the $workCapacity$ of a robot is its load capacity.

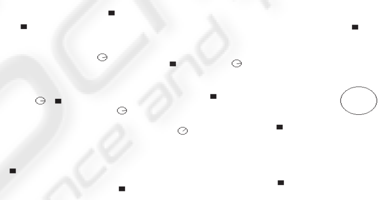


Fig. 2. Example of initial situation of the experiments

During the experiments we have used 10 objects to gather and 5 robots. Three different configurations of robots has been tested. In configuration 1 all robots have the same load capacity (3) for any task. The load capacities of configuration 2 and 3 can be seen in tables 1 (a) and 1 (b). As it can be seen in these tables the robots can have different load capacities for each type of task. During these experiments all objects have the same weight (45 units) and we use two objects of each type, that is, two objects with type 1, two objects with type 2, etc. For each configuration, we have used group threshold values (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. Also if TH=0 the robots use a method very similar to a greedy algorithm to select the task to execute. Using a greedy algorithm the robots select in each moment the best task for itself. Figure 3 shows the percentage of increment in transported weight using different values of threshold and robot configurations compared to a system with a null threshold. As it can be seen, in most cases when the threshold is not null, and, therefore when our task allocation mechanism is used, the total transported weight is increased. In all cases seems that there are an optimum threshold value, that during the experiments is about 4 or 2. In other words, it exists a optimal number of robots to execute a task. The use of learning algorithms to find the optimum TH value is under study.

Table 1. Robot's load capacity for each type of tasks. R1..R5 represent the robots and T1..T5 represent the type of the task. (a) Configuration 2. (b) Configuration 3

-	R1	R2	R3	R4	R5
T1	10	1	1	1	1
T2	1	10	1	1	1
T3	1	1	10	1	1
T4	1	1	1	10	1
T5	1	1	1	1	10

(a)

-	R1	R2	R3	R4	R5
T1	2	4	5	6	10
T2	2	12	15	1	9
T3	5	5	5	3	4
T4	8	13	14	2	3
T5	7	6	10	6	5

(b)

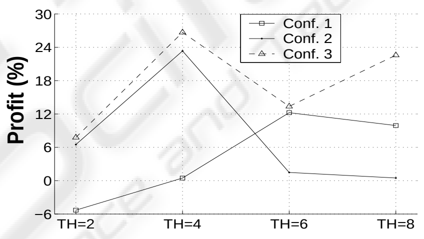
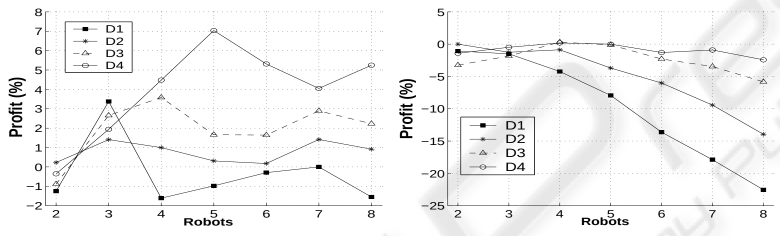


Fig. 3. Percentage of increment in transported weight when threshold (TH) is not null.

5.2 Motion Coordination Experiments

This section will analyze the results of the proposed motion coordination methods: follow the header and stay on the side. During the first set of experiments robots must transport a single object and total weight transported by the robots after 40000 time units is calculated. Four different distances from the object to the delivery point has been tested: $D_1 = 278$ units, $D_2 = 370$ units, $D_3 = 493$ and $D_4 = 600$ units. The number of robots vary from 1 to 8 and all of them have the same load capacity (2 units). Figure 4 (a) shows the percentage of increment in transported weight using the stay on the side strategy compared to not using it. During these experiments, the robots cannot stop and wait to access to the shared resources. As it can be seen, the benefit of the stay on the side increases as the distance from the object to the delivery point is greater. Figure 4 (b) shows the percentage of increment between transported weight using follow the header strategy compared with a system without any strategy. As it can be seen, unlike the stay on the side strategy, follow the header in most cases reduces the total transported weight. Therefore, unlike it seems in principle, and unlike exposed by other authors in other foraging tasks [8], a flocking behavior, like follow the header, produces a counterproductive effect in this kind of tasks. That is more evident as the distance between gather and delivery point is reduced.



(a) (b)

Fig. 4. (a) Percentage of increment in transported weight using stay on the side compared to a system without this strategy. The wait process is not used. (b) Percentage of increment in transported weight using the follow the header strategy.

During a second set of experiments we tested the stay on the side behavior using several objects with a finite weight. The task to do is the same than during the task allocation experiments, but now all tasks belong to the same type and the same weight (80 units). All robots have also the same load capacity (3 units). As it has been said during the last paragraph, the stay on the side can produce good results only when the distance between the object and the delivery point is large and there are a minimum number of robots. Thus, during these experiments the robots only use the stay on the side if the distance is grater than 160 and there are more than 3 robots in the group. Figure 5 shows the total transported weight, after 42000 time units, using and not using stay on the side method for different values of TH. As it can be seen, the stay on the side strategy increases the transported weight in all cases.

6 Conclusion and future work

First of all, this paper presents a simple method of task allocation that extends the current auction mechanisms to find the number of robots needed to execute a task. This number is unknown and depends on the task, the number of robots, the environment, etc. In this methods the robots have diferents skills (load capacities) that depends on the type

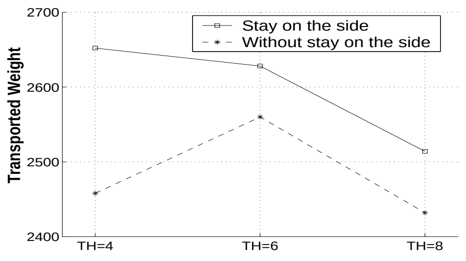


Fig. 5. Results using multiple objects and stay on the side strategy

of the task. Thus, our system can find a good task allocation. Then the paper explain two new motion coordination strategies: follow the header and stay on the side. The follow the header is a flocking-like method, but unlike happens with other foraging tasks, this strategy decrease the performance. It's important to note that to our knowledge this is the first work that study how motion strategies can decrease the interference effect in this kind of tasks. On the other hand, stay on the side can increase the total transported weight.

The work presented is in progress and has some challenging aspects to add. For the time being we are focused on a deep analysis of the data available. Another aspect of the systems that should be improved is the use of a non fixed threshold. We are also developing a learning method, based on reinforcement, to know when the robots must apply the stay on the side strategy.

References

1. Campos M., Bonabeau E. and Thraulaz G., Deneubourg J. L. Dynamic Scheduling and Division of Labour in Social Insects, Adaptive Behaviour Vol. 8-2, Singapore (2001) 83-92.
2. Chamowicz L., Campos M. and Kumar C. Dynamic Role Assignment for Cooperative Robots. 2002 IEEE International Conference on Robotics and Automation, Washington - DC (USA)(2002) 292-298.
3. Gerkey B. P. and Mataric M. J. 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.
4. Goldberg D. and Mataric M., Design and Evaluation of Robust Behavior-Based Controllers, Robot Teams from Diversity to Polymorphism, eds. AK Peters. (2002) 315-342.
5. Guerrero J., Oliver G., Multi-Robot Task Allocation Method for Heterogeneous Tasks with Priorities, 7th. International Symposium on Distributed Autonomous Robotic Systems, Tolouse (France), (2004) 171-180.
6. Guerrero J. and Oliver G. Multi-robot Task Allocation Strategies Using Auction-Like Mechanisms, Artificial Intelligence Research and Development in Frontiers in Artificial Intelligence and Applications, Vol. 100, IOS Press, (2003) 111-122.
7. Guerrero J., Oliver G. and Ortiz A., On Simulating Behaviour-based Robot Colonies in the Classroom, First EURON Workshop on Robotics Education and Training, Weingarden (Germany), 2001, 29-34.
8. Mataric M., Designing and Understanding Adaptive Group Behavior, Adaptative Behavior, vol. 4, no. 1, (1995) 51-80.